UNIVERSITY OF SOUTHAMPTON

On the efficiency of data collection and aggregation for the combination of multiple classifiers

by

Edoardo Manino

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

August 19, 2020

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Edoardo Manino

Many classification problems are solved by combining the output of a group of distinct predictors. Whether it is voting, consulting domain experts, training an ensemble method or crowdsourcing, the collective consensus we reach is typically more robust and accurate than the decisions of an individual predictor alone. However, aggregating the predictors' output efficiently is not a trivial endeavour. Furthermore, when we need to solve not just one but multiple classification problems at the same time, the question of how to allocate the limited pool of available predictors arises.

These two questions of collecting and aggregating the data from multiple predictors have been addressed to various extents in the existing literature. On the one hand, aggregation algorithms are numerous but are mostly designed for predictive accuracy alone. Achieving state-of-the-art accuracy in a computationally efficient way is currently an open question. On the other hand, empirical studies show that the collection policies we use to allocate the available pool of predictors have a strong impact on the performance of the system. However, to date there is little theoretical understanding of this phenomenon.

In this thesis, we tackle these research questions from both a theoretical and an algorithmic angle. First, we develop the theoretical tools to uncover the link between the predictive accuracy of the system and its causal factors: the quality of the predictors, their number and the algorithms we use. We do so by representing the data collection process as a random walk in the posterior probability space, and deriving upper and lower bounds on the expected accuracy. These bounds reveal that the tradeoff between number of predictors and accuracy is always exponential, and allow us to quantify its coefficient.

With these tools, we provide the first theoretical explanation of the accuracy gap between different data collection policies. Namely, we prove that the probability of error of adaptive policies decays at more than double the exponential rate of non-adaptive ones. Likewise, we prove that the two most popular adaptive policies, uncertainty sampling and information gain maximisation, are mathematically equivalent. Furthermore, our

analysis holds both in the case where we know the accuracy of each individual predictor exactly, and in the case where we only have access to some noisy estimate of it.

Finally, we revisit the problem of aggregating the predictors' output by proposing two novel algorithms. The first, Mirror Gibbs, is a refinement of traditional Monte Carlo sampling and achieves better than state-of-the-art accuracy with fewer samples. The second, Streaming Bayesian Inference for Crowdsourcing (SBIC), is based on variational inference and comes in two variants: Fast SBIC is designed for computational speed, while Sorted SBIC is designed for predictive accuracy. Both deliver state-of-the-art accuracy, and feature provable asymptotic guarantees.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

$a(t)$	predictor available at time t
α, β	prior parameters of Beta-distributed predictors
$B(\bullet, \bullet)$	beta function
c_π	exponential decay rate of the error under policy π
$d(X_j, \boldsymbol{y})$	number of labels from predictor j that match $\boldsymbol{y}$
$e(X_j, \boldsymbol{y})$	number of labels from predictor j that do not match $\boldsymbol{y}$
f_p	probability density function of p_j
g_k	ground-truth class of trial item k
G	trial set of items
h	halved log-odds z on a single item
$\hat{h}$	halved estimated log-odds z on a single item
i	item index
$\mathbb{I}(\bullet)$	indicator or characteristic function
j	predictor index
L_j	number of items labelled by predictor j
M	set of items
M_j	set of items labelled by predictor j
$\mu_i(y_i)$	item factor for mean-field approximation
n	number of predictors required to achieve the target probability p_e
n_{part}	number of particles in Mirror Gibbs
n_{flip}	number or rejuvenation steps in Mirror Gibbs
N	set of predictors
N_i	set of predictors who labelled item i
$\nu_j(p_j)$	predictor factor for mean-field approximation
o_{kj}	label of predictor j on trial item k
O	dataset containing all labels o_{kj}
p_e	target probability of a classification error
p_j	accuracy of predictor j
$\hat{p}_j$	estimated accuracy of predictor j
$\bar{p}_j$	expected accuracy of predictor j
$\pi(t)$	item selected by the policy π at time t
q	prior probability on a positive item class

R	(average) number of labels per item
s_{ij}	step induced by predictor j on item i
$\hat{s}_{ij}$	estimated step induced by predictor j on item i
s_i^k	view on the log-odds of item i according to item k
$\text{sig}(\bullet)$	sigmoid function (inverse of the log-odds)
t	current time step
T	total number of labels (timesteps)
w_e	log-odds of the target error probability p_e
w_j	weight associated to predictor j (log-odds of p_j)
$\hat{w}_j$	estimated weight of predictor j
$\bar{w}_j$	log-odds of the expected accuracy of predictor j
w_k	importance weight of particle $\boldsymbol{y}_k$
w_q	log-odds of the prior q
x_{ij}	label of predictor j on item i
X	dataset containing all labels x_{ij}
y_i	ground-truth class of item i
$\hat{y}_i$	estimated class of item i
$\boldsymbol{y}_k$	candidate item vector class (particle)
$\bar{\boldsymbol{y}}$	opposite of the item vector class $\boldsymbol{y}$
z_i	log-odds of a positive class on item i
$\hat{z}_i$	estimated log-odds on item i
$\|\bullet\|_2$	Euclidean norm of a vector

Declaration of Authorship

I, Edoardo Manino, declare that this thesis entitled *On the efficiency of data collection and aggregation for the combination of multiple classifiers* and the work presented in it is my own and has been generated by me as the result of my own original research. I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published in a number of conference and journal papers (see Section 1.3 for a detailed list).

Signature:

Date:

Acknowledgements

I would like to express my sincere gratitude to my primary supervisor Prof. Nicholas R. Jennings for bringing me to Southampton and guiding me through my PhD. Thanks for your patience, encouragement, and advice. I would also like to extend my gratitude to my second supervisor Dr. Long Tran-Thanh for pushing me to learn more and think about the bigger picture. Thanks for being always very enthusiastic about my research.

A special mention to a few other people who directly contributed to this thesis. I would like to acknowledge Alexandry Augustin for introducing me to Bayesian graphical models and sharing useful bibliographic material, Matteo Venanzi for the in-depth discussions about crowdsourcing, and Tin Leelavimolsilp for crucial suggestions on how to improve my theoretical analysis. Moreover, I would like my current post-doc supervisor, Markus Brede, for giving me plenty of time to write this thesis properly.

This research is funded by the UK Research Council under the ORCHID project grant EP/I011587/1. As an EU citizen, I am extremely thankful for their sponsorship. I would also like to acknowledge the IRIDIS High Performance Computing Facility and associated support services at the University of Southampton for allowing me to run my most computationally intensive experiments.

To all my friends in the lab, thanks for the laughter, the tears, the board games, the meals together and the absurd debates in the coffee room. You made this PhD journey as fun and enjoyable as it could have possibly been.

To all my friends in performing art societies, thanks for being an incredible creative outlet. Before coming to Southampton I would have never dreamt of performing a Shakesperian comedy, several Gilbert and Sullivan operettas, traditional folk dances and, for some reasons, Jewish sacred music. You are all completely crazy, but there have been days I was waking up only for that.

Last but not least to Dorota. You picked me up from a pit of despair and made me breathe happiness again. Thanks for showing me every day what true resilience looks like. I love you.

To my father, who taught me programming,
and to my brother, who thought it was a good idea.

Chapter 1

Introduction

Throughout history, many critical decisions have been taken not just by a single individual, but by a collective of several people. Indeed, whether it is the board of a company, a cabinet of ministers, a steering committee, the jury of a trial or a townhall meeting, the idea of consulting a plurality of opinions has an important role in modern society. The most extreme example, of course, is that of large democratic exercises, where all the adult citizens of an entire nation are asked to cast their ballot.[1]

Thus, it is not surprising that this societal phenomenon has attracted the attention of the scientific community. Following the early efforts of Marquis de Condorcet (1785), one of the main questions has been establishing a mathematical justification for the need of consulting multiple people. Accordingly, a recurrent answer is that each person, however biased or ill-informed, has a chance of contributing some information about the problem at hand. Thus, the more people we consult, the more we are likely to form a consensus on the correct choice (Berend and Paroush, 1998). Vice versa, relying on a single individual's opinion leaves us more prone to commit mistakes.

At the same time, in many practical scenarios we are confronted with more concrete questions. A classic example is that of a clinical setting where a patient's case is reviewed by several physicians before a decision is made (Dawid and Skene, 1979). How many doctors should evaluate the patient's symptoms before we can be confident that the diagnosis is correct? In the case of a life-and-death situation, for instance a major surgery or a tumor (Fanshawe et al., 2008), the answer to this question may have a crucial impact on hospital policies and health insurance coverage.

On a less harrowing note, the same question appears in the unrelated field of machine learning. There, a successful idea is training multiple distinct predictive models, and aggregating their output to form a so-called ensemble method (Dietterich, 2000). As with human experts, the predictions of an ensemble method are usually more accurate

[1] In the 2019 Indian election the turnout was more than 69% out of the 880 million eligible voters (official statistics from `www.indiavotes.com`, retrieved on March 4, 2020).

than those of an individual model alone. Similarly, the number of predictive models and the way they are trained play a crucial role in determining the predictive performance of the ensemble as a whole (Freund and Schapire, 1996; Breiman, 2001).

But perhaps the example that best illustrates the power of collective decision making is crowdsourcing. In general, this term is used to describe the practice of outsourcing work to large crowds of people (Howe, 2006). Differently from the traditional job market, the workforce is not limited to trained professionals, but any member of the public can participate (Ross et al., 2010). Furthermore, the whole process of hiring, distributing the work and collecting the results is usually performed remotely through one of the many dedicated websites.[2]

The open call nature of crowdsourcing means that there is no guarantee that the workers will provide meaningful data, which in some cases may lead to nonsensical outcomes.[3] Thus, some effort is required to develop a methodology that could effectively channel the collective effort of the crowd. Among the early crowdsourcing initiatives, the ESP game achieved this goal by asking two workers to execute the same task and accepting their submissions only when they matched (von Ahn and Dabbish, 2004). Reportedly, a set of images labelled in this way has been used to train the first iteration of the Google Image Labeler.[4] On a similar note, the Galaxy Zoo 1 project took advantage of the vibrant community of amateur astronomers to classify the shape of more than 900 000 images of galaxies taken by the Hubble telescope. By collecting the labels of dozens of workers for each image, they achieved a similar classification accuracy to that of an expert astronomer (Lintott et al., 2011).

Importantly, the two crowdsourcing projects mentioned above are both centred on the execution of a set of independent *microtasks*, small units of work that can be processed in parallel by the crowd (Difallah et al., 2015). This crowdsourcing methodology has become popular in the past decade, and it has been used to take advantage of human skills that current state-of-the-art machine learning algorithms cannot replicate. Among them we have video annotation (Vondrick et al., 2013), speech recognition (Lasecki et al., 2013), natural language processing (Snow et al., 2008) and even psychological research (Buhrmester et al., 2011).

Generally speaking, all three examples of collective decision making we mention, the clinical setting, ensemble methods and crowdsourcing, have a number of characteristics in common. First, the individual predictors are potentially inaccurate: physicians may have different levels of expertise on a specific condition, machine learning models may be able to discriminate only certain types of instances, and crowdworkers may try and

[2]For instance, Amazon Mechanical Turk (`www.mturk.com`), Figure Eight (`www.figure-eight.com`) and CrowdSource (`www.crowdsource.com`).

[3]A recent example is the victory of "Boaty McBoatface" in a poll for naming a British research vessel on `nameourship.nerc.ac.uk`.

[4]According to `crowdsource.google.com/imagelabeler`, consulted on May 9, 2016

execute the tasks as fast as possible to increase their revenue. To compound this, we might have little information about the predictors. While in machine learning we usually have full control on the training of the individual models, in the other setting the predictors are human beings and, as such, behave more like a black box of unknown properties (Kim and Ghahramani, 2012).

Second, the available resources are limited: physicians can only visit a given number of patients per day, training a very large number of base models in an ensemble method is computationally impractical, and mobilising large crowds of workers requires an equally large monetary budget. This poses some restrictions on the size of the datasets we can gather, and thus our confidence on the collective decisions we take. However, it also opens interesting avenues for research, as the methods we use for aggregating the predictors' output can be optimised to make an efficient use of the available resources (Liu et al., 2012).

Third, the predictive effort needs to cover not just a single target but a whole set of them: a medical centre has to assess the health status of a number of patients, an ensemble method is designed to process entire datasets at once, and a crowdsourcing project typically includes hundreds or thousands of tasks to solve. An important consequence of this is that the process of collecting the data is not instantaneous, but extends over a period of time that can range from a few seconds for an ensemble methods, to several days in a clinical setting or a crowdsourcing project. Thus, early results can potentially be used to inform later decisions, and improve the performance of our predictive system (Slivkins and Vaughan, 2014).

Together, these three properties define the background of our research effort. More specifically, they highlight a need for efficiency when combining the output of multiple predictors. This goal is the guiding principle of this thesis and underlies all our research requirements, as we detail in the next Section 1.1.

1.1 Research requirements

As the problem of combining the output of multiple predictors is quite broad, in this thesis we focus on the specific case of *classification* tasks. This includes any situation where we need to reach a collective agreement on a finite set of options. For instance, determining whether a patient has cancer in a clinical setting (Fanshawe et al., 2008), using an ensemble method to recognize handwritten digits (Xu et al., 1992), or asking a group of crowdworkers to assign images the correct label (Welinder et al., 2010).

An advantage of this choice is that the classification framework provides us with a well-defined metric to assess the performance of our system. Namely, we can assume that each item we want to classify comes with an unknown ground-truth class, and then

measure our ability to recover it. Clearly, the latter is deeply influenced by three factors: the accuracy of the individual predictors, whether humans or machines, the available budget, whether computational or monetary, and the techniques we use to collect and aggregate the predictors' output. In loose mathematical notation, we can summarise the link between these three factors and the probability of a classification error as the following function:

$$\mathbb{P}(error) = f(predictors, budget, algorithms) \tag{1.1}$$

Clearly, understanding the properties of the relationship in Equation 1.1 is not just an idle academic endeavour. From the perspective of a hospital manager, a data scientist, or a crowdsourcing employer, any guarantees over the predictive quality of the combined predictions is vital. This includes knowing which algorithms to deploy given the characteristics of the problem at hand and, as importantly, being able to choose the correct budget to achieve the desired accuracy. While similar concerns are routinely examined for most business services (Ackoff, 1969), the complexity of the present setting poses a major obstacle to this goal, and the existing research has not yet addressed it in a satisfactory way (Fanshawe et al., 2008; Luz et al., 2015).

Accordingly, we set our research effort to be both theoretical and algorithmic. The first is required to produce strong guarantees on the predictive performance of our system. Ideally, these guarantees shed a light on the mathematical form of the function in Equation 1.1, thus giving us a deep understanding on the influence of each factor therein. With it, we can answer a range of crucial planning questions. For instance, how much the result is affected by the quality of the predictors, how many individual predictors we need to achieve the desired accuracy, and what is the impact of different techniques for polling the predictors and aggregating their opinions.

The second is required to satisfy our need for efficiency in settings with limited resources. On the one hand, the obvious goal is to improve the classification accuracy of the system. From a classical machine learning perspective, this is achieved by refining the aggregation algorithm we use to combine the individual predictors' output. At the same time, when we have control over the process of polling the predictors, we can deploy different adaptive strategies to increase the quality and informativeness of the data we collect. Optimising the complex interplay between the collection and aggregation algorithms is vital for improving the predictive accuracy of our system.

On the other hand, while predictive accuracy is important, in some settings computational speed is of the essence. More specifically, if we want deploy an adaptive data collection strategy, our decisions on how to poll the predictors are subject to a time constraint. This is particularly important in crowdsourcing, where workers queue on an online website and might switch to a different employer if we do not serve them quickly

enough. Similarly, a slow data collection process may not be desirable for an ensemble method, when used in a time-critical application.

1.2 Research challenges

Satisfying our requirements of theoretical guarantees, predictive accuracy and computational speed is not straightforward. A first challenge is choosing the right model of classifier combination (Xu et al., 1992; Kim and Ghahramani, 2012). Since Marquis de Condorcet (1785), most authors adopt a probabilistic interpretation of the predictors' output. That is, each predictor behaves like a random variable, the observations of which are correct with some, possibly unknown, probability. Yet, each existing model introduces dependencies on a number of different underlying factors, ranging from simple and universal voting models (Nitzan and Paroush, 1982; Berend and Paroush, 1998) to complex and specific crowdsourcing models (Welinder et al., 2010; Kamar et al., 2012).

The reason for this variety is that we can make additional assumptions based on domain knowledge of each potential application. For instance, combination models for the clinical setting, where predictors are humans, tend to be fairly simple (Dawid and Skene, 1979; Fanshawe et al., 2008). Conversely, ensemble methods come with more control over the properties of the individual predictors, and thus we can afford training complex stacked models (Wolpert, 1992). Instead, for crowdsourcing applications the limiting factor is usually the amount of data we can gather from each worker. Take for example the Galaxy Zoo 2 project (Willett et al., 2013): without any constraints on their productivity, the majority of the volunteers labeled only a handful of images, as shown in Figure 1.1. From such sparse data, it is impossible to train detailed models of the predictors' behaviour.

The abundance of models in the current literature has two direct consequences. On the one hand, we lack a common basis to make a meaningful comparison between different algorithms. On the other hand, general theoretical results are rare. In fact, most authors focus on specific algorithms (Ghosh et al., 2011; Dalvi et al., 2013; Karger et al., 2014; Zhang et al., 2016; Bonald and Combes, 2017) or are only interested in proving asymptotic convergence (Berend and Paroush, 1998; Baharad et al., 2011). Still, there are a few works that propose more universal guarantees: both Berend and Kontorovich (2015) and Gao et al. (2016) derive bounds on the classification accuracy of a group of predictors that apply to several algorithms and estimators. We use these results as a baseline to compare our own theoretical contributions to.

The other challenges are related to efficiency. In this regard, the problem of aggregating the output of the individual predictors and forming a collective decision has been already explored at length in the current literature (see Zheng et al. (2017) and references therein). Nonetheless, the focus is usually put on improving the predictive accuracy,

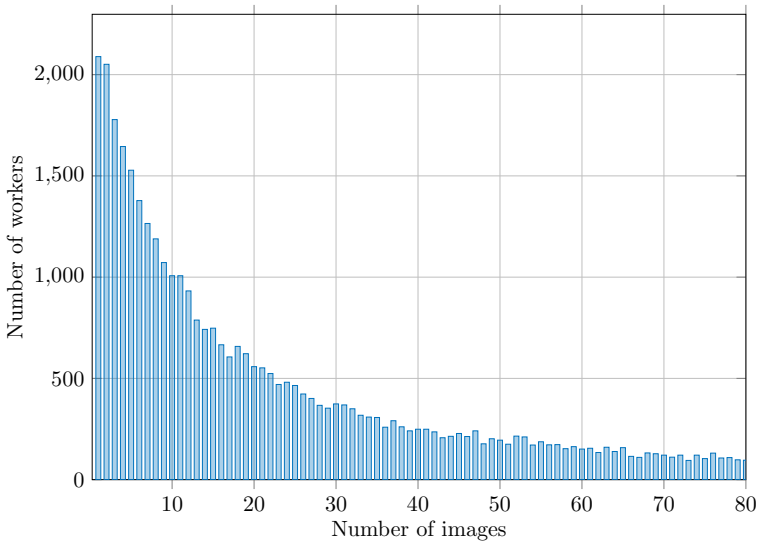


FIGURE 1.1: Number of crowdworkers who took part in the Galaxy Zoo 2 project, sorted by the number of images they labelled. For the sake of clarity, workers with more than 80 images are not shown here.

as this is the standard goal in machine learning. Unfortunately, this means that most of the existing algorithms have a high computational cost, which goes against one of our main requirements. The only notable exception is the recent work by Bonald and Combes (2017) which, however, is not suitable for settings where the properties of each individual predictor are difficult to estimate precisely. Thus, achieving both predictive accuracy and computational speed in all settings is currently an open research question.

At the same time, the aggregation problem is only one side of the coin. Several studies have shown that improving our strategies for polling the predictors has a positive effect on the collective predictive accuracy. However, the majority of the existing results are either empirical (Welinder and Perona, 2010; Barowy et al., 2012; Simpson and Roberts, 2014) or rely on restrictive assumptions on the problem setting (Chen et al., 2013; Ho et al., 2013; Khetan and Oh, 2016). Indeed, the only general theoretical result (Karger et al. (2014)) states that any data collection strategy exhibits the same asymptotic relationship between accuracy and budget:

$$\mathbb{P}(error) \leq \exp(-cR) \tag{1.2}$$

where R is the number of predictors we poll on each item we need to classify. Equation 1.2 suggests that the difference in performance between different collection strategies is hidden in the constant factor c, but no theoretical analysis exists to confirm that.

1.3 Research contributions

Against this background, we propose several contributions to the state of the art. From a high-level perspective, we identify the Dawid-Skene model (Dawid and Skene, 1979) as a reasonable common ground between all models proposed in the current literature. Thanks to this choice, we are able to develop the theoretical tools to express the relationship in Equation 1.1 in closed-form. Using these tools, we review and compare all existing data collection strategies, and prove that adaptive strategies deliver exponentially better predictive accuracy than non-adaptive ones. Furthermore, we introduce two novel algorithms to aggregate the predictors' output in an accurate and computationally efficient way.

More specifically, we solve the modelling, theoretical and algorithmic challenges outlined in Section 1.2 in the following way:

- We study three variants of the popular Dawid-Skene model that span different levels of knowledge about the properties of the predictors. In order of complexity, we study first the case where each predictor's accuracy is known, then the case where the predictor's accuracy can only be tested beforehand, and finally the case where we have no information about them. These three variants cover a large number of practical applications of the multiple classifier setting. Moreover, they allow us to evaluate the impact of different modelling choices on the corresponding theoretical and empirical results.
- We derive new theoretical bounds on the probability of a classification error under all possible combinations of predictors' accuracies, available budget, aggregation algorithm and collection strategy. With this, we completely characterise the relationship between all the factors in Equation 1.1, and give a value for the asymptotic constant c in Equation 1.2 for all possible scenarios. This is not only the first effort of this kind, but it is also an improvement over the existing results by Berend and Kontorovich (2015) and Gao et al. (2016).
- We make the first structured comparison of all data collection strategies proposed in the literature. In this regard, we prove the equivalence between the two most popular adaptive strategies: uncertainty sampling and information gain maximisation. Furthermore, we propose a new interpretation of the runtime behaviour of these two strategies in terms of a random walk in the log-odds domain. Finally, we prove in general that adaptive strategies have a faster exponential decay in the error rate as the budget increases, by at least a power of two. With these results, practitioners can finally make informed choices when deploying these strategies in real-world applications.
- We tackle the aggregation problem by introducing the Mirror Gibbs algorithm, a refinement of Monte Carlo sampling that is specifically tuned for combining

multiple independent classifiers. This technique allows us to evaluate the exact posterior distribution over the ground-truth classes in a computationally-efficient way, and infer the correct classes with better than state-of-the-art accuracy in most settings.

- We expand the portfolio of existing algorithmic solutions further with Streaming Bayesian Inference for Crowdsourcing (SBIC). This variational inference method comes in two variants. The first, called Fast SBIC, privileges computational speed over accuracy. Specifically, it runs in the same time as the simple majority voting algorithm, while delivering better predictive accuracy by orders of magnitude. The second, called Sorted SBIC, is more expensive computationally but improves over the performance of Mirror Gibbs by a noticeable margin.

Together, these contributions are an important step in addressing the requirements for theoretical guarantees, predictive accuracy and computational speed we introduced in Section 1.2. As such, they have led to a number of recent peer-reviewed publications in top conferences and journals:

- In (Manino et al., 2018) we introduce the random walk interpretation of the collection process, prove that uncertainty sampling and information gain maximisation are equivalent and make the first theoretical comparison between different data collection strategies.
- In (Manino et al., 2019a) we prove that the classification error of adaptive collection strategies decays at more than double the exponential rate of non-adaptive ones. Our analysis improves over the results in (Berend and Kontorovich, 2015) and (Gao et al., 2016), while covering both the known and unknown predictors accuracy cases.
- In (Manino et al., 2019b) we introduce the Streaming Bayesian Inference for Crowdsourcing (SBIC) algorithm and its two variants Fast SBIC and Sorted SBIC. Our analysis shows that these algorithms are competitive with the state of the art and have tractable theoretical properties.

Additionally, we are preparing a further journal paper containing the Mirror Gibbs algorithm, a general theoretical result on the aggregation problem and a third variant of SBIC. We discuss the first two in Chapter 5 and the last in Chapter 6.

1.4 Thesis outline

The remainder of this thesis is structured as follows:

- In Chapter 2 we review the relevant literature on the combination of multiple classifiers. We begin with the issue of modelling this setting, introduce the Dawid-Skene model, and discuss all its possible variants and extensions. Then, we turn to aggregation methods and touch on both simple voting schemes and advanced probabilistic inference approaches. Finally, we examine the existing data collection strategies, and trace them back to the fields of crowdsourcing and active learning.

- In Chapter 3 we begin our journey on the combination of multiple classifiers by assuming we know their accuracy. Thanks to this assumption, we can optimally aggregate the predictors' output with the weighted majority voting algorithm. This creates an ideal setting where we can formally define the different data collection policies, study their properties theoretically, and compare their performance.

- In Chapter 4 we add a layer of uncertainty, as the accuracy of the predictors is unknown but can be tested in advance. There, most of our efforts are directed towards generalising the results from Chapter 3 to this new, more challenging, setting. Fortunately, we are able to do so without invoking any restrictive assumption.

- In Chapter 5 we take our last step towards generality, and deal with predictors of completely unknown accuracy. This setting reopens the question of efficiently aggregating the predictors' output, which we address by introducing our two new algorithms Mirror Gibbs and Streaming Bayesian Inference for Crowdsourcing. After doing so, we give theoretical guarantees on the accuracy of our algorithms, we compare their performance with the state of the art empirically, and derive a general lower bound on the probability of a classification error.

- In Chapter 6 we draw our conclusions and outline possible directions for future work.

Chapter 2

Background

In Chapter 1 we explain how the same setting with multiple independent classifiers appears in different contexts across the whole computer science field. There, we introduce three motivating examples. The first, are ensemble methods, where we combine the predictions of a set of base classifiers. The second, is a clinical setting, where multiple physicians are asked their opinion on the health status of a patient. The third, is crowdsourcing, where online anonymous workers are paid to execute some repetitive labelling task.

In this section, we expand on our previous discussion by examining how this setting is treated in the existing literature. The current state of the art is quite fragmented, and many results are grounded on different assumptions (Slivkins and Vaughan, 2014). Thus, we pay particular attention to the works that are general enough to cover all of our motivating examples, rather than dealing with the minute details of a specific scenario. Furthermore, we highlight the main existing theoretical contributions, as they represent the baseline against which we benchmark many of our results in Chapters 3, 4 and 5.

We divide our review of the literature in three main sections. In the first Section 2.1, we focus on how the multiple classifier setting is usually formalised. There, we show that the classic choice is the Dawid-Skene model (see Section 2.1.1), and give a taxonomy of its variants and extensions (see Section 2.1.2). In the second Section 2.2, we look at methods to combine the raw output of each predictor in an accurate solution of the classification problem. These *aggregation* methods range from voting schemes (Section 2.2.1) to probabilistic inference, either Bayesian (Section 2.2.2) or frequentist (Section 2.2.3). In the third Section 2.3, we move on to the complementary question of data collection. When we have control on how the dataset is acquired, we can deploy a number of different collection strategies to improve its quality. These come mostly from crowdsourcing (see Sections 2.3.1 and 2.3.2) or the field of active learning (see Section 2.3.3). Finally, we give a brief summary in Section 2.4.

2.1 Modelling the multiple classifiers setting

Let us begin the review of the existing models by introducing some general notation. In this thesis, we assume that our objective is recovering the ground-truth classes of a set M of distinct items. On the one hand, this aligns well with the structure of crowdsourcing, where the workers are solving several tasks in parallel, e.g. tagging images and videos, flagging websites or translating sentences (Snow et al., 2008; Lintott et al., 2011; Lasecki et al., 2013; Vondrick et al., 2013). On the other hand, this generalises on the standard setting in ensemble methods (Dietterich, 2000) and voting (Coughlin, 1992), where the focus is usually on classifying a single item (in other terms $|M| = 1$). Similarly, this covers the full spectrum of existing research on medical consensus, which ranges from a single patient (Fanshawe et al., 2008) to a full set M of them (Dawid and Skene, 1979).

The main source of information we have in achieving our classification goal is the output X of a set of predictors N. The nature of these predictors can vary wildly: in ensemble methods we have base classifiers, in medical consensus problems we have physicians, and in crowdsourcing we have crowdworkers. This variety means that modelling how the predictors' output X is produced is a major point of contention. While in ensemble methods we usually have full control on the type and training of each base classifier (Dietterich, 2000), in the other examples the predictors can be little more than a black box of unknown properties. Furthermore, if the predictors are human beings we run into additional uncertainty: for instance, while medical professionals usually undergo some form of standardised training, in crowdsourcing the workers come from a wide range of backgrounds, age group and nationality (Ross et al., 2010).

We deal with the predictors' complexity by focusing on models that place the smallest number of assumptions on them. As argued in (Kim and Ghahramani, 2012), this choice leads to the most general models of classifier combination. Furthermore, the resulting mathematical structures allow for a more thorough theoretical analysis, which is one of the objectives of this work. At the same time, while some of these assumptions may be restrictive, generalising to more complex scenarios is always possible. In this light, we first present the classic Dawid-Skene model in the next Section 2.1.1, and then review the variants and extensions that have been proposed in the current literature.

2.1.1 Independent classifiers and the Dawid-Skene model

Among all models of the multiple classifier setting, the one proposed by Dawid and Skene (1979) has survived the test of time. Originally designed with the clinical setting in mind, it has undergone a new surge of popularity in the crowdsourcing field in recent years (Welinder and Perona, 2010; Ghosh et al., 2011; Liu et al., 2012; Dalvi et al., 2013; Karger et al., 2014; Zhang et al., 2016; Gao et al., 2016; Bonald and Combes, 2017). At the same time, it has been shown in (Simpson et al., 2011; Kim and Ghahramani, 2012)

that this model is an instance of a more general framework for combining classifiers which encompasses ensemble methods. Furthermore, a number of extensions have been proposed: we list them in Section 2.1.2.

In this thesis we focus on the so-called *one-coin* Dawid-Skene model (Ghosh et al., 2011; Liu et al., 2012; Dalvi et al., 2013; Karger et al., 2014; Bonald and Combes, 2017). Like other prototypical works in the field of classifier combination (Nitzan and Paroush, 1982; Berend and Kontorovich, 2015), the main simplifying assumption is that the ground-truth classes y_i of each item $i \in M$ are binary, i.e. $y_i \in \{\pm 1\}$. The reason for this choice is that it allows us to study the fundamental properties of the multiple classifier setting, without dealing with the peculiarities of more complex scenarios. At the same time, generalising the corresponding results to the multiclass setting is not difficult, as shown by Zhang et al. (2016) and Gao et al. (2016). In Chapter 6, we outline the steps necessary to do the same with our own contributions.

The Dawid-Skene model makes two other key assumptions. First, the output of the predictors is modelled as a stochastic process. That is to say, when predictor $j \in N$ labels any item $i \in M$, its corresponding output x_{ij} is extracted from an underlying probability distribution p_j. In the one-coin variant, p_j is nothing more than a single parameter representing the accuracy of the predictor.[1] We show how to extend this modelling choice to the multiclass case in Equation 2.2.

Second, the predictors are independent. More specifically, the probability of observing x_{ij} is only conditional on the ground-truth class y_i of the corresponding item. This assumption of independence is a standard feature in machine learning, since it yields more tractable methods (Murphy, 2012). In the present setting, it is also supported by practical considerations. In fact, ensemble methods strive to create independent base classifiers by subsampling the training set via bagging (Breiman, 1996), dagging (Ting and Witten, 1997) or boosting (Freund and Schapire, 1996). Conversely, the output of human predictors tends to be independent unless they are sharing information in the background. In crowdsourcing, the workers rarely share information, and when it happens we can detect it with statistical methods (Naroditskiy et al., 2014; KhudaBukhsh et al., 2014; Wang et al., 2015).

Due to its structure, the Dawid-Skene model belongs to the larger family of probabilistic graphical models (Koller and Friedman, 2009). These models are characterised by the property that the joint distribution of all the random variables can be factorised as a product of independent terms. In this case, for all choices of the ground-truth classes $\boldsymbol{y}$ and the predictors' distributions $\boldsymbol{p}$, we can write:

$$\mathbb{P}(X|\boldsymbol{y}, \boldsymbol{p}) = \prod_{x_{ij} \in X} \mathbb{P}(x_{ij}|y_i, p_j) \tag{2.1}$$

[1] This is also the reason for the model's name: each predictor behaves like a biased coin with a fixed probability of landing on the correct side.

where $X = \{x_{ij}\}$ is the set of all observed labels. We report a compact representation of the graph associated with the Dawid-Skene model in Figure 2.1, where we use the visual notation proposed by Dietz (2010): random variables are represented as nodes, probabilistic dependency as directed arcs, and multiple copies of the same structure as plates. Furthermore, observed variables are shaded.

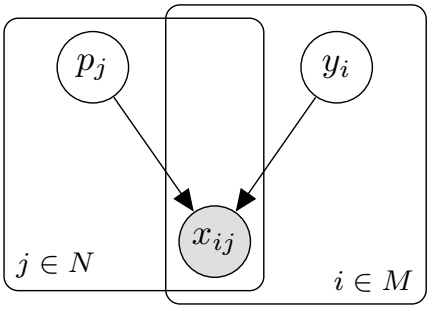


FIGURE 2.1: A graphical representation of the Dawid-Skene model.

While the original Dawid-Skene model is fully represented by the graphical model in Figure 2.1, additional assumptions are often imposed on the unobserved variables $\boldsymbol{p}$ and $\boldsymbol{y}$. This is because it would be otherwise impossible to infer their values from the dataset X alone (Bonald and Combes, 2017). We review the impact of these additional assumptions on the resulting inference algorithms in Section 2.2. In turn, they inform our modelling choices in Chapters 3, 4 and 5, where we analyse the inference accuracy under the one-coin Dawid-Skene model as we progressively remove information about the underlying generative process.

2.1.2 A taxonomy of extended models

The Dawid-Skene model has been extended in various directions, which can be classified according to two main aspects: the nature of the items and the behaviour of the predictors. A further difference exists, namely in the availability of the individual predictors. However, since the latter pertains to the data collection process, we review it in the corresponding Section 2.3.

As we highlight in Section 2.1.1, a number of extensions of the one-coin Dawid-Skene model regard the number of classes the items can have. In particular, Ipeirotis et al. (2010), Simpson et al. (2011), Kim and Ghahramani (2012), Zhou et al. (2012), Zhang et al. (2016) and Gao et al. (2016) move away from the binary case and consider items with k possible classes. As a result, every predictor j is characterised by a *confusion matrix* $P_j \in [0,1]^{k \times k}$, whose entries p^j_{xy} represent the conditional probability of reporting

label x given that the true class of the item is y:

$$P_j = \begin{pmatrix} p^j_{11} & \cdots & p^j_{1k} \\ \vdots & \ddots & \vdots \\ p^j_{k1} & \cdots & p^j_{kk} \end{pmatrix} \tag{2.2}$$

where each column sums up to one.

Other authors generalise the basic model by adding a notion of *difficulty* of an item. That is to say, the predictors are more likely to commit a mistake on a difficult item than an easy one. This can take the form of a single parameter for each item as in Whitehill et al. (2009), Dai et al. (2011), Chen et al. (2013) and Khetan and Oh (2016), or a confusion matrix indicating which classes are more likely to be misreported (Zhou et al., 2012). Alternatively, we can have a *heterogeneous* model where the probability of a mistake depends on the type of item at hand and the expertise of each predictor (Welinder et al., 2010; Ho et al., 2013; Zhang et al., 2015).

In some cases the items are not isolated entities, but can be mapped to a feature space. For example, in image classification we may have access to a set of high-level features extracted from the pixels of the image, and use them to predict the label of the image and the performance of the predictors (Kamar et al., 2012). More in general, if the objective is to train a supervised classifier by using our predictions over the item classes $\boldsymbol{y}$ as part of the training set, we can optimise over the accuracy of such a classifier rather than the accuracy on $\boldsymbol{y}$ itself (Yan et al., 2011; Lin et al., 2014).

In a scenario with repeated interactions over a long stretch of time, we may observe fluctuations in the predictors' performance. Donmez et al. (2010) are the first to propose a method to monitor the predictors' accuracy, and select only the best subset of them. Conversely, Simpson et al. (2013) propose a dynamic model of the predictors for the sake of improving the accuracy of their aggregated answers. The authors of both papers allow the parameters of each predictor to drift in small stochastic steps, and propose a method to track their values over time.

Finally, there have been attempts to move away from the assumption of independence between the predictors. Kim and Ghahramani (2012) proposes two general *dependent* classifier combination models, but find minimal empirical improvement over their independent counterparts. On a different note, KhudaBukhsh et al. (2014) proposes a method to detect when a group of predictors copies each others' answers. This is vital to avoid crowdworker *collusion* in crowdsourcing applications. Similarly, Wang et al. (2015) is concerned with explicit coordinated attacks to change the system predictions on a specific subset of items.

While all of these extended models are undoubtedly interesting, they target specific characteristics of the problem at hand. Moreover, a general issue with richer models is

that we need more data to train them and avoid overfitting (Murphy, 2012). For this reason, in this thesis we mainly focus on the basic Dawid-Skene model, as it is a good balance between expressiveness and simplicity. Studying more complex models is the focus of our future work (see Chapter 6).

2.2 Aggregating the predictors' output

Once we have a model that describes the properties of the individual predictors and how their collective output X is generated, we need a way to combine the data X into a prediction $\hat{y}_i$ over the class of each item $i \in M$. The goal here is to recover the unknown vector of ground-truth classes $\boldsymbol{y}$, even though the noise in the dataset X may hinder the accuracy of our predictions. In general, we can define an *aggregation method* as a function $g : \{+1, 0, -1\}^{|M|\times|N|} \to \{\pm 1\}^{|M|}$ from the space of the dataset X to the space of item classes. Of course, some specific aggregators might have access to more information than just X, if that is the case we adapt our notation accordingly. More importanly, with this definition we can measure the performance of an aggregation method by means of its expected zero-one loss over the item set M:

$$\mathbb{P}(\hat{y}_i \neq y_i) = \mathbb{E}_{X,\boldsymbol{y}}\Big\{\mathbb{I}\big(g_i(X) \neq y_i\big)\Big\} \tag{2.3}$$

which we refer to as predictive accuracy, misclassification rate or probability of a classification error throughout the whole thesis.

In this section, we review a wide range of aggregation methods and discuss their properties. We begin with voting schemes, which underpin much of our contributions in Chapters 3 and 4, and are the building block for more complex aggregation methods. Then, we introduce probabilistic inference methods that are based on Bayesian statistics. These are known for delivering state-of-the-art performance in any setting, but rarely offer any theoretical guarantees. We dedicate most of Chapter 5 to address this research gap. Finally, we move onto aggregation algorithms which are rooted in frequentist statistics and the method of moments. These come with good theoretical guarantees, but have strict requirements on the properties of the dataset X. As we show in Chapter 5, the latter can limit their empirical performance in many settings.

2.2.1 Voting schemes

The literature on voting is extensive and cover various facets of this fascinating field (Coughlin, 1992). However, there are only a few methods that are relevant to our work, and so we present them here in order of complexity.

2.2.1.1 Majority voting

The problem of reaching a consensus in the presence of contrasting opinion is as old as human societies. A common solution is collecting the votes of a group of individuals and selecting the option that received the largest number of preferences. This *majority voting* scheme has been successfully used in computer science as well, where its applications range from ensemble methods (Freund and Schapire, 1996; Breiman, 2001) to crowdsourcing (Snow et al., 2008; Barowy et al., 2012). We can formalise majority voting as follows:

$$\hat{y}_i = \underset{\ell \in \{\pm 1\}}{\operatorname{argmax}} \Big\{ \sum_{j \in N} \mathbb{I}(x_{ij} = \ell) \Big\} \tag{2.4}$$

where ties are broken uniformly at random.

The first theoretical analysis of this aggregation method dates back to 1785, when De Condorcet published his jury theorem (Marquis de Condorcet, 1785). This theorem states that, by increasing the number of voters, the probability of committing a mistake under majority voting goes to zero in the limit. This result is valid if each voter chooses the correct answer with probability larger than random chance, i.e. $p_j > 1/2$ for all $j \in N$, and their votes are independent.

More recently, this result has been generalized to crowds where some workers have $p_j < 1/2$, as long as the average accuracy $\bar{p} = \frac{1}{|N|} \sum_{j \in N} p_j$ remains above random chance. Formally, Berend and Paroush (1998) state that the probability of a classification error goes to zero for $|N| \to \infty$, as long as the following condition holds:

$$\lim_{|N| \to \infty} (\bar{p} - \frac{1}{2}) \sqrt{|N|} \to \infty \tag{2.5}$$

Both these results support the robustness of majority voting as an aggregation method when we have an abundance of votes. However, there is scope for improvement when this is not the case, as we explain in the next Section 2.2.1.2.

2.2.1.2 Weighted majority voting

Once we consider that each individual predictor may have a different accuracy p_j, we can see the limitations of the majority voting method. Specifically, the presence of predictors with $p_j < 1/2$ decreases the predictive performance of the system, as they tend to select the wrong item class more often than not. Ironically, we could achieve a better result by toggling the sign of all their respective labels x_{ij}. More in general, we can define a better aggregation method if we take into account the accuracy of each predictor, and treat their output differently.

Along this line of reasoning, we can define a *weighted majority voting* method that

associates a scalar weight $w_j \in \mathbb{R}$ to each predictor $j \in N$, and generates a prediction on the item classes as follows:

$$\hat{y}_i = \text{sign}\Big\{ \sum_{j \in N} x_{ij} w_j \Big\} \tag{2.6}$$

where ties are broken uniformly at random.

Equation 2.6 is flexible enough to accomodate a wide range of behaviours. As an illustration, a noisy predictor can be associated with a weight close to zero, whereas an accurate one can be assigned a large positive weight. According to Nitzan and Paroush (1982), when the votes are independent and the parameters p_j are known, we can define a set of *optimal* weights as follows:

$$w_j = \log\Big(\frac{p_j}{1-p_j}\Big) \tag{2.7}$$

We explore the connection between the weights in Equation 2.7 and a probabilistic interpretation of weighted majority voting in Chapter 3, as it is relevant to our discussion therein. For now, let us note that w_j is zero for uninformative predictors ($p_j = 1/2$), negative for $p_j < 1/2$ and positive for $p_j > 1/2$. In this sense, this set of optimal weights satisfies all the properties outlined in the present section.

Finally, the work of Berend and Kontorovich (2015) and Gao et al. (2016) provide us with guarantees on the convergence of Equation 2.6 as the number of predictors grows to infinity. We review these results in Chapter 3, where we propose our own bounds on the predictive accuracy of weighted majority voting. These compare favourably with the aforementioned results (see Remarks 3.2, 3.3, 3.5 and 3.6).

2.2.1.3 The plug-in aggregator

However, in most settings we do not have access to the accuracy p_j of each predictor. For example, in ensemble methods we can only estimate p_j on a validation set. Similarly, in crowdsourcing we can only test each worker on a limited number of *golden* tasks.[2] The question is whether this sort of estimates carry enough information to be used in a weighted majority scheme.

We formalise the setting by assuming that each predictor is subjected to $|G|$ trials, and its number of correct answers d_j is known. Given this, we have two main options for computing an estimate of the predictors' accuracies (Murphy, 2012). On the one hand, we can choose the maximum likelihood estimate $\hat{p}_j^{ML} = d_j/|G|$. On the other hand, we can choose a Bayesian approach and compute the posterior expectation $\hat{p}_j^{BA} =$

[2] Interestingly, some authors advocate for hiding these golden tasks in the normal workflow of the crowdworkers, so that they are not aware of being monitored (Oleson et al., 2011).

$\mathbb{E}_{p_j|d_j,|G|,f_p}(p_j)$, given a prior distribution f_p. With either of these options, we can deploy the so-called *plug-in* aggregator (Berend and Kontorovich, 2015; Gao et al., 2016), which is defined as follows:

$$\hat{y}_i = \text{sign}\Big\{\sum_{j\in N} x_{ij}\hat{w}_j\Big\} \qquad \text{where} \qquad \hat{w}_j = \log\Big(\frac{\hat{p}_j}{1-\hat{p}_j}\Big) \tag{2.8}$$

where ties are broken uniformly at random.

Note that Equation 2.8 is almost indentical to the weighted majority voting scheme of Section 2.2.1.2. At the same time, the introduction of the approximate weights $\hat{w}_j$ changes its properties in a significant way. In this regard, Berend and Kontorovich (2015) study the plug-in aggregator in conjunction with the maximum likelihood estimates $\hat{p}_j^{ML}$. Their analysis shows that the fact that these estimates can hit both extreme values $\hat{p}_j^{ML} = 0$ and $\hat{p}_j^{ML} = 1$ is not an issue. However, the resulting guarantees depend heavily on the estimates $\hat{p}_j$ being close to their ground-truth value p_j, as we detail in Appendix E. Our own results from Chapter 4 do not suffer from this limitation. Conversely, there are no existing results on the accuracy of the plug-in aggregator used with the Bayesian estimates $\hat{p}_j^{BA}$. We fill this gap in the current literature in Chapter 4.

2.2.2 Bayesian inference

The issue with the more advanced voting schemes we review in Sections 2.2.1.2 and 2.2.1.3 is that they require some prior knowledge on the accuracy p_j of each predictor. When this piece of information is not available, we are left with two options: revert back to the simpler majority voting scheme, or try and learn the values of p_j from the dataset X itself. The latter is possible if we can measure how often each predictor agrees with the output of the other ones. In this section we review a number of Bayesian approaches to do so, while we delve into the frequentist ones later in Section 2.2.3.

At a high level, Bayesian inference works by defining a prior distribution on the latent variables of interest (Murphy, 2012). In our case, this invariably includes the item classes $\boldsymbol{y}$, and the predictors' accuracies $\boldsymbol{p}$, as in (Whitehill et al., 2009; Welinder et al., 2010; Dai et al., 2011; Kim and Ghahramani, 2012; Karger et al., 2014). Then, after observing the dataset X, the objective is to compute a posterior distribution of the latent variables, e.g. $\mathbb{P}(\boldsymbol{y}, \boldsymbol{p}|X)$. If the data is informative enough, such a posterior distribution concentrates around the ground-truth values with high probability, which allows us to form a meaningful prediction.

The details of how the posterior is defined depends on the model we choose. For this reason we defer most of the related technical details to our discussion in Chapter 5, where they become relevant (See Section 5.2 in particular). Here, we focus on the algorithmic side of Bayesian inference.

2.2.2.1 Approximate mean-field variational inference

A classic approach to Bayesian inference for our setting is the approximate mean-field method proposed in (Liu et al., 2012). This method builds on the intuition of measuring the level of agreement of each predictors with the other ones in an iterative way. The idea is to begin with a majority voting estimate of the item classes, and use it to assess the predictors' accuracies. Then, we can use the latter to refine our estimate on the items, and so on until convergence.

More formally, the prior of choice for the predictor's accuracy is the Beta distribution $p_j \sim \text{Beta}(\alpha, \beta)$. This induces identical estimates $\hat{p}_j = \alpha/(\alpha + \beta)$ of each predictor's accuracy. With them, we can iterate between the following two steps until convergence:

$$\text{Item step: } \hat{z}_i \propto \sum_{j \in N} x_{ij} \log\left(\frac{\hat{p}_j}{1 - \hat{p}_j}\right) \tag{2.9}$$

$$\text{Predictor step: } \hat{p}_j = \frac{\sum_{i \in M_j} \text{sig}(x_{ij}\hat{z}_i) + \alpha}{|M_j| + \alpha + \beta} \tag{2.10}$$

where M_j is the set of items labelled by predictor j and $\text{sig}(\bullet)$ is the inverse of the logit function as detailed in Appendix D. Finally, we can form the final predictions on the item classes as $\hat{y}_i = \text{sign}\{\hat{z}_i\}$.

Notably, this algorithm is very similar to the original expectation-maximisation method in (Dawid and Skene, 1979), with the only difference of some additional smoothing in Equation 2.10. Likewise, both algorithms build on top of the plug-in estimator we present in Section 2.2.1.3, as Equation 2.9 shows. Crucially, these iterative methods are provably more accurate than majority voting, as shown in (Baharad et al., 2011). However, there are no other guarantees on their performance, with the exception of the general bounds by Gao et al. (2016), which we discuss in Chapter 5.

Other authors have employed variational inference on more complex models of classifier combination. For example, both Welinder and Perona (2010) and Simpson et al. (2011) study a multiclass model where we also learn the share of items per each class. Similarly, Whitehill et al. (2009) and Zhou et al. (2012) extend the basic model to include predictors with varying accuracy depending on the item they label. However, since our focus is directed towards the foundations of the original Dawid-Skene model, all these methods are outside the scope of our work.

2.2.2.2 Belief propagation as matrix factorisation

Next, we introduce another iterative method proposed by Karger et al. (2014). Their work stems from the observation that, if we represent X in matrix form, we obtain a regular form. Specifically, the expected value of the $|M| \times |N|$ full matrix $\mathbb{E}\{X\} =$

$\boldsymbol{y}(2\boldsymbol{p}-1)^T$ has rank one. Therefore, we can try and estimate the values of $\boldsymbol{y}$ and $\boldsymbol{p}$ by computing the principal components of the observed matrix X. While this observation is not new, see for instance (Ghosh et al., 2011), the author's approach is particularly effective, and can be applied to extended variants of the Dawid-Skene model, as done in (Khetan and Oh, 2016).

With this in mind, the authors propose an iterative message-passing algorithm that resembles a power iteration method. First, initialise the predictor messages $\hat{w}_{ij}$ to Gaussian-distributed values with positive mean. Then, repeat the two following steps for each pair i, j such that $x_{ij} \neq 0$ until convergence:

$$\text{Item messages: } \hat{z}_{ij} \propto \sum_{j' \in N \setminus j} x_{ij'} \hat{w}_{ij'} \tag{2.11}$$

$$\text{Predictor messages: } \hat{w}_{ij} \propto \sum_{i' \in M \setminus i} x_{i'j} \hat{z}_{i'j} \tag{2.12}$$

Finally, we can predict the item classes by $\hat{y}_i = \operatorname{sign}\big\{\sum_{j \in N_i} x_{ij} \hat{w}_{ij}\big\}$, which corresponds to recovering the sign of the entries of the first eigenvector.

This algorithm has a strong connection with the plug-in aggregator we present in Section 2.2.1.3. In fact, Equation 2.11 is nothing but weighted majority voting with an approximate set of weights $\hat{\boldsymbol{w}}$. At the same time, the weights in Equation 2.12 can be justified if we assume that the prior over p_j is discrete, and $\mathbb{P}(p_j = 1) > \mathbb{P}(p_j = 0)$ with the probability of all other values being zero. This distribution, commonly referred to as the *spammer-hammer* prior (Liu et al., 2012; Karger et al., 2014), makes the problem of inferring every $\hat{w}_{ij}$ symmetric to that of inferring $\hat{z}_{ij}$. Moreover, it allows us to interpret the algorithm as a belief propagation method (Liu et al., 2012).

Crucially, this algorithm is the only relevant Bayesian method with convergence guarantees. Specifically, the authors are able to bound the probability of a classification error from above:

$$\mathbb{P}(\hat{y}_i \neq y_i) \leq 2 \exp\Big(-\frac{R}{32} \mathbb{E}_{p_j}\big((2p_j - 1)^2\big)\Big) \tag{2.13}$$

where each item must be labelled by R different predictors. Incidentally, the authors are also able to show that this exponential relationship between R and the probability of an error is valid for any inference algorithm, albeit the constant factor inside the exponential may differ. Most of our theoretical results in this thesis are directed towards quantifying and comparing this constant factor under different settings.

2.2.2.3 Other probabilistic methods

Both the methods we describe in Sections 2.2.2.1 and 2.2.2.2 are iterative. This is a common choice in the existing literature, but not unique. We conclude this review of Bayesian inference by offering two alternative approaches.

On the one hand, the posterior distribution can be estimated directly via Monte Carlo sampling. For the Dawid-Skene model, the standard approach is Gibbs' sampling, as shown in (Kim and Ghahramani, 2012). This allows us to sample the posterior distribution, without the need of expressing it in analytical form. In Chapter 5 we show that such an approach is not efficient, and introduce an improved version of Sequential Monte Carlo sampling (Chopin, 2002) to address this issue.

On the other hand, some authors propose to compute the maximum-a-posteriori (MAP) value of all the latent variables by gradient ascent. In this regard, Welinder et al. (2010) applies this strategy on a large hierarchical generative model of image tagging. Similarly, Jung and Lease (2012) applies this strategy to factorise the matrix X. Unlike the algorithm in Section 2.2.2.2, their approach assumes that the matrix has rank larger than one in expectation. We take advantage of the connection betweem gradient ascent and variational inference in Chapter 5, where we introduce our Streaming Bayesian Inference for Crowdsourcing algorithm.

2.2.3 Methods of moments

We conclude our review of aggregation methods by looking at a few algorithms rooted in frequentist statistics. Although this approach is less popular than its Bayesian counterpart, the resulting algorithms offer some theoretical guarantees on the accuracy of their predictions. Unfortunately, in order to do so they often require strong assumptions on the input dataset X, that are not necessarily met in practice.

2.2.3.1 Frequentist matrix factorisation

In this regard, Dalvi et al. (2013) propose an algorithm to compute a set of approximate weights to use in the plug-in aggregator we present in Section 2.2.1.3. Their method works as follows. Call $A = \mathbb{I}\{X\}$ the predictor-item adjacency matrix, and $E = \mathbb{E}_{X|A,\boldsymbol{p},\boldsymbol{y}}\{X\}$ the corresponding expected value of the dataset X in matrix form. Then, it is easy to see that the relationship $E^T E = (A^T A) \otimes (\boldsymbol{w}\boldsymbol{w}^T)$ holds, where $\boldsymbol{w} = (2\boldsymbol{p} - 1)$ and $\otimes$ is the element-wise product. If we ignore the expectation operator, we can compute an approximation of the weights $\boldsymbol{w}$ as $\hat{\boldsymbol{w}} = v_1(X^T X) \oslash v_1(A^T A)$, where $v_1(\bullet)$ is the principal eigenvector and $\oslash$ is the element-wise division.

Note that the weights computed by the algorithm in (Dalvi et al., 2013) are linear in our estimate of the predictor's accuracy, i.e. $\hat{w}_j = 2\hat{p}_j - 1$, as opposed to the non-linear ones in Equation 2.2.1.3 for the plug-in agggregator. This makes the algorithm very conservative in its estimates, and reduces its predictive accuracy. Furthermore, the way the weights are computed imposes some restrictions on which settings this algorithm can be applied to. Specifically, we get a convergence guarantee only

if the number of labels per item is $R > \sqrt{|N|}/(8\epsilon\|\boldsymbol{w}\|_2^2)$, the number of labels per predictor is $L > 256\log(|N|/\delta)\sqrt{|N|}/(\epsilon^2\|\boldsymbol{w}\|_2^2)$, and the second eigenvalue of A^TA is $\mu_2 < \epsilon\|\boldsymbol{w}\|_2^2 RL/(16\sqrt{|N|})$ for some positive constants $\epsilon, \delta < 1$. When all these technical constraints are satisfied, we know the probability of a classification error is bounded as follows:

$$\mathbb{P}(\hat{y}_i \neq y_i) \leq \exp\left(- RO\left(\frac{\|\boldsymbol{w}\|_2^2}{\sqrt{|N|}} - \frac{1}{R} - \frac{1}{\sqrt{L}}\right)^2\right) \tag{2.14}$$

In Chapter 5 we present our own algorithmic contributions which are both guaranteed to work in all settings, and exhibit a more direct relationship between the probability of a classification error and the paramenters R, L and $\boldsymbol{p}$.

2.2.3.2 A spectral method to ensure EM convergence

Conversely, Zhang et al. (2016) study the convergence properties of the expectation-maximisation algorithm from a frequentist perspective. Expectation maximisation (EM), like the other iterative algorithms in Section 2.2.2.1, optimises a non-convex objective function. In order to ensure convergence to the global optimum, the authors focus on initialising it properly. They do so with a tensor factorisation algorithm, which is inspired by the work of Anandkumar et al. (2015). We review here its binary one-coin variant.

First, define the level of agreement between two predictors a and b as:

$$H_{ab} = \frac{1}{2}\left(\frac{\sum_{i\in M_{ab}} \mathbb{I}(x_{ia} = x_{ib})}{|M_{ab}|} - \frac{1}{2}\right) \tag{2.15}$$

where M_{ab} is the subset of items labelled by both a and b. Then, for each predictor j, choose a pair of predictors that maximises $(a_j, b_j) = \text{argmax}_{a\neq b\neq j}\{|H_{ab}|\}$. Finally, initialise the accuracy estimates of each predictor to the following value:

$$\hat{p}_j = \frac{1}{2} + \text{sign}(H_{ja_j})\sqrt{\frac{H_{ja_j}H_{jb_j}}{H_{a_jb_j}}} \tag{2.16}$$

The main drawback of this algorithm is that we need considerable overlap between the output of the predictors for the estimates in Equation 2.16 to be different from 1/2. If this is not the case, the EM algorithm cannot converge. For this reason, it is common practice to initialise the predictors estimates with a small positive value, as suggested by (Liu et al., 2012).

2.2.3.3 Triangular estimation

Finally, Bonald and Combes (2017) propose the *triangular estimation* algorithm (TE), which shares many ideas with the algorithm we present in Section 2.2.3.2, but does not need the expectation-maximisation phase. At its core, the TE algorithm estimates the correlation between the output of each pair of predictors a and b as follows:

$$C_{ab} = \frac{\sum_{i \in M_{ab}} x_{ia} x_{ib}}{\max\left(|M_{ab}|, 1\right)} \tag{2.17}$$

where M_{ab} is the subset of items labelled by both a and b. Then, for each predictor j, we choose the most informative pair of predictors as $(a_j, b_j) = \text{argmax}_{a \neq b \neq j}\{|C_{ab}|\}$. With it, we can estimate the magnitude of the predictor's accuracy:

$$|\hat{w}_j| \equiv |2\hat{p}_j - 1| = \sqrt{\left|\frac{C_{ja_j} C_{jb_j}}{C_{a_j b_j}}\right|} \tag{2.18}$$

Similarly, the sign of $\hat{w}_j$ can be recovered from the correlations as follows. First, define $k = \text{argmax}_j\{|\hat{w}_j^2 + \sum_{j' \neq j} C_{j'j}|\}$. Then, choose $\text{sign}(\hat{w}_j) = \text{sign}(\hat{w}_k C_{jk})$ for all predictors $j \neq k$, and $\text{sign}(\hat{w}_k) = \text{sign}(\hat{w}_k^2 + \sum_{j' \neq k} C_{j'k})$ for predictor k instead.

The triangular estimation algorithm suffers from the same drawback as the algorithm in Section 2.2.3.2, in that we need a considerable overlap between the output of the workers to get meaningful values of $\hat{p}_j$. However, when this condition is met, TE delivers state-of-the-art performance, while offering some loose theoretical guarantees on the accuracy of its estimates $\hat{p}_j$. For this reason, we include this algorithm in our experimental comparison in Chapter 5.

2.3 Collecting the predictors' data

Throughout our review of models and aggregation methods in Sections 2.1 and 2.2 we always assume that the observations in the dataset X are given. At the same time, we know that number of labels in X affects the accuracy of our predictions, as proven by the theoretical results in Equations 2.13 and 2.14. Moreover, it is often the case that we can control the number of labels we observe. For instance, in ensemble methods we can choose how many base classifiers we train. Similarly, in crowdsourcing we can ask additional workers to label our set of items. Thus, in this section we take a closer look at the problem of collecting the labels in the dataset X, and review relevant existing work.

The first issue in managing the data collection process is a matter of resources. These can be computational resources in the case of ensemble methods or actual money in the case of crowdsourcing (Sorokin and Forsyth, 2008). In the latter case, setting the

right monetary compensation in order to attract enough crowdworkers without wasting resources is a research question in itself, and has been studied in (Mason and Watts, 2009; Horton and Chilton, 2010; Singla and Krause, 2013). A simpler scheme is studied in (Nguyen et al., 2015), where expert labelers are more costly than regular crowdworkers. On a similar note, there have been recent efforts to evaluate the value of each data point in supervised machine learning, by measuring their impact on the prediction accuracy (Ghorbani and Zou, 2019). In the present discussion we adopt the common assumption that each data point comes at a fixed cost, by convention $c = 1$, and that we have a limit T on the total *budget* that we can spend (Chen et al., 2013; Tran-Thanh et al., 2014; Zhang et al., 2015).

The second issue is assessing how much control we have on the data collection process. A classic assumption from the online learning community is that all the predictors in the current pool N are always available (Littlestone and Warmuth, 1994; Donmez et al., 2010; Chen et al., 2013; Tran-Thanh et al., 2014; Zhang et al., 2015). In this setting, the objective is to quickly identify which predictors are the most accurate, and discard the rest. However, this assumption is too strong for one of our main applications, crowdsourcing, where the workers are free to join and leave a project whenever they please. In this setting, it is difficult to predict the quality of incoming labels (Mao et al., 2013), and the availability of workers is best modelled as a stochastic process (Faridani et al., 2011). As a consequence, the corresponding literature has focused on algorithms that employ all the predictors that are available at each time (Welinder and Perona, 2010; Barowy et al., 2012; Simpson and Roberts, 2014; Bonald and Combes, 2017). In this thesis we adopt the second approach, as it is more general.

All things considered, we are left with the problem of collecting a fixed number T of labels across the set of items M. The existing literature proposes some potential solutions. On the one hand, the crowdsourcing community has designed a number of *collection policies* that regulate the data acquisition process. We review them in Sections 2.3.1 and 2.3.2. On the other hand, the machine learning community has developed the concept of *active learning*. We explain how it applies to our setting in Section 2.3.3.

2.3.1 Budget sharing

Since the total budget of labels T is fixed, we can treat it as a limited resource, and choose a policy that shares it as evenly as possible across the tasks. In Section 2.2.2.2 we mentioned the work of Karger et al. (2014), which is based on the assumption that each item receives the same number of labels $R = T/|M|$. The main advantage of this *uniform allocation* policy is that it needs no prior information to be executed, and the collection schedule could even be fixed in advance. However, Simpson and Roberts (2014) show that the empirical performance of the uniform policy is far from optimal, a result that we are able to confirm theoretically in Chapters 3 and 4.

Alternatively, we can try and improve the collection strategy by taking into account the different accuracy of the individual workers. In the crowdsourcing literature, Ho et al. (2013) build an allocation strategy on top of a variant of the plug-in aggregator (see Section 2.2.1.3). Their idea is to associate each predictor with a positive weight $\hat{w}_j > 0$ that measures how accurate they are. Then, they compute an allocation schedule that evenly balances the sum of weights on each item. We revisit this *weight balancing* policy in Chapters 3 and 4, where we show that it is only marginally better than the simple uniform allocation policy.

Outside of the crowdsourcing field, the problem of *budget sharing* has received considerable attention. Both the fair allocation of indivisible goods in the economic field (Golovin, 2005; Asadpour and Saberi, 2007) and multi-set partitioning in combinatorics (Karmarkar and Karp, 1983; Korf, 1998) are well understood and offer efficient algorithms for their solutions. However, these techniques require us to know in advance the weights and the number of labels from each predictor, which makes them unsuitable for our setting with stochastic arrivals. Conversely, the work on online scheduling for parallel machines is more suitable to our case, since it does not assume the prior knowledge of the list of incoming jobs (Albers, 1997; Avidor et al., 2001). Specifically, in Chapters 3 and 4 we use a greedy algorithm originally proposed by Graham (1966) to implement the weight balancing policy efficiently.

2.3.2 Measuring disagreement

Another approach to the label collection problem is to track the current consensus on the items, and acquire additional labels where the predictors are disagreeing. Following this line of reasoning, Barowy et al. (2012) proposes a policy that keeps collecting new labels until all the items are classified with confidence larger than a threshold δ. In order to do so, the authors consider the output of a group of N predictors who choose their labels completely at random, and compute the probability that fewer than r of them will agree on the labeling of item i:

$$\epsilon(N, r) = \mathbb{P}\Big(\max_{\ell \in \{\pm 1\}} \Big\{ \sum_{j \in N} \mathbb{I}(x_{ij} = \ell) \Big\} < r \Big) = \sum_{k=|N|-r+1}^{r-1} \frac{\binom{|N|}{k}}{2^{|N|}} \tag{2.19}$$

Then, their policy begins with soliciting a minimum number of labels $R = \min_N \{|N| : \epsilon(N, r) \leq \delta\}$ for each task, and keeps doubling that amount until a majority larger than r is reached. When this happens, we know with confidence δ that the predictions are not the result of random chance. This is similar to the Kappa measure in medical research (Fanshawe et al., 2008).

In a similar way, (Parameswaran et al., 2012) explore the design of complex policies entirely based on the number of positive and negative labels received on each item. Besides

giving guarantees on the probability of a classification error when all the predictors share the same accuracy $p_j = \bar{p}$, they also propose an efficient algorithm to design a policy with a given expected budget $\mathbb{E}\{T\}$. In our discussion of Chapter 3, we explain how these *majority-based* policies are a subset of the active learning strategies we introduce in the next Section 2.3.3.

2.3.3 Active learning

A more structured approach to the problem of label collection is offered by the existing literature on *active learning* (Settles, 2010). Here, the goal is to improve the performance of a standard machine learning algorithm by carefully selecting the samples in its training set, in order to acquire more informative data. The active learning paradigm is iterative: we collect the first few samples, train a partial model, consider its current output, and then decide which samples to query next. Depending on the problem at hand and the specific data collection strategy we use, we may achieve the same predictive performance of a random training set, but with an exponentially smaller number of samples (Freund et al., 1997; Dasgupta, 2004).

Applying the active learning paradigm to our setting requires a few adaptations (Yan et al., 2011). First, we should restrict our focus to *pool-based* methods, a subset of active learning techniques that query labels on a fixed set of points, in our case the M items. Second, we must take into account the fact that the labels we collect from the predictors are noisy, whereas the standard assumption in active learning is that we can observe ground-truth values $\boldsymbol{y}$.

Along this line of reasoning, a few authors have studied the impact of active learning techniques when a *supervised* machine learning algorithm is trained on crowdsourced data. In particular, Lin et al. (2014) answer the question of whether it is more efficient to acquire a single label on most of the items or label a subset of them multiple times. Conversely, Nushi et al. (2016) investigate a scenario where we need to crowdsource the features of each item too, and thus we do not want to waste our budget on uninformative ones. Finally, Yan et al. (2011) and Nguyen et al. (2015) explore a different model of interaction that allows the learner to select individual predictors.

However, in our setting we do not have access to a feature space and the individual items are independent (see Section 2.1). Thus, the supervised learning paradigm is not appropriate, since collecting a new label on any item i does not necessarily bring more information on the other items $i' \neq i$. Moreover, our goal is to improve the classification accuracy on the pool of M items itself, and not to generalise over other unseen items. As a consequence, we need to turn to other, more general, label collection policies (Fu et al., 2013).

The first active learning policy that is relevant to our research is *uncertainty sampling*

(Lewis and Gale, 1994). According to this heuristic, the learner will always acquire new data on the items whose classification she is the least confident about. Depending on the particular application, the exact definition of uncertainty may vary: Welinder and Perona (2010) use the posterior probability of choosing the wrong class and show the efficiency of their system experimentally, while Khetan and Oh (2016) use the magnitude of the first eigenvector of X and prove its asymptotical optimality, though in a more complex model with heterogeneous items. For non-binary classification problems, a further measure of uncertainty is the entropy of the posterior distribution (Settles, 2010). We study the theoretical properties of the uncertainty sampling policy in Chapters 3 and 4.

Looking at the problem from a different angle, we can see that the active learner is training a new model after acquiring every subset of samples. We can exploit this fact and define a label collection policy that takes into account the impact of the new labels on the current model. Both Lizotte et al. (2003) and Nguyen et al. (2015) propose to target the probability of error, and always choose the label that reduces it the most in expectation. However, this strategy has never been studied on the basic Dawid-Skene model. We fill this research gap in Chapter 3, where we show that minimising the expected prediction error is a weaker form of uncertainty sampling.

In contrast, there are accounts of using the *information gain* of the incoming data to define an active learning policy in probabilistic settings (MacKay, 1992). Specifically, we can measure the information gain of a future probabilistic model $g(\bullet)$ with respect to the current one $f(\bullet)$ in terms of the Kullback-Leibler divergence between the two distributions (Kullback and Leibler, 1951):

$$KL(g||f) = \sum_{\boldsymbol{y}} g(\boldsymbol{y}) \log \left(\frac{g(\boldsymbol{y})}{f(\boldsymbol{y})} \right) \tag{2.20}$$

where we assume that the domain of the probabilistic model spans all the possible combinations of item classes $\boldsymbol{y}$. Since we cannot predict the values of the incoming labels, we use the expected information gain $\mathbb{E}_g\{KL(g||f)\}$ over all possible outcomes of our current query as an optimisation proxy.

In this vein, Simpson and Roberts (2014) employ the idea just introduced to create a label collection policy for crowdsourcing. However, the *intelligent tasking* algorithm they introduce suffers from a major drawback: without a closed form for the expected value of Equation 2.20, we need to simulate the arrival of all the possible future labels and retrain the probabilistic model on each of them. In Chapter 5 we comment on the high computational requirements of this policy on the full Dawid-Skene model. Conversely, in Chapters 3 and 4 we prove that this strategy is equivalent to uncertainty sampling when more information about the predictors' accuracy is available.

2.4 Summary

In this chapter we present the three building blocks of the multiple classifiers setting and the related literature. First, in Section 2.1 we review different models of this setting, and the related assumptions. There, we present the one-coin Dawid-Skene model as a reasonable compromise between expressiveness and simplicity. Second, in Section 2.2 we introduce several state-of-the-art techniques to aggregate the output of the individual predictors, and form accurate estimates of the item classes. These come from different fields, namely voting, Bayesian statistics and frequentist statistics. Third, we move to the question of collecting the data from the predictors in Section 2.3. The existing collection policies come both from the crowdsourcing community and, more in general, from the active learning literature.

Along the way, we show that this corpus of literature is fragmented. Not only are a large number of the relevant works grounded in different assumptions, but also we miss a structured comparison between alternative techniques, let alone a theoretical understanding of their properties and performance. This is particularly evident for the data collection problem, where the only theoretical result applies to a policy that has been proven suboptimal in practice. As a consequence, the pratictioner who plans to use one of these existing techniques may be unsure on which to select, while the researcher working on a generalisation of some of the classic approaches we present in this chapter may not have enough foundational knowledge to build upon.

The contributions we present in this thesis address this gap in the literature, and provide a structured study of both the data aggregation and the data collection problems in the multiple classifier setting. More specifically, in Chapter 3 we study the impact of all the different collection policies on the predictive accuracy of the system from a theoretical standpoint. This approach allows us to explain their behaviour and make a meaningful comparison among them. However, in order to do so, we are forced to use a idealised variant of the one-coin Dawid-Skene model. Thus, we spend most of Chapter 4 generalising our result to a more realistic setting. Thanks to this background, we are able to tackle the data aggregation problem in the original one-coin Dawid-Skene model in Chapter 5. There, we propose two new Bayesian inference methods which consistently beat the state-of-the-art in terms of predictive accuracy and computational speed. In addition to this, we are able to give theoretical guarantees on their accuracy, both in conjunction with non-adaptive collection policies and adaptive ones. The latter is the first instance of this kind in the current literature.

Chapter 3

Known predictors' accuracy

As we show in the previous chapter, the existing literature offers a number of different approaches to collect and aggregate the predictions of different classifiers. On the practical side, this variety is useful to those willing to implement a particular system, as they have a library of examples from which to take inspiration. On the theoretical side, comparing the existing results is challenging as they are rooted in different models and assumptions. As a consequence, we are left without an answer to why a specific technique works best in some cases, but may fail dramatically in others.

In this chapter, we take a first step towards addressing this question by reducing the aggregation problem to a bare minimum, and studying the impact of different collection policies. This includes considering binary classes only, and assuming that the accuracy of each individual predictor is known. We argue that this model, though too abstract for most practical purposes, still captures the essential properties of the problem at hand. Most importantly, it defines a common ground on which we can compare different collection policies in a meaningful way.

The work we present in this chapter is mainly theoretical. Primarily, we study the relationship between the number of data points R we collect on each item, and the probability of a classification error. We show that this relationship is exponential in the form $\mathbb{P}(error) = \exp(-cR)$, where $c > 0$ is a policy-dependent constant. Our goal is identifying the value of this constant for each collection policy, and using it to make a comparison between them.

Specifically, our contribution unfolds in the following way. After explaining the details of our binary model in Section 3.1, we divide the existing collection policies into two categories. The first is that of *non-adaptive* policies, which we explore in Section 3.2. These include all policies that make decisions based on the number and accuracy of the predictors, but ignore the data points they provide. The second is that of *adaptive* policies, which instead take the latter into account too. We study them in Section 3.3,

where we analyse their performance under ideal assumptions. In reality, we do not need these assumptions to hold in a strict sense, as we show in Section 3.4. With these results in mind, we can quantify the advantage that adaptive policies have over non-adaptive ones, as we do in Section 3.5. Finally, in Section 3.6 we draw our conclusions and explain how the results in the present chapter inform the content of the following ones.

3.1 An idealised model

We introduce here a model of classification for multiple items that we carry throughout the whole thesis. This is based on the celebrated Dawid-Skene model (Dawid and Skene, 1979) for aggregating multiple classifiers, that we briefly presented in Chapter 2. However, in this chapter we make several simplifying assumptions, and reduce the aforementioned model to an ideal setting where most of the parameters are known. In the following chapters we remove these assumptions one by one, and show how our results for the idealised model extend to more general settings.

Here we recap the basic properties of our setting. First, our objective is to recover the ground-truth class of a set of items M. For now, we assume that the ground-truth class $y_i \in \{\pm 1\}$ of each item i in M is binary, and that the prior probability on a positive class is $\mathbb{P}(y_i = +1) = q$. To achieve this objective we have a set of predictors N which provide us with data points, or *labels*, which we denote as $X = \{x_{ij}\}$, where i is the item index and j is the predictor index. To make our analysis more general, we assume that we have no control over the order in which the predictors label the items. Instead, we model their arrival with a vector $\boldsymbol{a}$, where $j = a(t)$ is the predictor available at time t. However, we assume that we can choose which (single) item $i \in M$ the available predictor $j = a(t)$ labels at time t. This job is performed according to a collection policy π. Finally, after T timesteps the label collection process concludes.

In the course of the following discussion, we use some additional notation. First, we denote with N_i the subset of predictors that labelled item i. Similarly, we use M_j to represent the subset of items labelled by predictor j. Second, since the collection process is inherently sequential, we use the superscript t to denote the information available up to each time step t. For example, we use X^t for the subset of data points observed so far.

A key feature of the Dawid-Skene model is that the predictors cast their labels independently. More formally, conditional on the ground-truth class y_i of the item they are labelling, the predictor's chance of producing a correct answer is $\mathbb{P}(x_{ij} = y_i) = p_j$. Oftentimes, the literature refers to this property as the *one-coin* assumption, as each predictor is modelled with a single Bernoulli parameter. In this chapter, we assume that the accuracy vector of the predictors $\boldsymbol{p} = (p_1, \ldots, p_{|N|})$ is known. Furthermore,

we assume that the values of the parameters p_j are extracted from an underlying common distribution f_p. All together, these assumptions form a full generative model, for which we give a graphical representation in Figure 3.1. There, we highlight in grey the variables we know the value of, and in white those we need to estimate.

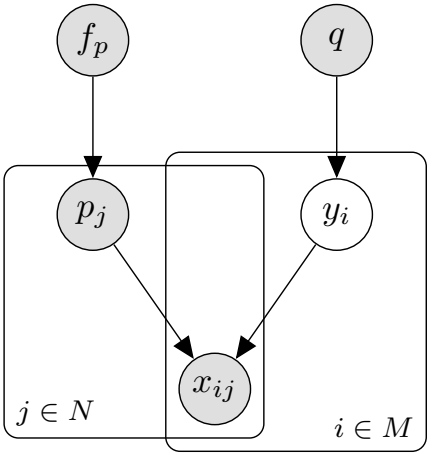


FIGURE 3.1: A graphical representation of the idealised Dawid-Skene model.

With this information, the problem of aggregating the data points X in a prediction over the class of the items reduces to weighted majority voting. In other terms, we can directly compute the posterior probability over y_i as follows:

$$\mathbb{P}(y_i = +1|X, \boldsymbol{p}, q) \propto q \prod_{j \in N_i} p_j^{\mathbb{I}(x_{ij}=+1)} (1 - p_j)^{\mathbb{I}(x_{ij}=-1)} \tag{3.1}$$

which translates to the optimal weighted majority voting rule (Nitzan and Paroush, 1982) when taken in the log-odds domain:

$$\begin{aligned} z_i &\equiv \log\left(\frac{\mathbb{P}(y_i = +1|X, \boldsymbol{p}, q)}{\mathbb{P}(y_i = -1|X, \boldsymbol{p}, q)}\right) \\ &= \log\left(\frac{q}{1-q}\right) + \sum_{j \in N_i} x_{ij} \log\left(\frac{p_j}{1-p_j}\right) \end{aligned} \tag{3.2}$$

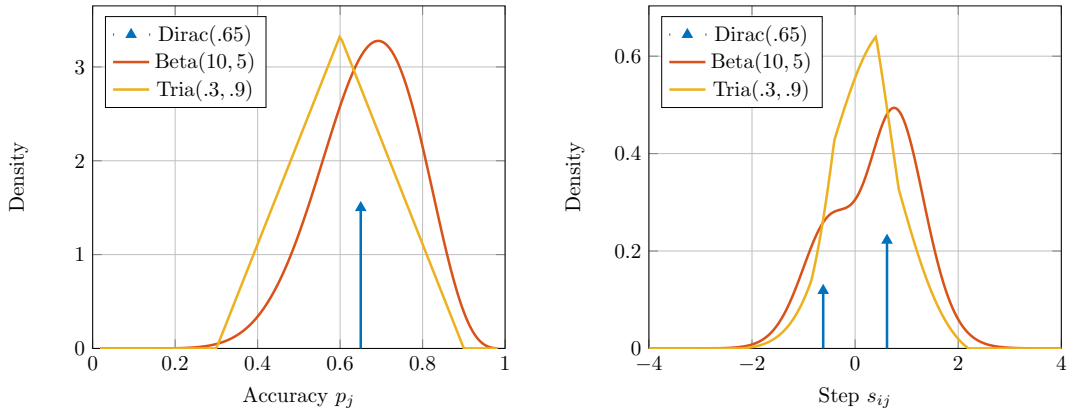
and where the final predictions are formed as $\hat{y}_i = \text{sign}(z_i)$ for all items $i \in M$. In the following discussion, we often refer to the log-odds of q and p_j in Equation 3.2 as *weights*, and we denote them as w_q and w_j respectively.

Our choice of assumptions for this chapter is not arbitrary. In fact, most of our results depend on the specific form Equation 3.2 takes. There, we can see that the log-odds z_i are the summation of a number of independent terms: a prior on the item class, and a set of weighted votes by the predictors. In this respect, we can interpret the evolution of z_i during the collection process as a *discrete random walk* that starts in $z_i^0 = w_q$, and moves with continuous-sized steps $s_{ij} \equiv x_{ij} w_j$ until it reaches its final value. We use this interpretation throughout the whole chapter.

An important property of the steps s_{ij} is that on average they always points towards the ground-truth class y_i. More formally, we can write the expected value of s_{ij} as follows:

$$\begin{aligned}\mathbb{E}_{x_{ij},p_j|y_i}(x_{ij}w_j) &= \mathbb{E}_{p_j}\big(\mathbb{E}_{x_{ij}|p_j,y_i}(x_{ij})w_j\big) \\ &= y_i\mathbb{E}_{p_j}\big((2p_j-1)w_j\big)\end{aligned} \tag{3.3}$$

where the last expectation is either positive or zero (when all accuracies equal $p_j = \frac{1}{2}$). As a consequence, the expected value of s_{ij} will always have the same sign as y_i. Furthermore, its magnitude depends solely on the probability density function of the predictors' accuracy $p_j \sim f_p$. To give an intuitive understanding of this relationship, we plot three examples in Figure 3.2, under the assumption that $y_i = +1$. Note how the density of the probability function is skewed towards the positive side in all three cases.



(a) Predictors' accuracy domain.

(b) Weighted vote domain.

FIGURE 3.2: Three examples of the predictors' population.

As a consequence, the random walk on z_i always exhibits a drift toward the correct class. However, in our following analysis we show that different collection policies capitalise on this drift at different rates. We present a quantitative comparison between such rates in Section 3.5.

3.2 Non-adaptive collection policies

In this section we present our analysis of *non-adaptive* policies. We put in this category all the collection policies that base their decisions solely on the number and accuracy of the predictors. More formally, a non-adaptive policy decides which subset of predictors N_i to allocate to each item i based on the following two pieces of information: the total number of items $|M_j|$ each predictor j can label and the accuracy $\boldsymbol{p} = (p_1, \dots, p_N)$ of the predictors. In general this information must be known in advance of the collection process. However, all the policies we study in this section can be implemented in an online fashion, where the availability vector $\boldsymbol{a}$ is revealed one time step at a time.

Given this definition, all non-adaptive policies satisfy the following property: the probability of a classification error on item i depends only on the corresponding subset N_i of predictors, and their accuracy. In other terms, once the collection policy has selected N_i, the predictors therein cast their labels x_{ij} independently. This is in contrast with the *adaptive* policies we study in Section 3.3, which populate the subset N_i one predictor at a time, depending on the labels collected so far.

This property allows us to study the probability of a classification error on each item i separately. Accordingly, we first derive some results with a single item $|M| = 1$ in Section 3.2.1, and compare them with the existing literature. Then, we use them to assess the performance of the UNI policy in Section 3.2.2, and the WB policy in Section 3.2.3.

3.2.1 Classification error on a single item

Before dealing with the behaviour of a specific non-adaptive collection policy, let us derive some general bounds on the probability of misclassifying a single item. The results we present in this section are not only useful for our discussion in the following Sections 3.2.2 and 3.2.3, but constitute an improvement on the existing error bounds for the weighted majority voting rule in (Berend and Kontorovich, 2015) and (Gao et al., 2016) as we explain below. Throughout this whole section we drop the index i, since we have only a single item to classify, and assume that we observe a label x_j from each predictors j in N. Moreover, we assume that the cardinality of N, i.e. the number of predictors, is independent from the value of the data X we observe, in accordance with the definition of a non-adaptive policy. We will drop this assumption in Section 3.3.1 where we analyse the adaptive case instead.

With this in mind, we can bound the probability of a classification error from above, conditional on the quality of the predictors and the prior on the class of the item:

Theorem 3.1. *Given a set N of predictors with known accuracies $\boldsymbol{p} = (p_1, \dots, p_{|N|})$ and prior on the positive class $q \equiv \mathbb{P}(y = +1)$, the probability of a classification error under weighted majority voting is upper bounded by:*

$$\mathbb{P}(\hat{y} \neq y | \boldsymbol{p}, q) \leq \sqrt{q(1-q)} \prod_{j \in N} 2\sqrt{p_j(1-p_j)} \tag{3.4}$$

Proof. By definition the probability of a classification error is:

$$\mathbb{P}(\hat{y} \neq y | \boldsymbol{p}, q) = q\mathbb{P}(\hat{y} \neq y | \boldsymbol{p}, y = +1) + (1-q)\mathbb{P}(\hat{y} \neq y | \boldsymbol{p}, y = -1) \tag{3.5}$$

Now, let us define the halved log-odds as $h \equiv \frac{1}{2}w_q + \sum_{j \in N} x_j \frac{1}{2} w_j$, and rewrite the conditional probabilities in Equation 3.5 by marginalising over the output X of the

individual predictors:

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, y = +1) = \sum_X \Big(\mathbb{I}(h < 0) + \frac{1}{2}\mathbb{I}(h = 0)\Big)\mathbb{P}(X|\boldsymbol{p}, y = +1) \tag{3.6}$$

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, y = -1) = \sum_X \Big(\mathbb{I}(h > 0) + \frac{1}{2}\mathbb{I}(h = 0)\Big)\mathbb{P}(X|\boldsymbol{p}, y = -1) \tag{3.7}$$

Here, the probability of observing each $x_j \in X$ is independent of the output of the other predictors, conditional on the ground-truth y. Therefore, we have that:

$$\begin{aligned}\mathbb{P}(X|\boldsymbol{p}, y = +1) &= \prod_{j \in N} \mathbb{P}(x_j|p_j, y = +1) \\ &= \prod_{j \in N} \frac{\mathbb{P}(x_j|p_j, y = +1)\exp(-x_j\frac{1}{2}w_j)\sqrt{p_j(1-p_j)}}{\sqrt{p_j(1-p_j)}\exp(-x_j\frac{1}{2}w_j)} \\ &= \exp(h - \frac{1}{2}w_q)\prod_{j \in N}\sqrt{p_j(1-p_j)}\end{aligned} \tag{3.8}$$

since $\mathbb{P}(x_j|p_j, y)$ is either p_j or $1 - p_j$, and the following holds:

$$\frac{\mathbb{P}(x_j|p_j, y = +1)\exp(-x_j\frac{1}{2}w_j)}{\sqrt{p_j(1-p_j)}} = 1 \qquad \text{for all} \qquad x_j \in \{\pm 1\} \tag{3.9}$$

Similarly, for the opposite ground-truth class we have:

$$\mathbb{P}(X|\boldsymbol{p}, y = -1) = \exp(-h + \frac{1}{2}w_q)\prod_{j \in N}\sqrt{p_j(1-p_j)} \tag{3.10}$$

By substituting Equations 3.8 and 3.10 in Equations 3.6 and 3.7 we get:

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, y = +1) \leq \sum_X \Big(\mathbb{I}(h < 0) + \frac{1}{2}\mathbb{I}(h = 0)\Big)\exp(-\frac{1}{2}w_q)\prod_{j \in N}\sqrt{p_j(1-p_j)} \tag{3.11}$$

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, y = -1) \leq \sum_X \Big(\mathbb{I}(h > 0) + \frac{1}{2}\mathbb{I}(h = 0)\Big)\exp(+\frac{1}{2}w_q)\prod_{j \in N}\sqrt{p_j(1-p_j)} \tag{3.12}$$

where the inequality signs comes from the fact that $\exp(h) \leq 1$ and $\exp(-h) \leq 1$ for all $h \leq 0$ and $h \geq 0$ respectively.

Finally, both the sum in Equation 3.11 and 3.12 amount to $2^{|N|-1}$, as for each h there exists an h' such that $h' = -h$, and the number of possible values of X is $2^{|N|}$. By combining these results according to Equation 3.5 we get the bound in the theorem. □

Two things must be noted. First, the prior q takes the role of an extra predictor in the bound of Theorem 3.1, reducing the chance of an error by $\sqrt{q(1-q)}$ instead of $\sqrt{p_j(1-p_j)}$. Second, the existing literature provides two alternative bounds in the special case $q = \frac{1}{2}$. In the following two remarks, we show that our result is tighter than both of them:

Remark 3.2. Theorem 3.1 is tighter than the corresponding result in (Gao et al., 2016)[1]:

$$\mathbb{P}\Big(\hat{y} \neq y|\boldsymbol{p}, q = \frac{1}{2}\Big) \leq \prod_{j \in N} 2\sqrt{p_j(1-p_j)} \tag{3.13}$$

Proof. Substituting $q = \frac{1}{2}$ in Equation 3.4 yields a factor of $\frac{1}{2}$ in front of the product. The corresponding bound in Equation 3.13 lacks this factor. □

Remark 3.3. Theorem 3.1 is tighter than the corresponding result in (Berend and Kontorovich, 2015):

$$\mathbb{P}\Big(\hat{y} \neq y|\boldsymbol{p}, q = \frac{1}{2}\Big) \leq \exp\Big(-\frac{1}{2}\sum_{j \in N}\big(p_j - \frac{1}{2}\big)w_j\Big) \tag{3.14}$$

Proof. By taking the logarithm of our result in Theorem 3.1, and setting $q = \frac{1}{2}$, we have:

$$\mathbb{P}\Big(\hat{y} \neq y|\boldsymbol{p}, q = \frac{1}{2}\Big) \leq \exp\Big(-\log(2) + \frac{1}{2}\sum_{j \in N}\log\big(4p_j(1-p_j)\big)\Big) \tag{3.15}$$

where for any p_j the function $\log\big(4p_j(1-p_j)\big)$ is never larger than the corresponding function $\big(\frac{1}{2} - p_j\big)w_j$ in the argument of Equation 3.14. Therefore, for any $\boldsymbol{p}$ the bound in Theorem 3.1 always produces estimates that are closer to the true probability of an error. □

Let us move now to our lower bound on the probability of a classification error:

Theorem 3.4. *Given a set N of predictors with known accuracies $\boldsymbol{p} = (p_1, \ldots, p_{|N|})$ and prior on the positive class $q \equiv \mathbb{P}(y = +1)$, the probability of a classification error under weighted majority voting is lower bounded by:*

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, q) \geq 0.73\exp\Big(-\frac{1}{2}\|\boldsymbol{w}\|_2\Big)\sqrt{q(1-q)}\prod_{j \in N} 2\sqrt{p_j(1-p_j)} \tag{3.16}$$

where $\boldsymbol{w}$ is the vector of weights $w_j = \log\big(p_j/(1-p_j)\big)$, and $\|\bullet\|_2$ denotes the Euclidean norm.

Proof. Here we can reuse some of the arguments from the proof of Theorem 3.1. Specifically, Equations 3.6, 3.7, 3.8 and 3.10 allow us to derive the following:

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, y = +1) = S\exp(-\frac{1}{2}w_q)\prod_{j \in N} 2\sqrt{p_j(1-p_j)} \tag{3.17}$$

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, y = -1) = S\exp(+\frac{1}{2}w_q)\prod_{j \in N} 2\sqrt{p_j(1-p_j)} \tag{3.18}$$

[1]The original formulation of this result is not straightforward. Refer to Appendix A for clarifications.

where we define the shared term S as:

$$S \equiv \frac{1}{2^{|N|}} \sum_{X} \Big(\mathbb{I}(h < 0) + \frac{1}{2}\mathbb{I}(h = 0) \Big) \exp(h) \tag{3.19}$$

Then, we notice that S does not depend on the predictor accuracy $\boldsymbol{p}$ except for the weights $\boldsymbol{w}$. In this regard, we can treat X as the observation of $|N|$ independent Rademacher variables $\boldsymbol{\epsilon}$ with corresponding probability $2^{-|N|}$. The following upper bound follows suit:

$$\begin{aligned} S &\geq \frac{1}{2^{|N|}} \sum_{X} \Big(\mathbb{I}(h < 0) + \frac{1}{2}\mathbb{I}(h = 0) - \mathbb{I}(h < -H) \Big) \exp(-H) \\ &= \left(\frac{1}{2} - \frac{1}{2^{|N|}} \sum_{X} \mathbb{I}(h < -H) \right) \exp(-H) \\ &= \Big(\frac{1}{2} - \mathbb{P}_{\boldsymbol{\epsilon}}(h < -H) \Big) \exp(-H) \end{aligned} \tag{3.20}$$

where $H \geq 0$ is an arbitrary constant, and the probability of h exceeding $-H$ when the observations X are Rademacher-distributed can be bounded with Hoeffding's inequality:

$$\mathbb{P}_{\boldsymbol{\epsilon}}(h < -H) \leq \exp\left(- \frac{8H^2}{\sum_{j \in N} w_j^2} \right) \tag{3.21}$$

Now, since the result in Equation 3.20 is valid for any $H \geq 0$, let us set $H = \frac{1}{2}\sqrt{\sum_{j \in N} w_j^2}$. This choice yields $\frac{1}{2} - \mathbb{P}_{\boldsymbol{\epsilon}}(h < -H) \geq 0.5 - \exp(-2) \approx 0.365$. By combining these results according to Equation 3.5 we get the bound in the theorem. □

Note that the lower bound in Theorem 3.4 introduces an additional term with respect to the upper bound in Theorem 3.1. Despite this, our result is still tighter than the bounds in the existing literature. Specifically, for $q = \frac{1}{2}$ we have:

Remark 3.5. Theorem 3.4 is tighter than the corresponding result in (Gao et al., 2016)[2]:

$$\mathbb{P}\Big(\hat{y} \neq y | \boldsymbol{p}, q = \frac{1}{2} \Big) \geq 0.25 \exp\big(- \|\boldsymbol{w}\|_2 \big) \prod_{j \in N} 2\sqrt{p_j(1 - p_j)} \tag{3.22}$$

Proof. Substituting $q = \frac{1}{2}$ in Equation 3.16 yields a factor of 0.365 which is larger than the factor 0.25 in Equation 3.22. More importantly, our bound in Equation 3.16 has an additional factor $\frac{1}{2}$ before the term $\|\boldsymbol{w}\|_2$, which makes it exponentially tighter. □

Remark 3.6. Theorem 3.4 is asymptotically tighter than the corresponding result in (Berend and Kontorovich, 2015):

$$\mathbb{P}\Big(\hat{y} \neq y | \boldsymbol{p}, q = \frac{1}{2} \Big) \geq \frac{3}{4\big(1 + \exp(2\Phi + 4\sqrt{\Phi})\big)} \quad \text{where} \quad \Phi = \sum_{j \in N} \Big(p_j - \frac{1}{2} \Big) w_j \tag{3.23}$$

[2] Again, refer to Appendix A for clarifications on the original formulation of this result.

Proof. Assume that the number of predictors is large, i.e. $|N| \to \infty$, and that their accuracy is non-random, i.e. $p_j \neq \frac{1}{2}$ for most j. Then, $\Phi \to \infty$ and the non-asymptotic terms in both Equation 3.16 and 3.23 become negligible. As a consequence, the statement we need to prove reduces to the following:

$$\prod_{j \in N} 2\sqrt{p_j(1-p_j)} \geq \exp(-2\Phi) \tag{3.24}$$

or alternatively in logarithm form:

$$\frac{1}{2} \sum_{j \in N} \log\left(4p_j(1-p_j)\right) \geq \sum_{j \in N} (1-2p_j)w_j \tag{3.25}$$

which holds for any $\boldsymbol{p}$, since the left-hand function is not smaller than the right-hand one for any $p_j \in (0,1)$, with equality for $p_j = \frac{1}{2}$. □

For a qualitative comparison between our results in Theorems 3.1, 3.4 and the existing bounds in the literature, refer to Figure 3.3 and its related discussion.

3.2.2 Classification error under the UNI policy

The results in Section 3.2.1 allow us to study the performance of the uniform (UNI) collection policy we introduce in Section 2.3.1. The idea behind this policy is simple: given the total amount of data points T we can afford, split them equally amongst the set of items we are trying to classify. In the following discussion, we assume a round-robin implementation of the UNI policy. That is, at every time step t we assign the currently available predictor $j = a(t)$ to the next item $i = \pi(t-1)+1$. In order to avoid assigning a predictor to the same item twice, we formally define the UNI policy as follows:

$$\pi_{uni}(t) = \operatorname*{argmin}_{i \in M \setminus M_{a(t)}^{t-1}} \left(|N_i^{t-1}|\right) \tag{3.26}$$

where $M_{a(t)}^{t-1}$ is the set of items labelled by predictor $j = a(t)$ in the past, which we need to skip, and ties are resolved in lexicographic order.

Overall, the UNI policy tries to assign $R = T/|M|$ predictors to each item, rounded to the nearest integer. The round-robin implementation guarantees this behaviour happens in most cases. In fact, the UNI policy is robust to several non-ideal conditions during the collection process, as we detail in Section 3.4. The only exception being when the number of predictors $|N|$ is smaller than R, and thus it is impossible to collect enough independent data points on each item. However, there exists no policy that can overcome such a degenerate scenario.

With this in mind, we can extend the results in Section 3.2.1 to the UNI policy. Recall,

that Theorems 3.1 and 3.4 bound the probability of a classification error on a single item given a set of predictors N. In the case of the UNI policy, these predictors are spread over the whole set of items M, and each item i receives only a subset $N_i \subseteq N$, with cardinality $|N_i| = R$. Crucially, the decisions on how to distribute the set of predictors N over the overlapping subsets N_i do not depend on the accuracy $\boldsymbol{p}$ of the predictors (see Equation 3.26). As a consequence, we can study the probability of a classification error on each item i in isolation, and assume that the accuracy of its corresponding subset of predictors N_i is extracted from the underlying probability density function $p_j \sim f_p$.

More specifically, we can tie the probability of a classification error under the UNI policy to the number of predictors per item R and the predictors' accuracy density f_p as follows:

Corollary 3.7. *Given a population of predictors with accuracy $p_j \sim f_p$, a prior on the positive class $q \equiv \mathbb{P}(y_i = +1)$, and R predictors per item, the probability of a classification error by the UNI policy under weighted majority voting is upper bounded by:*

$$\mathbb{P}(\hat{y}_i \neq y_i | f_p, q) \leq \sqrt{q(1-q)}\Big(2\mathbb{E}_{p_j \sim f_p}\big(\sqrt{p_j(1-p_j)}\big)\Big)^R \tag{3.27}$$

and lower bounded by:

$$\mathbb{P}(\hat{y}_i \neq y_i | f_p, q) \geq 0.73\sqrt{q(1-q)}\mathbb{E}_{\boldsymbol{p}' \sim f_p}\Big(\exp\big(-\frac{1}{2}\|\boldsymbol{w}'\|_2\big)\prod_{j=1}^{R} 2\sqrt{p_j(1-p_j)}\Big) \tag{3.28}$$

for any item $i \in M$, where $\boldsymbol{p}'$ is a vector of length R and $\boldsymbol{w}'$ contains its corresponding weights $w_j = \log\big(p_j/(1-p_j)\big)$.

Proof. Theorems 3.1 and 3.4 provide us with bounds on the probability of a classification error conditioned on the predictors' accuracy $\boldsymbol{p}$. Using the chain rule we can derive the probability of an error for any $\boldsymbol{p} \sim f_p$ as follows:

$$\mathbb{P}(\hat{y}_i \neq y_i | f_p, q) = \mathbb{E}_{\boldsymbol{p} \sim f_p}\big(\mathbb{P}(\hat{y}_i \neq y_i | \boldsymbol{p}, q)\big) \tag{3.29}$$

The results in the corollary follow from the fact that each p_j is i.i.d. □

Comparing the two bounds in Corollary 3.7, we can see that they both share the common terms $\sqrt{q(1-q)}$ and $2\sqrt{p_j(1-p_j)}$. At the same time, the dependency of the lower bound on the Euclidean norm of $\boldsymbol{w}'$ can give the impression that the two bounds do not match asymptotically. However, we can show that this is not the case. In fact, as the number of predictors $R \to \infty$ grows large, the term $\|\boldsymbol{w}'\|_2$ has a contribution of $c\sqrt{R}$ on the probability of an error, for some constant c. Conversely, the term $\sqrt{p_j(1-p_j)}$ has a contribution of $c'R$, for some other constant c', which quickly dominates the former. Since both upper and lower bound share the latter term, we can conclude that they match asymptotically.

As a sidenote, the results in Corollary 3.7 are only the last of a sequence of incremental improvements in our research. We list our previous bounds in Appendix B. While these bounds are less tight, the technique we use to derive them is more general, and as such it can be of separate interest.

We conclude this section with a qualitative assessment on how close the bounds in Corollary 3.7 are to the true probability of a classification error. To this end, we generate synthetic data and compute the empirical performance of the UNI policy. In the following experiment we set the number of items to $|M| = 1000$, the prior on the items to $q = \frac{1}{2}$, the number of labels provided by each predictor to $L = 1$, and we extract the predictors' accuracy according to $p_j \sim \text{Beta}(4, 3)$. This represents a scenario with a mixed population of predictors whose average accuracy is not too distant from $\frac{1}{2}$. Similar results can be obtained for different choices of f_p. The results are presented in Figure 3.3 for an increasing number of labels per item R.

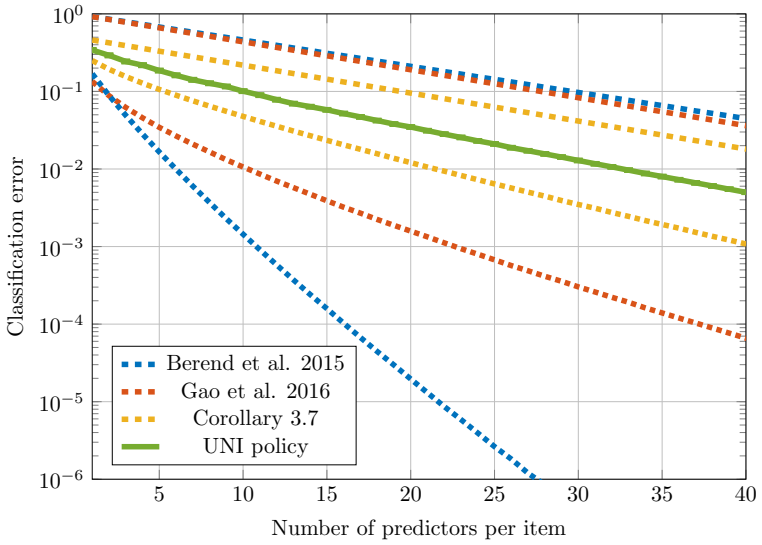


FIGURE 3.3: Comparison between the empirical classification error under the UNI policy, the bounds in the existing literature and our results. Dashed lines are the upper bounds, dotted lines are the lower bounds.

As expected, the probability of a classification error decreases as we observe more data points. In this regard, notice how the bounds in Corollary 3.7 closely match the empirical performance as R increases. This is in contrast with both the existing works of (Gao et al., 2016), which suffers from more conservative factors, and (Berend and Kontorovich, 2015), whose non-matching lower bound rapidly diverges away from the true empirical performance. For a more rigorous comparison of our results with these existing works, see Section 3.2.1. For a comparison with the performance of other collection policies, see Section 3.5.

3.2.3 Classification error under the WB policy

One of the drawbacks of the UNI policy is that it does not take into account the accuracy of the individual predictors during the collection process. Without this distinction, the UNI policy may collect a set of very noisy data points on some items, while assigning all the accurate predictors to some others. When the population of predictors is varied, this behaviour can be very inefficient.

In this section we investigate the properties of an *optimal* non-adaptive policy. First, we provide an asymptotical expression for its probability of a classification error. Second, we propose an efficient policy to approximate its behaviour. We call the latter the *weight balancing* (WB) policy (see Section 2.3.1), and we show that it has indeed a better performance than the UNI policy, albeit only marginally.

Let us begin with the probability of a classification error of an optimal non-adaptive collection policy:

Corollary 3.8. *Given a population of predictors with accuracy $p_j \sim f_p$, a prior on the item classes $q \equiv \mathbb{P}(y_i = +1)$, and a large number of predictors per item $R \to \infty$, the probability of a classification error by an optimal non-adaptive policy under weighted majority voting is:*

$$\mathbb{P}(\hat{y}_i \neq y_i | f_p, q) = \exp\left(\frac{R}{2}\mathbb{E}_{p_j \sim f_p}\Big(\log\big(4p_j(1-p_j)\big)\Big) + o\big(\sqrt{R}\big)\right) \tag{3.30}$$

for any item $i \in M$.

Proof. Let us focus on the asymptotic case where $R \to \infty$. Rewriting the bounds in Theorems 3.1 and 3.4 in exponential form for any item $i \in M$ we have:

$$\mathbb{P}(\hat{y} \neq y | \boldsymbol{p}, q) = \exp\Big(\frac{1}{2}\sum_{j \in N_i} \log\big(4p_j(1-p_j)\big) + o\big(\sqrt{|N_i|}\big)\Big) \tag{3.31}$$

where $o\big(\sqrt{|N_i|}\big)$ includes both the logarithm of the constant $\sqrt{q(1-q)}$ and the term $-\frac{1}{2}\sqrt{\sum_{j \in N_i} w_j^2}$ in the lower bound, as explained in the discussion of Theorem 3.7.

Three things must be noted. First, the dominant term in Equation 3.31 is the sum over $\log(4p_j(1-p_j))$. Second, the exponential is a convex function. Third, the decisions of any collection policy affect which subset of predictors N_i gets assigned to each item $i \in M$. Therefore, if we want to minimise the probability of a classification error across all the items M, we need to cleverly distribute the predictors in such a way that the sum in Equation 3.31 has the same value for each $i \in M$. The total amount that we can distribute after the whole collection process is the following:

$$S_{tot} = \frac{1}{2}\sum_{j \in N} L \log\big(p_j(1-p_j)\big) \tag{3.32}$$

where we assume that each predictor labels the same number of items $|M_j| = L$. Furthermore, for $R \to \infty$ the number of predictors $|N|$ must also become large. As a consequence we can rewrite Equation 3.32 taking the expectation over p_j as follows:

$$S_{tot} = \frac{L|N|}{2} \mathbb{E}_{p_j \sim f_p} \Big(\log \big(4p_j(1-p_j) \big) \Big) \tag{3.33}$$

where $L|N| = T$ is the total number of data points we collect.

In the best-case scenario, an optimal non-adaptive policy is able to split S_{tot} evenly over the M items. If this is the case, $L|N|/|M| = R$ and we get the result in the corollary. □

Note that the result in Corollary 3.8 differs from the corresponding results for the UNI policy (see Corollary 3.7) in that the expectation over p_j is outside of the logarithm. By Jensen's inequality we have:

$$\mathbb{E}_{p_j \sim f_p} \Big(\log \big(2\sqrt{p_j(1-p_j)} \big) \Big) \leq \log \Big(\mathbb{E}_{p_j \sim f_p} \big(2\sqrt{p_j(1-p_j)} \big) \Big) \tag{3.34}$$

thus establishing that the probability of a classification error under the UNI policy, which depends on the right-hand side term, is never smaller than that of an optimal non-adaptive policy (left-hand side term).

We show in Corollary 3.8 that such an optimal policy aims to distribute the quantity S_{tot} as evenly as possible across the set of items M. We can approximate this behaviour by greedily assigning the next predictor $j = a(t)$ to the item with the largest probability of a classification error according to Equation 3.31. This approach is akin to the greedy load-balancing algorithm in (Graham, 1966). We call the resulting collection strategy the weight balancing (WB) policy, which we formally define in the following way:

$$\pi_{wb}(t) = \underset{i \in M \setminus M_{a(t)}^{t-1}}{\operatorname{argmax}} \Big(\sum_{j \in N_i^{t-1}} \log \big(4p_j(1-p_j) \big) \Big) \tag{3.35}$$

We provide a qualitative assessment of this policy in Figure 3.4, where we compare it to the UNI policy under the same settings of Figure 3.3. Note that the improvement in empirical performance is minimal. Similarly, the corresponding bound in Corollary 3.8 shows only a slightly better asymptotic decay than the bound in Corollary 3.7 for the UNI policy. Regardless, we use the stronger result in Corollary 3.8 to compare adaptive and non-adaptive policies in Section 3.5.

3.3 Adaptive collection policies

The policies we analyse in Section 3.2 are designed around a notion of resource sharing. In the case of the UNI policy the resource is the total number of data points T, whereas

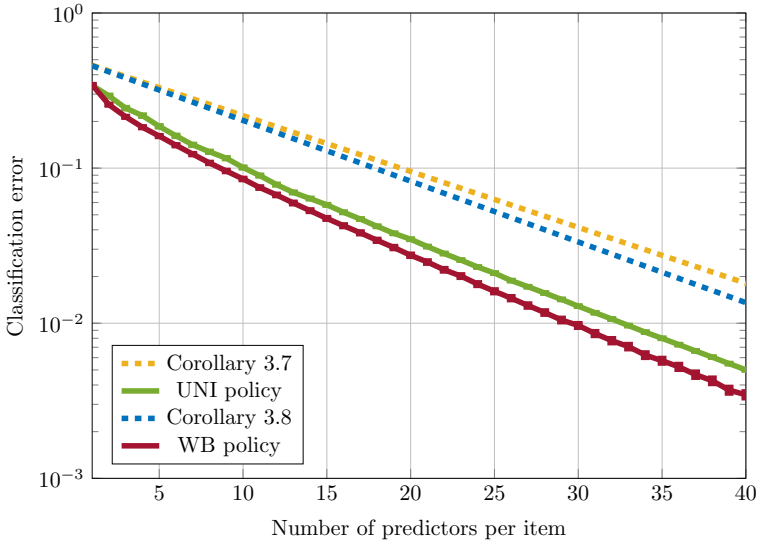


FIGURE 3.4: Comparison between the empirical classification error under the WB and UNI policies and their corresponding upper bounds.

for the WB policy it is the total predictive power of the available set of predictors N. However, using a non-adaptive policy means ignoring another potential source of information, that is the *actual data points* we collect. As a result, we may end up collecting too many labels on some items where our predictions are already very accurate, and neglecting others where more labels are needed to discriminate the ground-truth class effectively.

In contrast, the decisions of an *adaptive* policy also depend on the data points we collect. Since the value of the labels x_{ij} are revealed one by one as the collection process progresses, adaptive policies are sequential by their very nature. More formally, at any time step t an adaptive policy chooses the next item i to label based on the following pieces of information: the set of data points X^{t-1} observed so far, the current subset of items M_j^t labelled by each predictor j, and their accuracy $\boldsymbol{p} = (p_1, \ldots, p_N)$. In the following discussion, this information is usually summarised in terms of the current state of the log-odds z_i on each item i, and the characteristics of the incoming predictor $j = a(t)$.

Before dealing with the individual collection policies, we introduce some general results for a single item $|M| = 1$ in Section 3.3.1. These results are crucial in our analysis of the US policy, which we cover in Section 3.3.2. Then, in Section 3.3.3 we show that the IG policy, despite optimising a different objective function, is equivalent to the US policy. Finally, we dedicate Section 3.3.4 to the LOS policy, and discuss why it is inferior to the previous two.

3.3.1 Achieving a target classification error on a single item

Adaptive policies introduce additional complexity to our analysis, since the decision of assigning an individual predictor j to a specific item i depends on all the information X observed in the past. Thus, before dealing with the complexity of each collection policy as we do in Sections 3.3.2 and 3.3.3, let us derive some general results with a single item. This is the same approach we take in Section 3.2.1 for non-adaptive policies. However, in this section we reverse our argument: that is, we fix a target probability p_e for the classification error, and we compute the number of predictors $n \equiv |N|$ we need to reach it. To this end, we rely on the assumptions that the predictors' accuracy is i.i.d. according to an underlying probability density function $p_j \sim f_p$.

With this in mind, we can put an upper bound on the expected number of predictors we need as follows:

Theorem 3.9. *Given a population of predictors with accuracy $p_j \sim f_p$, and prior on the positive class $q \equiv \mathbb{P}(y = +1)$, the expected number $n \equiv |N|$ of predictors we need to lower the chance of a classification error below p_e is upper bounded by:*

$$\mathbb{E}_{X,\boldsymbol{p},y|q,p_e}(n) < \frac{-w_e + (1-2q)w_q}{\mathbb{E}_{p_j \sim f_p}\big((2p_j - 1)w_j\big)} + 1 \tag{3.36}$$

where w_e, w_q and w_j are the log-odds of p_e, q and p_j respectively.

Proof. Let us define $z^0 = w_q$ as the starting point of the random walk in the log-odds domain, and z^n as its final value, after observing the output of $n \equiv |N|$ predictors. By definition, we have $z^n \notin (w_e, -w_e)$, as this ensures a chance of a classification error $\mathbb{P}(\hat{y} \neq y) = \text{sig}(-|z^n|)$ smaller than $\text{sig}(-|w_e|) = p_e$. Notice that n is a *stopping time*, since our decision to stop collecting additional data depends only on the information collected up to time n.

The expected number of predictors n we need to cross the threshold w_e on either the positive or negative side depends on the ground-truth y as follows:

$$\mathbb{E}_{X,\boldsymbol{p},y|q,p_e}(n) = q\mathbb{E}_{X,\boldsymbol{p}|y=+1,p_e}(n) + (1-q)\mathbb{E}_{X,\boldsymbol{p}|y=-1,p_e}(n) \tag{3.37}$$

At the same time, there is a connection between the value of n and the average drift in the random walk caused by each step $s_j = x_j w_j$. Specifically, since n is a stopping time we can apply Wald's equation (Wald, 1944) to our case:

$$\mathbb{E}_{X,\boldsymbol{p}|y,p_e}(z^n) = \mathbb{E}_{X,\boldsymbol{p}|y,p_e}(n)\mathbb{E}_{x_j,p_j \sim f_p|y}(s_j) + w_q \tag{3.38}$$

Furthermore, we can link the expected value of z^n with the threshold w_e. To do this,

we recall that each step s_j in the random walk is independent, and thus:

$$\mathbb{E}_{X,\boldsymbol{p}|y,p_e}(z^n) = \mathbb{E}_{X,\boldsymbol{p}|y,p_e}(z^{n-1}) + \mathbb{E}_{x_j,p_j\sim f_p|y}(s_j) \tag{3.39}$$

Additionally, we know that the value of z^{n-1} falls in the interval $(w_e, -w_e)$ by definition. Therefore, we can use Equation 3.39 to establish the following inequalities:

$$\mathbb{E}_{X,\boldsymbol{p}|y=+1,p_e}(z^n) < -w_e + \mathbb{E}_{x_j,p_j\sim f_p|y=+1}(s_j) \tag{3.40}$$

$$\mathbb{E}_{X,\boldsymbol{p}|y=-1,p_e}(z^n) > w_e + \mathbb{E}_{x_j,p_j\sim f_p|y=-1}(s_j) \tag{3.41}$$

Now, we can derive a bound on the expected value of n. First, we combine Equations 3.38, 3.40 and 3.41 as follows:

$$\mathbb{E}_{X,\boldsymbol{p}|y=+1,p_e}(n)\mathbb{E}_{x_j,p_j\sim f_p|y=+1}(s_j) < -w_e + \mathbb{E}_{x_j,p_j\sim f_p|y=+1}(s_j) - w_q \tag{3.42}$$

$$\mathbb{E}_{X,\boldsymbol{p}|y=-1,p_e}(n)\mathbb{E}_{x_j,p_j\sim f_p|y=-1}(s_j) > w_e + \mathbb{E}_{x_j,p_j\sim f_p|y=-1}(s_j) - w_q \tag{3.43}$$

Then, we notice that the expected value of s_j when $y = -1$ has the same magnitude but opposite sign than the expected value of s_j when $y = -1$. Thus, we can rewrite Equation 3.37 as:

$$\mathbb{E}_{X,\boldsymbol{p},y|q,p_e}(n) < \frac{-w_e + \mathbb{E}_{x_j,p_j\sim f_p|y=+1}(s_j) - w_q(2q-1)}{\mathbb{E}_{x_j,p_j\sim f_p|y=+1}(s_j)} \tag{3.44}$$

Finally, we compute the expected value of the step $s_j = x_j w_j$ in terms of the predictor's accuracy p_j only, since $\mathbb{E}_{x_j|p_j,y=+1}(s_j) = (2p_j - 1)w_j$, which leads to the result in the theorem. □

Likewise, we can bound the expected number of predictors from below:

Theorem 3.10. *Given a population of predictors with accuracy $p_j \sim f_p$, and prior on the positive class $q \equiv \mathbb{P}(y = +1)$, the expected number $n \equiv |N|$ of predictors we need to lower the chance of a classification error below p_e is lower bounded by:*

$$\mathbb{E}_{X,\boldsymbol{p},y|q,p_e}(n) \geq \frac{-w_e + (1-2q)w_q - 0.56 - \mathbb{E}_{p_j\sim f_p}\big(|w_j|\big)}{\mathbb{E}_{p_j\sim f_p}\big((2p_j-1)w_j\big)} \tag{3.45}$$

where w_e, w_q and w_j are the log-odds of p_e, q and p_j respectively.

Proof. We reuse here some of the arguments and the notation from the proof of Theorem 3.9. Let us begin with establishing a lower bound on the expected number of predictors n when the ground-truth is $y = +1$.

By definition, at the end of the collection process the absolute value of the log-odds z^n

is larger than or equal to the threshold $-w_e$. More formally:

$$\begin{aligned} -w_e &\leq \mathbb{E}_{X,\boldsymbol{p}|y=+1,p_e}\big(|z^n|\big) \\ &= \mathbb{E}_{X,\boldsymbol{p}|y=+1,p_e}(z^n) - 2\mathbb{E}_{X,\boldsymbol{p}|y=+1,p_e}\big(z^n\mathbb{I}(z^n<0)\big) \end{aligned} \tag{3.46}$$

where the last equality comes from the definition of absolute value. Now, we can link the first summand in Equation 3.46 to the expected value of n with the result in Equation 3.38. Together, they yield the following inequality:

$$\mathbb{E}_{X,\boldsymbol{p}|y=+1,p_e}(n)\mathbb{E}_{x_j,p_j\sim f_p|y=+1}(s_j) \geq -w_e - w_q + 2\mathbb{E}_{X,\boldsymbol{p}|y=+1,p_e}\big(z^n\mathbb{I}(z^n<0)\big) \tag{3.47}$$

Notice that a negative z^n is never larger than w_e. Thus, we can rewrite its expected value as follows (we omit the subscript $X,\boldsymbol{p}|y=+1,p_e$ for simplicity):

$$\begin{aligned} \mathbb{E}\big(z^n\mathbb{I}(z^n<0)\big) &= \int_{-\infty}^{w_e} z\mathbb{P}(z^n=z)dz \\ &= w_e\int_{-\infty}^{w_e}\mathbb{P}(z^n=z)dz + \int_{-\infty}^{w_e}(z-w_e)\mathbb{P}(z^n=z)dz \\ &= w_e\mathbb{P}(z^n\leq w_e) + S \end{aligned} \tag{3.48}$$

Furthermore, we can show that the term S is linked to the probability of a negative step s_j as follows. Let us introduce the change of variables $z = z' + s$ and rewrite the probability of z^n in terms of z^{n-1} and s_j:

$$\begin{aligned} S &= \int_{w_e}^{-w_e}\mathbb{P}(z^{n-1}=z')\int_{-\infty}^{-z'+w_e}(z'+s-w_e)\mathbb{P}(s_j=s)dsdz' \\ &= \int_{w_e}^{-w_e}\mathbb{P}(z^{n-1}=z')(z'-w_e)\int_{-\infty}^{-z'+w_e}\mathbb{P}(s_j=s)dsdz' \\ &\qquad + \int_{w_e}^{-w_e}\mathbb{P}(z^{n-1}=z')\int_{-\infty}^{-z'+w_e}s\mathbb{P}(s_j=s)dsdz' \\ &\geq 0 + \int_{-\infty}^{0}s\mathbb{P}(s_j=s)ds \\ &= \mathbb{E}\big(s_j\mathbb{I}(s_j<0)\big) \end{aligned} \tag{3.49}$$

where the last inequality follows from the fact that $z' - w_e$ is always non-negative, and that the integral over s is always non-positive. Finally, we can combine Equations 3.48 and 3.49 to derive the following upper bound:

$$\begin{aligned} \mathbb{E}\big(z^n\mathbb{I}(z^n<0)\big) &\geq w_e p_e + \mathbb{E}\big(s_j\mathbb{I}(s_j<0)\big) \\ &\geq -0.28 - \frac{1}{2}\mathbb{E}\big(|s_j|\big) \end{aligned} \tag{3.50}$$

where we derive the first addend by minimising $w_e p_e$ over all possible $p_e \in (0,1)$, and the second by noting that $\mathbb{E}\big(s_j\mathbb{I}(s_j<0)\big) \geq -\mathbb{E}\big(s_j\mathbb{I}(s_j>0)\big)$ and using the definition

of absolute value. By substituting Equation 3.50 in Equation 3.47 we have:

$$\mathbb{E}_{X,\boldsymbol{p}|y=+1,p_e}(n) \geq \frac{-w_e - w_q - 0.56 - \mathbb{E}_{x_j,p_j\sim f_p|y=+1}\big(|s_j|\big)}{\mathbb{E}_{x_j,p_j\sim f_p|y=+1}(s_j)} \tag{3.51}$$

Let us move now to the case when the ground-truth takes the opposite value $y = -1$. Luckily, we can borrow most of the derivation from the $y = +1$ case. Namely, it follows from the definition of absolute value that:

$$\mathbb{E}_{X,\boldsymbol{p}|y=-1,p_e}\big(|z^n|\big) = \mathbb{E}_{X,\boldsymbol{p}|y=-1,p_e}(-z^n) + 2\mathbb{E}_{X,\boldsymbol{p}|y=-1,p_e}\big(z^n\mathbb{I}(z^n > 0)\big) \tag{3.52}$$

which, combined with the result in Equation 3.38 and noting that $\mathbb{E}_{X,\boldsymbol{p}|y=+1}(s_j) = -\mathbb{E}_{X,\boldsymbol{p}|y=-1}(s_j)$, yields:

$$\mathbb{E}_{X,\boldsymbol{p}|y=-1,p_e}(n)\mathbb{E}_{x_j,p_j\sim f_p|y=+1}(s_j) \geq -w_e + w_q - 2\mathbb{E}_{X,\boldsymbol{p}|y=-1,p_e}\big(z^n\mathbb{I}(z^n > 0)\big) \tag{3.53}$$

Lastly, the expected value of a positive z^n (conditional on $y = -1$) can be bounded with the same procedure shown in Equations 3.48, 3.49 and 3.50 for the opposite case. With these results, we can write:

$$\mathbb{E}_{X,\boldsymbol{p}|y=-1,p_e}(n) \geq \frac{-w_e + w_q - 0.56 - \mathbb{E}_{x_j,p_j\sim f_p|y=+1}\big(|s_j|\big)}{\mathbb{E}_{x_j,p_j\sim f_p|y=+1}(s_j)} \tag{3.54}$$

By substituting Equations 3.51 and 3.54 into Equation 3.37, and noting that we can write the expected value of $s_j = x_j w_j$ in terms of the predictor's accuracy p_j only, we get the result in the theorem. □

Together, the bounds in Theorems 3.9 and 3.10 constitute the foundation of our analysis of the US policy in the following section.

3.3.2 Classification error under the US policy

The uncertainty sampling (US) policy comes from the field of active learning, as we explain in Section 2.3.3. Its original goal is to collect more data points in the regions of larger uncertainty, and thus achieving higher accuracy than random sampling. In our context, we can apply the same principle and collect more labels on the items whose classification is not yet settled. More formally, we define the US policy as a greedy collection strategy that always tries to assign the currently available predictor $j = a(t)$ to the item i whose log-odds z_i are closest to zero:

$$\pi_{us}(t) = \underset{i\in M\setminus M^{t-1}_{a(t)}}{\operatorname{argmin}} \big(|z^t_i|\big) \tag{3.55}$$

where ties are broken in lexicographic order.

From the point of view of a single item i, the US policy operates in short bursts of activity. As long as i is the most uncertain item, it keeps receiving new data points. Once the confidence in its classification improves enough, the policy switches its attention to the other items left behind. The only exception to this behaviour happens when the currently available predictor $j = a(t)$ has already labelled the most uncertain item i, and thus a different item $i' = \pi_{us}(t)$ is chosen. We call these events *collisions*, and study their effect on the performance of the US policy separately in Section 3.4. In the remainder of the present discussion we will assume that each predictor j provides a single data point, and therefore no collision occurs.

Note that the behaviour of the US policy amounts to a separate bounded random walk over z_i for each item $i \in M$. However, unlike our results in Section 3.3.1, the value of the stopping threshold w_e is not known in advance. Instead, we can only define it in hindsight as the maxmin magnitude of the log-odds throughout the collection process:

$$w_e \equiv -\max_{t\in[1,T]} \left(\min_{i\in M} \left(|z_i^t| \right) \right) \tag{3.56}$$

where the negative sign is to ensure consistency with our notation in Section 3.3.1.

If we take a snapshot of the log-odds $\boldsymbol{z}$ at the end of the collection process, we end up with a situation like the example in Figure 3.5. There, the magnitude of the log-odds z_i has crossed the threshold w_e on all items except one, shown in red. This can happen frequently, as the value of z_i can always regress toward zero, and thus we may run out of data points before pushing it back towards the threshold. However, given our definition of w_e in Equation 3.56, this can happen only on a single item. When the number of items $|M|$ grows large, this phenomenon becomes negligible, as we explain in the following discussion.

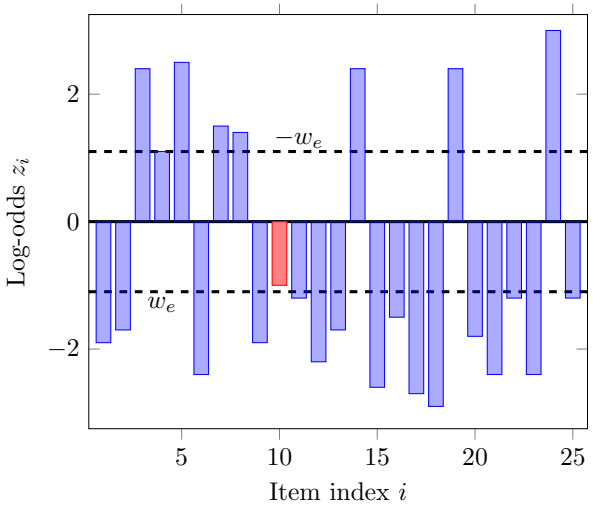


FIGURE 3.5: An example of the state of the US policy at the end of the collection process with $|M| = 25$ items. The red bar represents the most uncertain item, the dashed line represents the threshold w_e.

With this in mind, let us bound the probability of a classification error under the US policy:

Corollary 3.11. *Given a population of predictors with accuracy $p_j \sim f_p$, a prior on the positive class $q \equiv \mathbb{P}(y_i = +1)$, a large number of items $|M| \to \infty$, and an average number of predictors per item R, the probability of a classification error by the US policy under weighted majority voting is upper bounded by:*

$$\mathbb{P}(\hat{y}_i \neq y_i | f_p, q) < \exp\big(-(R-1)\mathbb{E}_{p_j \sim f_p}\big((2p_j - 1)w_j\big) - (2q-1)w_q\big) \tag{3.57}$$

for all items $i \in M$.

Proof. Let us assume for the moment that we can collect as many data points T as we want. Under this assumption, we can run $|M|$ independent random walks on each item $i \in M$ until all their respective log-odds z_i cross a threshold $\pm w_e$ of our choice. Furthermore, we can compute the average number of predictors per item $|N_i|$ we need according to Theorem 3.9. The latter provides the following expression:

$$-w_e > \Big(\mathbb{E}_{X,\boldsymbol{p},y_i|q,p_e}\big(|N_i|\big) - 1\Big)\mathbb{E}_{p_j \sim f_p}\big((2p_j - 1)w_j\big) + (2q-1)w_q \tag{3.58}$$

At the same time, when the number of items $|M|$ grows large, we can show that the total number of data points $T = \sum_{i \in M} |N_i|$ is close to its expected value with high probability. We do so by bounding the dispersion of T around its expected value with Chebyshev's inequality:

$$\mathbb{P}(|T - \mathbb{E}_{X,\boldsymbol{p},\boldsymbol{y}|q,p_e}(T)| \geq \epsilon) \leq \frac{\mathrm{Var}_{X,\boldsymbol{p},\boldsymbol{y}|q,p_e}(T)}{\epsilon^2} \tag{3.59}$$

and the making the following substitutions. First, the random walks are independent, thus $\mathbb{E}(T) = |M|\mathbb{E}\big(|N_i|\big)$ and similarly $\mathrm{Var}(T) = |M|\mathrm{Var}\big(|N_i|\big)$. Second, we can introduce the average number of predictors per item $R = T/|M|$ by dividing the left-hand side by $|M|$. Third, we can replace our measure of dispersion with $\delta = \epsilon/|M|$. Together, this changes yield:

$$\mathbb{P}(|R - \mathbb{E}_{X,\boldsymbol{p},\boldsymbol{y}|q,p_e}(|N_i|)| \geq \delta) \leq \frac{\mathrm{Var}_{X,\boldsymbol{p},\boldsymbol{y}|q,p_e}(|N_i|)}{\delta^2 |M|} \tag{3.60}$$

which goes to zero as $|M| \to \infty$ for all $\delta > 0$, since the variance at the numerator is provably finite (see Appendix C).

Finally, let us drop the assumption that we can collect as many data points as we need. Additionally, let us choose w_e such that $\mathbb{E}_{X,\boldsymbol{p},\boldsymbol{y}|q,p_e}\big(|N_i|\big) = R$, i.e. the expected number of predictors per item is equal to the amount we can actually afford. According to Equation 3.60, this is enough to push the log-odds z_i across the threshold $\pm w_e$ for each item $i \in M$ with high probability. Therefore, we can link the value of the log-odds z_i^T at

the end of the collection process with the probability of a classification error as follows:

$$\begin{aligned}\mathbb{P}(\hat{y}_i \neq y_i | f_p, q) &= \mathbb{E}_{X,\boldsymbol{p},\boldsymbol{y}|q}\Big(\operatorname{sig}\big(-|z_i^T|\big)\Big) \\ &\leq \operatorname{sig}\big(-|w_e|\big) \\ &\leq \exp\big(w_e\big)\end{aligned} \tag{3.61}$$

where the last inequality comes from the properties of the sigmoid function and the fact that w_e is negative for any $p_e < \frac{1}{2}$. By bounding the value of w_e further with Equation 3.58, and replacing the expected value of $|N_i|$ with R, we get the result in the corollary. □

Corollary 3.12. *Given a population of predictors with accuracy $p_j \sim f_p$, a prior on the positive class $q \equiv \mathbb{P}(y_i = +1)$, a large number of items $|M| \to \infty$, and an average number of predictors per item R, the probability of a classification error by the US policy under weighted majority voting is lower bounded by:*

$$\mathbb{P}(\hat{y}_i \neq y_i | f_p, q) \geq \exp\big(-R\mathbb{E}_{p_j\sim f_p}\big((2p_j-1)w_j\big)-(2q-1)w_q-1.25-\mathbb{E}_{p_j\sim f_p}\big(|w_j|\big)\big) \tag{3.62}$$

for all items $i \in M$.

Proof. Let us assume for the moment that we can collect as many data points T as we want. Then, we can run $|M|$ independent random walks on the log-odds z_i of each item $i \in M$, until they cross a threshold $\pm w_e$ of our choice. Under these conditions, the techniques described in the proof of Theorem 3.10 are valid. We use them to put an upper bound on the expected value of the absolute log-odds $|z_i^T|$ at the end of the collection process as follows:

$$\mathbb{E}_{X,\boldsymbol{p}|y_i,p_e}\big(|z_i^T|\big) \leq y_i\mathbb{E}_{X,\boldsymbol{p}|y_i,p_e}(z_i^T) + 0.56 + \mathbb{E}_{p_j\sim f_p}\big(|w_j|\big) \tag{3.63}$$

where p_e is the probability of an error associated to the threshold w_e. Furthermore, we can use Equation 3.38 to link the expected value of z_i^T with the number of predictors $|N_i|$:

$$\mathbb{E}_{X,\boldsymbol{p}|y_i,p_e}(z_i^T) = \mathbb{E}_{X,\boldsymbol{p}|y_i,p_e}(|N_i|)\mathbb{E}_{x_{ij},p_j\sim f_p|y_i}(x_{ij}w_j) + w_q \tag{3.64}$$

We also know that the probability of y_i being positive is q. With this in mind, we can take the expectation of Equations 3.63 and 3.64 over y_i, and combine them as follows:

$$\begin{aligned}\mathbb{E}_{X,\boldsymbol{p},y_i|q,p_e}\big(|z_i^T|\big) \leq\ &\mathbb{E}_{X,\boldsymbol{p},y_i|q,p_e}(|N_i|)\mathbb{E}_{p_j\sim f_p}\big((2p_j-1)w_j\big) \\ &+ (2q-1)w_q + 0.56 + \mathbb{E}_{p_j\sim f_p}\big(|w_j|\big)\end{aligned} \tag{3.65}$$

However, for the reasons explained in the proof of Corollary 3.11, we can choose the threshold w_e in such a way that the number of data points T we need to complete all random walks converges to its expected value, when the number of items $|M|$ grows to

infinity. In other terms, there exists a value of w_e such that $R \to \mathbb{E}(|N_i|)$ when $|M| \to \infty$. Additionally, we can establish the following relationship between the probability of an error and the absolute value of the log-odds $|z_i^T|$ at the end of the collection process:

$$\begin{aligned} \mathbb{P}(\hat{y}_i \neq y_i | f_p, q) &= \mathbb{E}_{X,\boldsymbol{p},\boldsymbol{y}|q}\Big(\operatorname{sig}\big(-|z_i^T|\big)\Big) \\ &\geq \operatorname{sig}\Big(-\mathbb{E}_{X,\boldsymbol{p},\boldsymbol{y}|q}\big(|z_i^T|\big)\Big) \\ &\geq \frac{1}{2}\exp\Big(-\mathbb{E}_{X,\boldsymbol{p},\boldsymbol{y}|q}\big(|z_i^T|\big)\Big) \end{aligned} \tag{3.66}$$

where the first inequality comes from Jensen's inequality, and the second from the properties of the sigmoid function. By substituting the right-hand side of Equation 3.65 into Equation 3.66, and replacing the expected value of $|N_i|$ with R, we get the result in the corollary. □

Three things must be noted in the results of Corollaries 3.11 and 3.12. First, the two bounds match asymptotically for $R \to \infty$, as they share the common term $\exp(-Rc_{us})$, where c_{us} is the expected value of a step in the random walk. We use this fact to compare the US policy to the other policies in Section 3.5. Second, since c_{us} corresponds to the average drift in the random walk, as explained in Section 3.1, we can say that the US policy is able to take full advantage of the available predictive power. In this sense, the US policy is *optimal*. Third, the prior q takes the role of an extra predictor by reducing the probability of a classification error by $(2q-1)w_q$, instead of the term $(2p_j-1)w_j$ in the definition of c_{us}. This property mirrors our findings in Section 3.2.1 for non-adaptive policies.

To conclude this section, we present in Figure 3.6 a qualitative comparison between the bounds in Corollaries 3.11 and 3.12 and the empirical probability of a classification error under the US policy. For consistency reasons, we keep the same setting as the experiments in Figures 3.3 and 3.4 for the UNI and WB policies respectively. Notice how the upper bound in Corollary 3.11 closely matches the probability of a classification error. Furthermore, notice how the US policy outperforms the non-adaptive UNI and WB policies as R grows large. We quantify this performance gap in a general setting in Section 3.5. The discussion on the performance of the other two adaptive policies, IG and LOS, is presented in the following Sections 3.3.3 and 3.3.4.

3.3.3 Classification error under the IG policy

From the information-theoretical perspective, each data point x_{ij} we collect brings additional information on the ground-truth. A natural goal is to try and maximise the amount of information we gain during the collection process (see Section 2.3.3). We can measure this amount of information by computing the Kullback-Leibler (KL) divergence between the posterior distribution on the item classes before and after observing

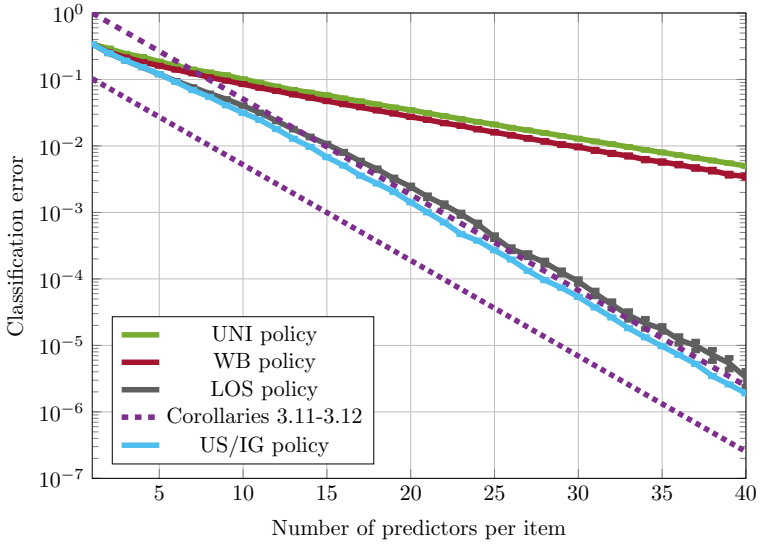


FIGURE 3.6: Comparison between the empirical classification error under the US/IG policies, and the bounds we derive in Corollaries 3.11 and 3.12. The dashed line is the upper bound, the dotted one is the lower bound. For reference we include the UNI, WB and LOS policies as well.

x_{ij}. Since, the posterior changes only for the item i we collect the new data point on, the information we gain observing x_{ij} takes the following simplified form:

$$\begin{aligned} KL(\boldsymbol{z}^{t+1} \parallel \boldsymbol{z}^t) &= KL(z_i^{t+1} \parallel z_i^t) \\ &= \sum_{y \in \{\pm 1\}} \text{sig}(y z_i^{t+1}) \log\left(\frac{\text{sig}(y z_i^{t+1})}{\text{sig}(y z_i^t)}\right) \\ &= \text{sig}(z_i^{t+1})\left(\log\left(\frac{\text{sig}(z_i^{t+1})}{\text{sig}(-z_i^{t+1})}\right) + \log\left(\frac{\text{sig}(-z_i^t)}{\text{sig}(z_i^t)}\right)\right) + \log\left(\frac{\text{sig}(-z_i^{t+1})}{\text{sig}(-z_i^t)}\right) \\ &= \text{sig}(z_i^t + x_{ij} w_j) x_{ij} w_j + \log\left(\frac{\text{sig}(-z_i^t - x_{ij} w_j)}{\text{sig}(-z_i^t)}\right) \end{aligned} \tag{3.67}$$

where we used several of the properties of the sigmoid listed in Appendix D.

However, the expression in Equation 3.67 is not enough to define a corresponding collection policy. In fact, we do not just want to measure the information gain after collecting a data point x_{ij}. Instead, we want to choose the item i so as to maximise this quantity at every time step t. Since x_{ij} cannot be observed before choosing the item i, our only option is to compute Equation 3.67 in expectation over the future observation x_{ij}. We

define the corresponding expected information gain for each item i in M as follows:

$$\begin{aligned}\mathbb{E}_{x_{ij}|\boldsymbol{z}^t,p_j}\big(KL(\boldsymbol{z}^{t+1} \parallel \boldsymbol{z}^t)\big) &= \sum_{x\in\{\pm 1\}} \mathbb{P}(x_{ij}=x|z_i^t,p_j)KL(z_i^t + xw_j \parallel z_i^t) \\ &= w_j(2p_j-1)\,\mathrm{sig}(z_i^t) \\ &\quad + \sum_{x\in\{\pm 1\}} \mathbb{P}(x_{ij}=x|z_i^t,p_j)\log\Big(\frac{\mathrm{sig}(-z_i^t - x_{ij}w_j)}{\mathrm{sig}(-z_i^t)}\Big) \\ &= w_j(2p_j-1)\,\mathrm{sig}(z_i^t) - \log\big(\mathrm{sig}(-z_i^t)\big) \\ &\quad + \mathbb{P}(x_{ij}=+1|z_i^t,p_j)\log\big(\mathrm{sig}(-z_i^t - w_j)\big) \\ &\quad + \mathbb{P}(x_{ij}=-1|z_i^t,p_j)\log\big(\mathrm{sig}(-z_i^t + w_j)\big)\end{aligned} \tag{3.68}$$

Thanks to Equation 3.67, we can greedily maximise the expected information gain at each time step t. In so doing, we hope to improve the "usefulness" of each data point x_{ij} we collect, and thus reduce the probability of a classification error. We formally define this information gain maximisation (IG) policy in the following way:

$$\pi_{ig}(t) = \operatorname*{argmax}_{i\in M\setminus M_{a(t)}^{t-1}} \Big(\mathbb{E}_{x_{ia(t)}|\boldsymbol{z}^t,p_{a(t)}}\big(KL(\boldsymbol{z}^{t+1} \parallel \boldsymbol{z}^t)\big)\Big) \tag{3.69}$$

where ties are broken in lexicographic order.

To give the reader an intuitive understanding of the behaviour of the IG policy, we plot the expected information gain for different values of the log-odds z_i^t and the predictor's weight w_j in Figure 3.7. Note that for an uninformative predictor with weight $w_j = 0$, Equation 3.68 is zero for any z_i^t. In all the other cases, the expected information gain is larger when the log-odds are closer to zero, that is on items whose classification we are less confident about.

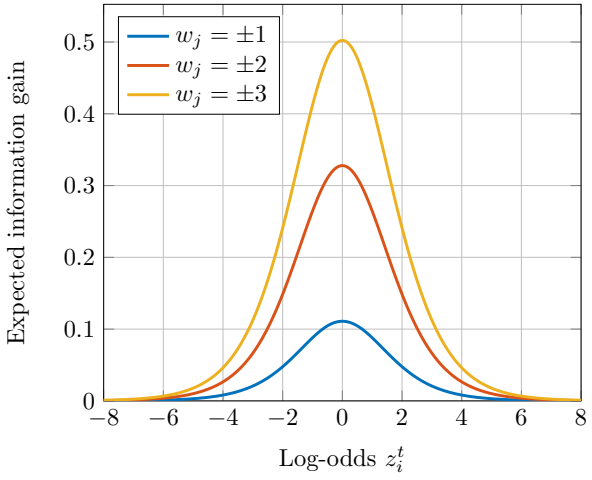


FIGURE 3.7: Expected information gain for different values of the log-odds z_i^t and the predictor's weight w_j.

This observation suggests that the IG policy behaves like the US policy, and always

selects the most uncertain item. In fact, we can prove this equivalence as follows:

Theorem 3.13. *For any values of the log-odds $\boldsymbol{z}^t$, and any predictor j with accuracy $p_j \neq \frac{1}{2}$, the US and IG policies select the same item, that is $\pi_{us}(t) = \pi_{ig}(t)$.*

Proof. Recall that the US policy (see Equation 3.55) selects the item with the smallest absolute log-odds $|z_i^t|$. This function is even, monotonically increasing on both sides, and has a minimum in $z_i^t = 0$. In order to prove that the IG policy has the same behaviour, it suffices to prove that the expected information gain in Equation 3.68 has similar properties for any $w_j \neq 0$. The only difference is that the IG policy maximises rather then minimises, and thus we need to prove that Equation 3.68 is monotonically decreasing on both sides, and has a maximum in $z_i^t = 0$.

First, note that Equation 3.68 is even both in z_i^t and w_j, as can be easily proven by substitution. Given this, we focus only on non-negative values of z_i^t and w_j and prove that the function is monotonically decreasing in z_i^t. To do so, we need to take the derivative of the expected information gain with respect to the log-odds z_i^t, and show that its sign is negative:

$$\begin{aligned}\frac{d}{dz_i^t}\Big\{\mathbb{E}_{x_{ij}|\boldsymbol{z}_i^t,p_j}\big(KL(\boldsymbol{z}_i^{t+1} \parallel \boldsymbol{z}_i^t)\big)\Big\} &= w_j(2p_j-1)\,\mathrm{sig}(z_i^t)\,\mathrm{sig}(-z_i^t) + \mathrm{sig}(z_i^t) \\ &+ \frac{d}{dz_i^t}\Big\{\mathbb{P}(x_{ij}=+1|z_i^t,p_j)\Big\}\log\big(\mathrm{sig}(-z_i^t-w_j)\big) \\ &- \mathbb{P}(x_{ij}=+1|z_i^t,p_j)\,\mathrm{sig}(z_i^t+w_j) \\ &+ \frac{d}{dz_i^t}\Big\{\mathbb{P}(x_{ij}=-1|z_i^t,p_j)\Big\}\log\big(\mathrm{sig}(-z_i^t+w_j)\big) \\ &- \mathbb{P}(x_{ij}=-1|z_i^t,p_j)\,\mathrm{sig}(z_i^t-w_j)\end{aligned} \tag{3.70}$$

However, because of the properties listed in Appendix D we have:

$$\mathrm{sig}(z_i^t) - \mathbb{P}(x_{ij}=+1|z_i^t,p_j)\,\mathrm{sig}(z_i^t+w_j) - \mathbb{P}(x_{ij}=-1|z_i^t,p_j)\,\mathrm{sig}(z_i^t-w_j) = 0 \tag{3.71}$$

and:

$$\frac{d}{dz_i^t}\Big\{\mathbb{P}(x_{ij}=x|z_i^t,p_j)\Big\} = \mathrm{sig}(z_i^t)\,\mathrm{sig}(-z_i^t)\big(\mathrm{sig}(xw) - \mathrm{sig}(-xw)\big) \tag{3.72}$$

Together, Equations 3.70, 3.71 and 3.72 yield the following expression for the derivative of the expected information gain:

$$\begin{aligned}&\frac{d}{dz_i^t}\Big\{\mathbb{E}_{x_{ij}|\boldsymbol{z}_i^t,p_j}\big(KL(\boldsymbol{z}_i^{t+1} \parallel \boldsymbol{z}_i^t)\big)\Big\} \\ &\qquad = (2p_j-1)\,\mathrm{sig}(z_i^t)\,\mathrm{sig}(-z_i^t)\left(w_j + \log\left(\frac{\mathrm{sig}(-z_i^t-w_j)}{\mathrm{sig}(-z_i^t+w_j)}\right)\right)\end{aligned} \tag{3.73}$$

We prove that Equation 3.73 is negative for any $z_i^t > 0$ and $w_j > 0$ by contradiction:

$$\begin{aligned}
\frac{d}{dz_i^t}\Big\{\mathbb{E}_{x_{ij}|\boldsymbol{z}_i^t,p_j}\big(KL(\boldsymbol{z}_i^{t+1} \parallel \boldsymbol{z}_i^t)\big)\Big\} &\geq 0 \\
\log\Big(\frac{\mathrm{sig}(-z_i^t - w_j)}{\mathrm{sig}(-z_i^t + w_j)}\Big) &\geq -w_j \\
\frac{1+\exp(z_i^t - w_j)}{1+\exp(z_i^t + w_j)} &\geq \exp(-w_j) \\
1+\exp(z_i^t - w_j) &\geq \exp(-w_j) + \exp(z_i^t) \\
\exp(z_i^t - w_j) - \exp(z_i^t) &\geq \exp(-w_j) - 1 \\
\exp(z_i^t)\big(\exp(-w_j) - 1\big) &\geq \exp(-w_j) - 1 \\
\exp(z_i^t) &\leq 1 \\
z_i^t &\leq 0
\end{aligned} \tag{3.74}$$

which is a contradiction under our assumptions. Finally, for $w_j \neq 0$ the derivative in Equation 3.72 has its only zero in $z_i^t = 0$, proving that the expected information gain has a global maximum in that location. □

Given that the US and IG policies are equivalent, as we prove in Theorem 3.13, their performance in terms of classification errors is the same. Therefore, our results in Section 3.3.2 apply.

3.3.4 Classification error under the LOS policy

As the previous two sections show, both the US and IG policies achieve an optimal asymptotic decay of the probability of a classification error. However, they do so by optimising an indirect objective, respectively the item uncertainty and information gain. A more direct approach would be trying to minimise the probability of a classification error itself. While this approach is standard in the active learning literature (see Section 2.3.3), we show that it is ineffective in the present context.

Recall that we can measure the overall probability of a classification error by looking at the current log-odds vector $\boldsymbol{z}^t$. This gives us the expected zero-one loss, i.e. the number of misclassified items, if no more data point are to be collected. More formally:

$$\mathcal{L}(\boldsymbol{z}^t) = \mathbb{E}_{\boldsymbol{y}|\boldsymbol{z}^t}\Big(\sum_{i' \in M} \mathbb{I}(\hat{y}_{i'} \neq y_{i'})\Big) = \sum_{i' \in M} \mathrm{sig}\big(-|z_{i'}^t|\big) \tag{3.75}$$

However, Equation 3.75 does not tell us which item i we need to allocate the incoming predictor $j = a(t)$ on. This is because we can only compute $\mathcal{L}(\boldsymbol{z}^t)$ after observing the next label x_{ij}. In the same manner as our work on the IG policy (see Section 3.3.3), we overcome this issue by taking the expectation of Equation 3.75 over x_{ij}. This gives us

the following expression:

$$\begin{aligned}\mathbb{E}_{x_{ij}|\boldsymbol{z}^t,p_j}\big(\mathcal{L}(\boldsymbol{z}^{t+1})\big) = &\sum_{i'\neq i} \operatorname{sig}\big(-|z_{i'}^t|\big) \\ &+ \mathbb{P}(x_{ij}=+1|z_i^t,p_j)\operatorname{sig}\big(-|z_i^t+w_j|\big) \\ &+ \mathbb{P}(x_{ij}=-1|z_i^t,p_j)\operatorname{sig}\big(-|z_i^t-w_j|\big)\end{aligned} \tag{3.76}$$

where the incoming label x_{ij} has only an effect on the probability of an error on item i, while the rest remain unchanged.

With Equation 3.76 in mind, we can formally define a zero-one loss minimisation (LOS) policy. For simplicity, we consider the expected reduction in zero-one loss, instead of using Equation 3.76 as is. This allows us to focus on what happens on item i alone. As a result, we maximise over this alternative objective function, instead of minimising the original one:

$$\pi_{los}(t) = \operatorname*{argmax}_{i\in M\setminus M_{a(t)}^{t-1}} \Big(\mathcal{L}(z_i^t) - \mathbb{E}_{x_{ij}|z_i^t,p_j}\big(\mathcal{L}(z_i^{t+1})\big)\Big) \tag{3.77}$$

where ties are broken in lexicographic order.

Now, let us consider the shape of the objective function in Equation 3.77. To this end, we plot it in Figure 3.8 for a range of values of the current log-odds z_i^t, and the incoming predictor's weight w_j. At a first glance, it becomes apparent that the function is not smooth everywhere. Specifically, we have three points of non differentiability in $z_i^t = 0$ and $z_i^t = \pm w_j$, where the former is also the global maximum. More importantly, the objective function goes to zero for any z_i^t outside of the range $\big(-|w_j|, |w_j|\big)$.

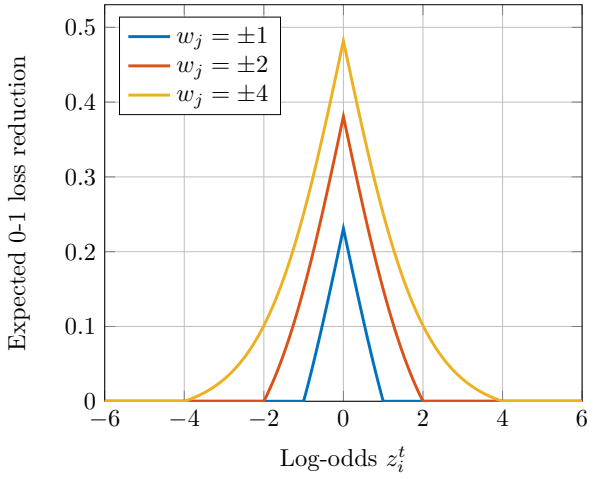


FIGURE 3.8: Expected zero-one loss reduction for different values of the log-odds z_i^t and the predictor's weight w_j.

We can show analytically that the last property does indeed hold as follows:

Proposition 3.14. *For any value of the log-odds z_i^t and the predictor's weight w_j such*

that $|z_i^t| \geq |w_j|$, the expected reduction in zero-one loss is zero.

Proof. First, let us rewrite the expected reduction in zero-one loss explicitly:

$$\begin{aligned}\mathcal{L}(z_i^t) - \mathbb{E}_{x_{ij}|z_i^t,p_j}\big(\mathcal{L}(z_i^{t+1})\big) &= \operatorname{sig}\big(-|z_i^t|\big) \\ &\quad + \mathbb{P}(x_{ij}=+1|z_i^t,p_j)\operatorname{sig}\big(-|z_i^t+w_j|\big) \\ &\quad + \mathbb{P}(x_{ij}=-1|z_i^t,p_j)\operatorname{sig}\big(-|z_i^t-w_j|\big)\end{aligned} \tag{3.78}$$

which corresponds to the argument of Equation 3.77. Then, notice that the function in Equation 3.78 is even in both z_i^t and w_j, as can be proven by substitution. Thus, let us focus on the positive quadrant and deduce the other cases by symmetry. When both $z_i^t > 0$, $w_j > 0$ and $z_i^t > w_j$, Equation 3.78 simplifies as follows:

$$\begin{aligned}\mathcal{L}(z_i^t) - \mathbb{E}_{x_{ij}|z_i^t,p_j}\big(\mathcal{L}(z_i^{t+1})\big) &= \operatorname{sig}\big(-z_i^t\big) \\ &\quad + \mathbb{P}(x_{ij}=+1|z_i^t,p_j)\operatorname{sig}(-z_i^t-w_j) \\ &\quad + \mathbb{P}(x_{ij}=-1|z_i^t,p_j)\operatorname{sig}(-z_i^t+w_j) \\ &= \operatorname{sig}(-z_i^t) - \operatorname{sig}(-z_i^t)\operatorname{sig}(-w_j) - \operatorname{sig}(-z_i^t)\operatorname{sig}(w_j) \\ &= 0\end{aligned} \tag{3.79}$$

where the second equality comes from the properties of the sigmoid function listed in Appendix D. □

The property in Proposition 3.14 has a crucial impact on the behaviour of the LOS policy. In fact, at the beginning of the collection process the log-odds $\boldsymbol{z}$ are close to zero, and the policy behaves like the US policy. However, once the magnitude of the log-odds goes beyond that of the incoming weight $|w_j|$ on the whole set M, the LOS policy has no preference on which item i to select. Even assuming that the ties in Equation 3.76 are broken at random, instead of the lexicographic order, the behaviour of this collection policy becomes worse than the UNI policy.

For a qualitative comparison of the LOS policy with the other collection policies, refer to Figure 3.6. There, we can see that this policy manages to improve over both non-adaptive policies UNI and WB, despite its erratic behaviour at the end of the collection process. We attribute this to its equivalence with the optimal US policy when the magnitude of the log-odds $\boldsymbol{z}$ is low. However, with respect to the latter, the LOS policy is noticeably inferior. For this reason, we exclude it from further analysis.

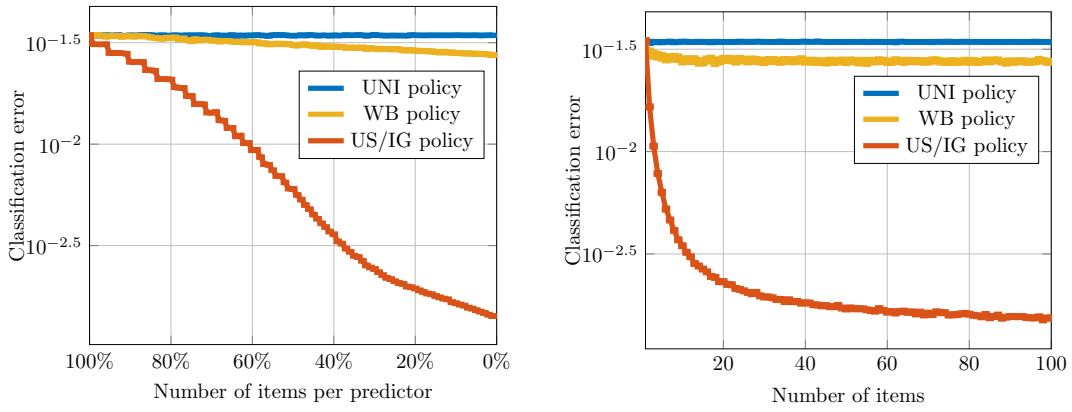
3.4 Running the policies under non-ideal conditions

Our analysis of the adaptive and non-adaptive collection policies in Sections 3.2 and 3.3 is based on the assumption that we can always assign the available predictor $j = a(t)$

to any item i of our choice. In scenarios where the predictors label multiple items, this assumption may not hold. Specifically, predictor $j = a(t)$ may have already labelled item i, and thus reassigning it to the same item will just elicit a repeated response.

When this happens, the collection policy is forced to choose another item i'. We call such events *collisions* and we study their impact on the performance of the collection policies here. Let us establish a worst-case and a best-case scenarios. On one end of the spectrum, we have the case where each predictor labels all items, and therefore all collection policies end up assigning all the N predictors to each item, as the UNI policy would do. In this case we observe no difference between the collection policies. On the other end of the spectrum, we have the case where each predictor labels only one item and thus no collision occurs. This is the assumption we used in our theoretical analysis in Sections 3.2.2, 3.2.3 and 3.3.2.

In order to study what happens in between these two extremes, we resort to an empirical analysis. To this end, we fix the number of items to $|M| = 1000$ and let each predictor label a subset of them ranging from $L = |M|$ (all the items) to $L = 1$ (only one). We extract the predictors' accuracy from a Beta$(4, 3)$ distribution, collect an average of $R = 20$ labels per item, and compute the empirical classification error for the UNI, WB and US/IG policies. Note that we make this choice of parameters to highlight the phenomenon under study, but similar results can be obtained with other choices. As the results in Figure 3.9(a) show, the US/IG policy retains its ability to deliver an order of magnitude lower classification error than both non-adaptive policies, even when the predictors label up to 40% of the items. This is because, when a collision occurs at time t, the US/IG policy is still able to assign some other predictor to the most uncertain item at a later stage. In broader terms, as long as there is enough scope to strategise during the collection process, the US/IG policy is able to compensate for the occurrence of most collisions. A similar phenomenon happens for the WB policy, though its advantage over the UNI policy is minimal even in the best case scenario.



(a) Collisions

(b) Limited number of items

FIGURE 3.9: Impact of non-ideal conditions on the performance of the policies.

On a related note, recall that our analysis of the US policy in Section 3.3.2 requires the assumption that the number of items is large. This is also valid for the IG policy, since the two are equivalent as we prove in Section 3.3.3, and the WB policy, albeit in a more indirect way. Intuitively, any of these policies benefits from a larger number of options during the collection process, as it can intelligently balance the assignment of predictors across the whole set of items. Conversely, if we only have a single item to classify, the policy is left with no option but to assign all predictors to that single item. What our theoretical results in Sections 3.2.3 and 3.3.2 do not tell us is how many items the WB and US/IG policies need to reach their full potential.

We give an answer to this question by running the following experiment. First, we fix the number of labels per predictor to $L = 1$ to avoid the chance of a collision. Then, we let the number of items increase from $|M| = 1$ to $|M| = 100$ and compute the empirical classification error for the UNI, WB and US/IG policies. All other parameters are the same as the experiment in Figure 3.9(a). As the results in Figure 3.9(b) show, for $|M| = 1$ the three policies are equivalent as expected. However, for larger values of $|M|$ the US/IG policy shows a definite advantage: for $|M| = 10$ the difference is already an order of magnitude, and for $|M| = 50$ the chance of a misclassification has almost reached its asymptotic value. Similarly, the WB policy only needs a few items to reach its asymptotic value.

In conclusion, we have shown that the UNI policy is immune to the occurrence of collisions and independent of the number of items. In contrast, the performance of more intelligent collection strategies like the WB and US/IG policies degrades both with a high rate of collisions and a limited number of items to be classified. These two conditions can be used to guide the design choices when deploying a data collection policy in the real-world.

For example, we report in Table 5.1 the characteristics of five publicly available crowdsourcing datasets, which are often used as a benchmark in the corresponding literature (Snow et al., 2008; Welinder et al., 2010; Lease and Kazai, 2011; Bonald and Combes, 2017). In three cases out of five the number of items labelled by each predictor is small enough for the US/IG policy to be worthwhile. Specifically, for the RTE dataset it is 6%, for the TEMP dataset it is 13% and for the TREC dataset it is 2%.

In contrast, the difference in performance between the UNI and WB policies is so small that it does not justify the added complexity of the latter, even under ideal circumstances. In the next Section 3.5, we quantify the advantage of using an adaptive policy instead of a non-adaptive one in a general setting.

3.5 Asymptotic policy comparison

Now that we have established several bounds on the performance of both adaptive and non-adaptive collection policies, we can compare their efficiency in aggregating the predictors' output. We do so by analysing their *asymptotic* behaviour when the average number of labels per item R grows large. Under this scenario, all policies exhibit an exponential dependency between R and the classification error, which takes the form:

$$\mathbb{P}(\hat{y} \neq y) = \exp\big(-c_\pi R + o(\sqrt{R})\big) \tag{3.80}$$

where the term $o(\sqrt{R})$ encompasses all the negligible contributions as $R \to \infty$.

At the same time, the value of the constant c_π varies according to the policy π we use. For the UNI policy, both the upper and the lower bounds in Corollary 3.7 contain the following asymptotic term:

$$c_{uni} = -\log\Big(2\mathbb{E}_{p_j}\big(\sqrt{p_j(1-p_j)}\big)\Big) \tag{3.81}$$

Similarly, Corollary 3.8 yields the following constant for an optimal non-adaptive policy:

$$c_{nada} = -\frac{1}{2}\mathbb{E}_{p_j}\Big(\log\big(4p_j(1-p_j)\big)\Big) \tag{3.82}$$

which the WB policy tries to approximate.

Likewise, we can derive the value of c_π for the US policy from Corollaries 3.11 and 3.12. Thanks to our equivalence result in Section 3.3.3, the IG policy shares the same value, which we define as follows:

$$c_{ada} = \mathbb{E}_{p_j}\Big((2p_j - 1)\log\Big(\frac{p_j}{1-p_j}\Big)\Big) \tag{3.83}$$

In general, a larger value of c_π means that the policy is more efficient in using additional data points to reduce the classification error. In this respect, we can prove that the adaptive US/IG policy is asymptotically superior to any non-adaptive policy:

Theorem 3.15. *Given a population of predictors with accuracy $p_j \sim f_p$ such that $\mathbb{P}(p_j \neq \frac{1}{2}) > 0$, a large number of items $|M| \to \infty$, and a large number of predictors per item $R \to \infty$, the asymptotic rate of the US and IG policies is at least twice as large as that of any non-adaptive policy. That is $c_{ada} \geq 2c_{nada}$.*

Proof. If we apply Jensen's inequality to the definitions of c_{uni} and c_{nada} in Equations 3.81 and 3.82 we get the following relationship:

$$2c_{uni} \leq 2c_{nada} = \mathbb{E}_{p_j}\Big(-\log\big(4p_j(1-p_j)\big)\Big) \tag{3.84}$$

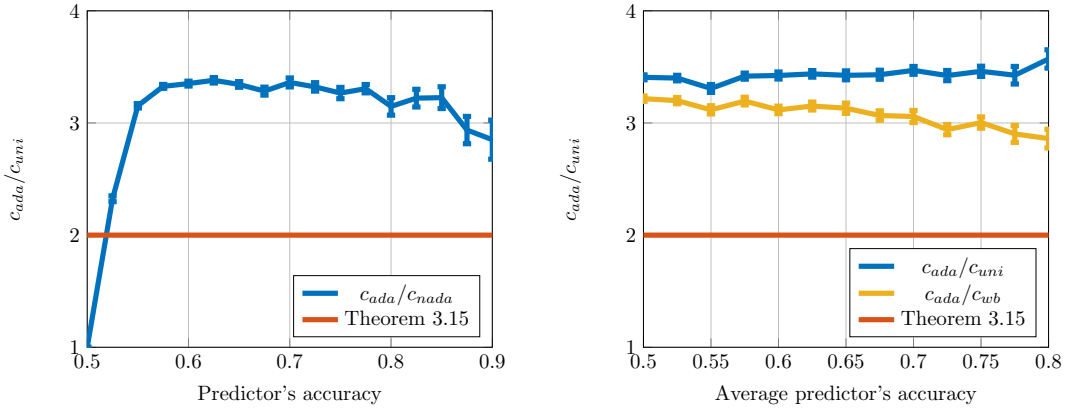
Then, by comparing the arguments of the expectations in Equations 3.82 and 3.83 we can notice that:

$$-\log\left(4p_j(1-p_j)\right) \leq (2p_j-1)\log\left(\frac{p_j}{1-p_j}\right) \qquad \text{for all} \qquad p_j \in (0,1) \tag{3.85}$$

Finally, the result in the theorem follows from the monotonicity property of the expectation operator. □

At the same time, the result in Theorem 3.15 does not tell how much the c_{ada}/c_{nada} ratio is larger than its theoretical baseline. In fact, we can only approximate the behaviour of an optimal non-adaptive policy either by using the WB or UNI policy. We investigate this through an empirical study with different populations of predictors f_p. For each choice of f_p we run experiments with $M = 1000$, $L = 1$ and increasing values of R. Then, we estimate the value of c_π by fitting the experimental data. The results are summarised in Figure 3.10.

Specifically, in Figure 3.10(a) we report the empirical value of the c_{ada}/c_{nada} ratio when all predictors have the same accuracy $p_j = \bar{p}$. Under this setting, both the UNI and WB policies exhibit the same behaviour, as we explain in Section 3.2.3. Note how the value drops as $\bar{p}$ approaches $\frac{1}{2}$. In this scenario, the output of the predictors is almost random, and the advantage of using an adaptive policy is reduced. In the extreme case of $\bar{p} = \frac{1}{2}$, which is not covered by Theorem 3.15, the weighted sum z_i is always zero, and thus the predictions are random for all policies. Conversely, for $\bar{p} > 0.9$ the classification error quickly goes to zero as R increases, and thus it is impossible to have a reliable estimate of c_π. At the same time, for all the other values of $\bar{p}$, the c_{ada}/c_{nada} ratio is above or close to three, a 50% larger value than its theoretical baseline.



(a) Identical predictors $p_j = \bar{p}$

(b) Mixed predictors $p_j \sim \text{Uniform}(\bar{p}-0.2, \bar{p}+0.2)$

FIGURE 3.10: Empirical advantage of adaptive policies.

Likewise, we report in Figure 3.10(b) our results for a mixed population of predictors. Specifically, in this set of experiments we extract p_j uniformly in the interval $(\bar{p}-0.2, \bar{p}+$

0.2) for different values of $\bar{p}$. Under this setting, the UNI and WB policies differ, and thus we report the c_{ada}/c_{uni} and c_{ada}/c_{wb} ratios separately. Note how the latter is smaller than the former across the whole range of f_p, as the WB policy achieves marginally better classification performance. At the same time, we still observe values above three for all choices of $\bar{p}$. The only difference is that, thanks to the mixed population of predictors, we do have neither a singularity around $\bar{p} = \frac{1}{2}$, nor a markedly larger uncertainty for the largest values of $\bar{p}$.

3.6 Summary

Using the results we present here, we now have the foundations for a principled study of data collection policies for the multiple classifier setting. The literature on the topic is sparse and the various existing ideas are often explored and tested under very different assumptions. By reducing all of them to a simplified model with binary classes and known predictor accuracy, we have been able to analyse each of them in depth. Furthermore, having a common ground allowed us to compare the different collection policies, and distill their drawbacks and merits.

The results of this chapter resurface in various forms in the next chapters of this thesis. For this reason, let us go through the key messages again. First, there are two fundamentally different categories of collection policies: non-adaptive and adaptive. In the former category, we discovered that the UNI policy, although slightly less accurate than the WB policy, has a simple implementation, and is robust in all non-ideal settings. Conversely, adaptive policies are more sensitive to the properties of the setting they are run in, but come with provably better performance. In this category, we proved that the US and IG policies, despite addressing different objectives, have identical behaviour, and achieve optimal behaviour. At the same time, we discovered that trying to greedily reduce the probability of a classification error, like the LOS policy does, results in non-optimal behaviour.

On a more technical note, in this chapter we introduce a range of theoretical tools that form the basis for our analysis. First and foremost, we model the behaviour of the collection process as a random walk. By looking at the aggregation algorithm, weighted majority voting, in the log-odds domain, we can separate the contribution of each predictor. This property informs all of our results. Second, we give ourselves the goal of inspecting the relationship between the number of data points per item R and the probability of a classification error. This relationship takes the exponential form of $\mathbb{P}(error) \propto \exp(-c_\pi R)$, with different constants c_π for each collection policy. Being able to derive analytical bounds on the probability of a classification error in this form, allows us to compare the different collection policies theoretically, and not just

empirically. We return to these tools in the next chapters, as we make our underlying model more general, and thus introduce new challenges in our theoretical analysis.

Chapter 4

Unknown predictors' accuracy

It is now time to drop the unrealistic assumption from Chapter 3 that we know the individual accuracies of the predictors. Specifically, this requires dealing with predictors of unknown accuracy. This is the underlying theme of the present and next chapter, as it yields models of greater generality. In fact, many of the practical applications that feature the combination of multiple classifiers bring some form of uncertainty on the accuracy of these. For instance, in crowdsourcing the predictors are anonymous human workers, and as such their accuracy, let alone their identity, cannot be known in advance. Similarly, if the predictors are human experts (medical pratictioners, financial advisers) we may only have access to a record of their past performance. Finally, even for machine learning predictors, like in ensemble classifiers, we can only test their accuracy on a training or validation set.

With this in mind, in this chapter we study an extension of the known accuracy model of Chapter 3 where the accuracy p_j of each predictor is unknown, but we can test it on a small set of trial questions. The number of correct answers during the trials gives a rough indication of the underlying accuracy of the predictor and can be used to aggregate their output. In this sense, the trials are a model of the side information we may have about the predictors, whether it is a report on the past performance of a human expert or the training-set accuracy of a machine learning classifier. When our setting does not include this type of information, we need an even more general model, which we will introduce later in Chapter 5.

Since the present chapter is an extension of the results presented in the previous one, we maintain the same structure. Namely, we introduce all the details of our extended model in Section 4.1. Then, we divide our analysis in two parts: the first tackles the non-adaptive collection policies (Section 4.2) and the second the adaptive ones (Section 4.3). Following that, we present a theoretical comparison between the two categories of policies in Section 4.4. Before concluding, we show how these results can inform the design choices of a practical implementation of a multiple classifier system. We do so in Section

4.5, where we explain how to choose the number of trials and data collection rounds each predictor should go through in order to ensure optimal classification performance. Finally, in Section 4.6 we summarise the contributions of this chapter and link them to the discussion of the next one.

Note that, except for the material in Section 4.5, all the results in this chapter are reminiscent of the corresponding ones for the known accuracy case in Chapter 3. In order to avoid unnecessary repetitions, we frequently refer there for details on the proofs, and general discussion on the behaviour of the policies. Even though we include enough information to make the present chapter self-contained, knowledge of Chapter 4.5 is necessary for a deep understanding of the following discussion.

4.1 An extended model

The model we present in this section is the intermediate step towards the fully agnostic model of Chapter 5. Still, most of the assumptions we make in Chapter 3 remain valid here. At the same time, we introduce a new layer of uncertainty, by dealing with predictors of unknown accuracy.

In order to avoid any confusion, let us quickly go through the one-coin Dawid-Skene model we introduced in Section 3.1 again. Our objective is recovering the ground-truth classes $y_i \in \{\pm 1\}$ of every item i in the set M. To do so, we can poll a set of N predictors which provides us with a limited number of labels $X = \{x_{ij}\}$, where $j \in N$. The predictors become available one by one according to the sequence $\boldsymbol{a}$, and get assigned to a specific item according to the collection policy $\hat{\pi}$. After T timesteps the label collection process ends, and we need to produce a guess $\hat{\boldsymbol{y}}$ on the item classes $\boldsymbol{y}$. The following information helps us with this task. First, we know the value of the prior on the item classes $q \equiv \mathbb{P}(y_i = +1)$. Second, we know that the predictors are independent, their labels are bernoulli-distributed according to $\mathbb{P}(x_{ij} = y_i) = p_j$, and their accuracies $\boldsymbol{p}$ are extracted from the underlying probability distribution f_p.

However, in contrast to the known accuracy model in Section 3.1, the vector $\boldsymbol{p}$ is hidden. Instead, we have access to an additional set of items G on which we can test the accuracy of the predictors. In this case, we assume we know the ground-truth classes $g_k \in \{\pm 1\}$ for each $k \in G$, and we observe all the predictors' labels $O = \{o_{kj}\}$ on them. Furthermore, we assume that their probability of success remains the same across both sets G and M, i.e. $\mathbb{P}(o_{kj} = g_k) = \mathbb{P}(x_{ij} = y_i)$ for any $k \in G$ and $i \in M$. For reference, we collate all these assumptions in the graphical model depicted in Figure 4.1.

The labels O we collect on the test items G allow us to form an estimate on the accuracy p_j of each predictor $j \in N$. We do so by counting the number of successes $v_j = \sum_{k \in G} \mathbb{I}(o_{kj} = g_k)$ each predictor j achieves over the total number of *trials* $|G|$, and

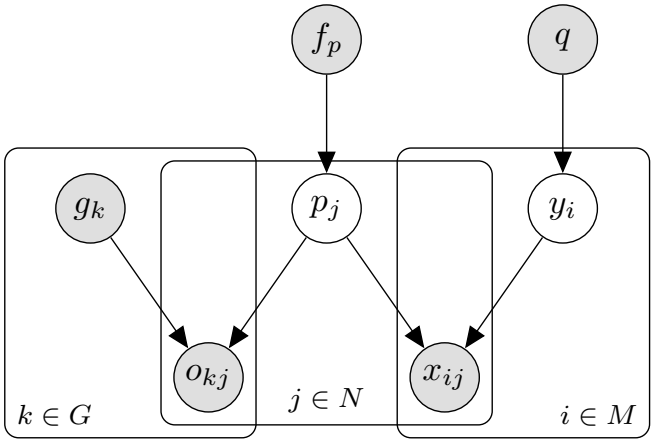


FIGURE 4.1: A graphical representation of the extended Dawid-Skene model.

build our estimates $\hat{p}_j$ accordingly (see Section 2.2.1.3 for examples). In the following discussion we strive not to put any unnecessary restriction on how the estimates $\hat{\boldsymbol{p}}$ are computed when possible. However, we often need to ensure that $\hat{p}_j \in (0, 1)$, for instance in Theorems 4.1 and 4.2, or that all the estimates are unbiased, i.e. $\hat{p}_j = \mathbb{E}_{p_j \sim f_p | O, G}(p_j)$ for all $j \in N$. Specifically, the latter is crucial for our analysis of the adaptive policies in Section 4.3.

No matter which estimator of $\boldsymbol{p}$ we choose, the corresponding estimates $\hat{\boldsymbol{p}}$ can always be useful in predicting the item classes $\boldsymbol{y}$. This is because after collecting the set of labels X as usual, we can use $\hat{\boldsymbol{p}}$ in the so-called *plug-in aggregator* (see Section 2.2.1.3), and form our final predictions on the item classes as follows:

$$\hat{y}_i = \text{sign}\Big(\sum_{j \in N_i} x_{ij}\hat{w}_j + \log\Big(\frac{q}{1-q}\Big)\Big) \qquad \text{where} \qquad \hat{w}_j = \log\Big(\frac{\hat{p}_j}{1-\hat{p}_j}\Big) \tag{4.1}$$

Thanks to Equation 4.1, the parallel with the known accuracy model in Section 3.1 is complete. Specifically, the predictor weight w_j becomes $\hat{w}_j$, the related weighted vote s_{ij} becomes $\hat{s}_i j = x_{ij}\hat{w}_j$, and the log-odds z_i becomes $\hat{z}_i = \sum_{j \in N_i} x_{ij}\hat{w}_j$. Finally, if the unbiasedness property $\hat{p}_j = \mathbb{E}_{p_j | O_j, \boldsymbol{g}}(p_j)$ holds, the following probabilistic interpretation of the log-odds $\hat{z}_i$ is guaranteed (see Equation 3.2 for comparison):

$$\hat{z}_i \equiv \log\left(\frac{\mathbb{P}(y_i = +1 | X, O, \boldsymbol{g}, q)}{\mathbb{P}(y_i = -1 | X, O, \boldsymbol{g}, q)}\right) \tag{4.2}$$

since

$$\begin{aligned}\mathbb{P}(y_i = +1|X, O, \boldsymbol{g}, q) &= \int_{\boldsymbol{p}} \mathbb{P}(y_i = +1, \boldsymbol{p}|X, O, \boldsymbol{g}, q)d\boldsymbol{p} \\ &= \int_{\boldsymbol{p}} \mathbb{P}(y_i = +1|X, \boldsymbol{p}, q)\mathbb{P}(\boldsymbol{p}|O, \boldsymbol{g})d\boldsymbol{p} \\ &\propto q \prod_{j \in N_i} \int_{p_j} p_j^{\mathbb{I}(x_{ij}=+1)}(1-p_j)^{\mathbb{I}(x_{ij}=-1)}\mathbb{P}(p_j|O_j, \boldsymbol{g})dp_j \\ &\propto q \prod_{j \in N_i} \hat{p}_j^{\mathbb{I}(x_{ij}=+1)}(1-\hat{p}_j)^{\mathbb{I}(x_{ij}=-1)}\end{aligned} \tag{4.3}$$

We refer to this last property throughout the present chapter, even though part of our results do not need it.

4.2 Non-adaptive collection policies

In this section we analyse all the collection strategies belonging to the category of non-adaptive policies. Carrying forward the definition we introduced for the known accuracy case (see Section 3.2), a non-adaptive policy chooses how to distribute the set of predictors N over the items $i \in M$ according to the total number of items $|M_j|$ each predictor j can label, and its respective estimated accuracy $\hat{p}_j$. Here, the introduction of the estimates $\hat{\boldsymbol{p}}$ instead of their ground-truth values $\boldsymbol{p}$ introduces a number of subtle differences in our analysis.

For this reason, we need to revise all the results we derived in the known accuracy case. In order to make the comparison easier, we keep the same section structure as Chapter 3. Accordingly, in Section 4.2.1 we derive some general results on the classification error on a single item. Then, in Section 4.2.2 we use them to analyse the performance of the UNI policy. Finally, in Section 4.2.3 we investigate the performance of an optimal non-adaptive policy and approximate its behaviour with the WB policy.

4.2.1 Classification error on a single item

We begin our analysis by deriving some general results on the classification error over a single item. This allows us to simplify our notation, drop the item index i, and assume we observe a single label x_j from each predictor $j \in N$. Later, in Sections 4.2.2 and 4.2.3, we extend these results to multiple items and use them to assess the performance of the UNI and WB policies. Note, that the contribution we present in this section is an improvement over the existing error bounds in (Berend and Kontorovich, 2015) and (Gao et al., 2016).

With this in mind, let us bound the probability of a classification error from above:

Theorem 4.1. *Given a set N of predictors with ground-truth accuracies $\boldsymbol{p}$ and estimated accuracies $\hat{\boldsymbol{p}}$ such that $\hat{p}_j \in (0,1)$ for all $j \in N$, the probability of a classification error under the plug-in aggregator is upper bounded by:*

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, \hat{\boldsymbol{p}}, q) \leq 2\sqrt{q(1-q)} \prod_{j \in N} \left(\frac{p_j}{\hat{p}_j} + \frac{1-p_j}{1-\hat{p}_j} \right) \sqrt{\hat{p}_j(1-\hat{p}_j)} \tag{4.4}$$

where $q \equiv \mathbb{P}(y=+1)$ is the prior on the positive class.

Proof. This proof uses similar arguments to that of Theorem 3.1. First, we rewrite the probability of a classification error as follows:

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, \hat{\boldsymbol{p}}, q) = q\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, \hat{\boldsymbol{p}}, y=+1) + (1-q)\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, \hat{\boldsymbol{p}}, y=-1) \tag{4.5}$$

Second, we define the halved log-odds under the plug-in aggregator as $\hat{h} \equiv \frac{1}{2}w_q + \sum_{j \in N} x_j \frac{1}{2}\hat{w}_j$, and compute the probabilities in Equation 4.5 by marginalising over the content of the dataset X:

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, \hat{\boldsymbol{p}}, y=+1) = \sum_X \Big(\mathbb{I}(\hat{h} < 0) + \frac{1}{2}\mathbb{I}(\hat{h}=0) \Big) \mathbb{P}(X|\boldsymbol{p}, \hat{\boldsymbol{p}}, y=+1) \tag{4.6}$$

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, \hat{\boldsymbol{p}}, y=-1) = \sum_X \Big(\mathbb{I}(\hat{h} > 0) + \frac{1}{2}\mathbb{I}(\hat{h}=0) \Big) \mathbb{P}(X|\boldsymbol{p}, \hat{\boldsymbol{p}}, y=-1) \tag{4.7}$$

Third, we introduce a helper function f which we define as follows:

$$f(x_j, \hat{p}_j, y) \equiv \begin{cases} \hat{p}_j & \text{if } x_j = y \\ 1-\hat{p}_j & \text{if } x_j \neq y \end{cases} \tag{4.8}$$

Fourth, we notice that the probability of observing each $x_j \in X$ is independent from the other data points, conditional on y. This property, together with Equation 4.8, allows us to write the following:

$$\begin{aligned} \mathbb{P}(X|\boldsymbol{p}, \hat{\boldsymbol{p}}, y=+1) &= \prod_{j \in N} \mathbb{P}(x_j|p_j, \hat{p}_j, y=+1) \\ &= \prod_{j \in N} \frac{\mathbb{P}(x_j|p_j, \hat{p}_j, y=+1) f(x_j, \hat{p}_j, +1) \exp(-x_j \frac{1}{2}\hat{w}_j) \sqrt{\hat{p}_j(1-\hat{p}_j)}}{f(x_j, \hat{p}_j, +1)\sqrt{\hat{p}_j(1-\hat{p}_j)} \exp(-x_j \frac{1}{2}\hat{w}_j)} \\ &= \exp(\hat{h} - \frac{1}{2}w_q) \prod_{j \in N} \frac{\mathbb{P}(x_j|p_j, \hat{p}_j, y=+1)}{f(x_j, \hat{p}_j, +1)} \sqrt{\hat{p}_j(1-\hat{p}_j)} \end{aligned} \tag{4.9}$$

where the last equality follows from the fact that:

$$\frac{f(x_j, \hat{p}_j, +1) \exp(-x_j \frac{1}{2}\hat{w}_j)}{\sqrt{\hat{p}_j(1-\hat{p}_j)}} = 1 \qquad \text{for all} \qquad x_j \in \{\pm 1\} \tag{4.10}$$

Similarly, for $y = -1$ we have the following:

$$\mathbb{P}(X|\boldsymbol{p}, \hat{\boldsymbol{p}}, y = -1) = \exp(-\hat{h} + \frac{1}{2}w_q) \prod_{j \in N} \frac{\mathbb{P}(x_j|p_j, \hat{p}_j, y = -1)}{f(x_j, \hat{p}_j, -1)} \sqrt{\hat{p}_j(1 - \hat{p}_j)} \tag{4.11}$$

Now, by substituing Equations 4.9 and 4.11 into Equations 4.6 and 4.7, we get the following two inequalities:

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, \hat{\boldsymbol{p}}, y = +1) \leq \exp(-\frac{1}{2}w_q) \prod_{j \in N} \left(\frac{p_j}{\hat{p}_j} + \frac{1 - p_j}{1 - \hat{p}_j}\right) \sqrt{\hat{p}_j(1 - \hat{p}_j)} \tag{4.12}$$

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, \hat{\boldsymbol{p}}, y = -1) \leq \exp(+\frac{1}{2}w_q) \prod_{j \in N} \left(\frac{p_j}{\hat{p}_j} + \frac{1 - p_j}{1 - \hat{p}_j}\right) \sqrt{\hat{p}_j(1 - \hat{p}_j)} \tag{4.13}$$

Finally, we can plug Equations 4.12 and 4.13 into Equation 4.5, which yields the result in the theorem. □

Let us compare Theorem 4.1 with the corresponding result for the known accuracy case (see Section 3.2.1). First, the two terms $\sqrt{q(1-q)}$ and $\sqrt{\hat{p}_j(1-\hat{p}_j)}$ remain, albeit with the estimates $\hat{p}_j$ taking the place of their ground-truth values p_j. Second, we have an additional term inside the product, which measures how far the estimates are from the aforementioned ground-truth values. Note that for perfect estimates, i.e. $\hat{p}_j = p_j$, the original coefficient of 2 is restored, bringing us back to the known accuracy result. Finally, for purely technical reasons, we have an additional factor of 2 in front of the bound. This means that, with $q = \frac{1}{2}$ and no predictors, our bound yields a maximum value of 1.

Overall, the result in Theorem 4.1 manages to relate the probability of a classification error to the main three underlying quantities: the prior, the estimation error and the number of predictors. In Corollary 4.3 we show how to refine this result even further, with the introduction of an additional assumption. Conversely, the bounds in the existing literature fail to deliver on these three properties. We discuss this in Appendix E.

Let us move on to bounding the classification error from below. Note that the following result is the first attempt at deriving a lower bound in the existing literature (Berend and Kontorovich, 2015; Gao et al., 2016):

Theorem 4.2. *Given a set N of predictors with ground-truth accuracies $\boldsymbol{p}$ and estimated accuracies $\hat{\boldsymbol{p}}$ such that $\hat{p}_j \in (0, 1)$ for all $j \in N$, the probability of a classification error under the plug-in aggregator is lower bounded by:*

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, \hat{\boldsymbol{p}}, q) \geq 0.73 \exp\Big(-\frac{1}{2}\|\hat{\boldsymbol{w}}\|_2\Big) \sqrt{q(1-q)} \prod_{j \in N} \min\left(\frac{p_j}{\hat{p}_j}, \frac{1 - p_j}{1 - \hat{p}_j}\right) 2\sqrt{\hat{p}_j(1 - \hat{p}_j)} \tag{4.14}$$

where $q \equiv \mathbb{P}(y = +1)$ is the prior on the positive class.

Proof. In this proof we reuse some of the arguments from Theorems 4.1 and 3.4. Specifically, Equations 4.6, 4.7, 4.9 and 4.11 allow us to derive the following inequalities:

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, \hat{\boldsymbol{p}}, y = +1) \geq S \exp(-\frac{1}{2} w_q) \prod_{j \in N} \min\left(\frac{p_j}{\hat{p}_j}, \frac{1-p_j}{1-\hat{p}_j}\right) 2\sqrt{\hat{p}_j(1-\hat{p}_j)} \tag{4.15}$$

$$\mathbb{P}(\hat{y} \neq y|\boldsymbol{p}, \hat{\boldsymbol{p}}, y = -1) \geq S \exp(+\frac{1}{2} w_q) \prod_{j \in N} \min\left(\frac{p_j}{\hat{p}_j}, \frac{1-p_j}{1-\hat{p}_j}\right) 2\sqrt{\hat{p}_j(1-\hat{p}_j)} \tag{4.16}$$

where we define the shared term $\hat{S}$ as:

$$\hat{S} \equiv \frac{1}{2^{|N|}} \sum_{X} \Big(\mathbb{I}(\hat{h} < 0) + \frac{1}{2}\mathbb{I}(\hat{h} = 0)\Big) \exp(\hat{h}) \tag{4.17}$$

The shared term $\hat{S}$ is identical to the corresponding term S in Equation 3.19 for the known accuracies case, after replacing the exact weights $\boldsymbol{w}$ with the estimates $\hat{\boldsymbol{w}}$ in the definitions of h therein. From the discussion in Equations 3.20 and 3.21, we can derive that $\hat{S} \geq 0.365 \exp(-\frac{1}{2}\|\hat{\boldsymbol{w}}\|_2)$, which together with Equations 4.5, 4.15 and 4.16 yields the result in the theorem. □

Unfortunately, the two bounds in Theorems 4.1 and 4.2 do not match. This is in contrast with our analysis of the known accuracy case in Section 3.2.1, which yielded the same asymptotic terms for both the upper and lower bound. Instead, in the lower bound of Theorem 4.2 the minimisation term inside the product can be arbitrarily smaller than the corresponding sum in the upper bound of Theorem 4.1. The match is restored only if the estimates $\hat{\boldsymbol{p}}$ are close enough to their ground-truth values $\boldsymbol{p}$.

At the same time, this issue can be completely eliminated by introducing the extra assumption that we have access to unbiased estimates. An example of this is when we use the Bayesian estimator $\hat{p}_j^{BA} = \mathbb{E}_{p_j|O_j,\boldsymbol{g}}(p_j)$ we introduce in Section 2.2.1.3. In such circumstances, the following corollary holds:

Corollary 4.3. *Given a set N of predictors with unknown ground-truth accuracies $\boldsymbol{p}$ and estimated accuracies $\hat{\boldsymbol{p}}$, such that $\hat{p}_j = \mathbb{E}_{p_j|O_j,\boldsymbol{g}}(p_j)$ after observing the output O_j of each predictor $j \in N$ on the trials $\boldsymbol{g}$, the probability of a classification error under the plug-in aggregator is upper bounded by:*

$$\mathbb{P}(\hat{y} \neq y|O, \boldsymbol{g}, q) \leq \sqrt{q(1-q)} \prod_{j \in N} 2\sqrt{\hat{p}_j(1-\hat{p}_j)} \tag{4.18}$$

and lower bounded by:

$$\mathbb{P}(\hat{y} \neq y|O, \boldsymbol{g}, q) \geq 0.73 \exp\big(-\frac{1}{2}\|\hat{\boldsymbol{w}}\|_2\big) \sqrt{q(1-q)} \prod_{j \in N} 2\sqrt{\hat{p}_j(1-\hat{p}_j)} \tag{4.19}$$

where $q \equiv \mathbb{P}(y = +1)$ is the prior on the positive class.

Proof. To prove this corollary we notice that $\hat{p}_j$ depends solely on the observations O_j, the trial classes $\boldsymbol{g}$, and the underlying distribution of p_j. As a consequence, we can take Equation 4.9, condition it on O and $\boldsymbol{g}$ instead of $\hat{\boldsymbol{p}}$, and marginalise it over all possible values of p_j as follows:

$$\begin{aligned}\mathbb{P}(X|O,\boldsymbol{g},y=+1) &= \int_{\boldsymbol{p}} \mathbb{P}(X|\boldsymbol{p},O,\boldsymbol{g},y=+1)\mathbb{P}(\boldsymbol{p}|O,\boldsymbol{g})d\boldsymbol{p} \\ &= \exp(\hat{h}-\frac{1}{2}w_q)\prod_{j\in N}\sqrt{\hat{p}_j(1-\hat{p}_j)}\int_{p_j}\frac{\mathbb{P}(x_j|p_j,y=+1)}{f(x_j,\hat{p}_j,y)}\mathbb{P}(p_j|O_j,\boldsymbol{g})dp_j \\ &= \exp(\hat{h}-\frac{1}{2}w_q)\prod_{j\in N}\sqrt{\hat{p}_j(1-\hat{p}_j)} \end{aligned} \tag{4.20}$$

where the last equality follows from the fact that $\int_{p_j} p_j\mathbb{P}(p_j|O_j,\boldsymbol{g})dp_j$ is another way of writing $\mathbb{E}_{p_j|O_j,\boldsymbol{g}}(p_j)$, and thus numerator and denominator are both $\hat{p}_j$ or $1-\hat{p}_j$ depending on the sign of x_j. Similarly, when the ground-truth class is $y=-1$ we can marginalise Equation 4.11 to obtain the following:

$$\mathbb{P}(X|O,\boldsymbol{g},y=-1) = \exp(-\hat{h}+\frac{1}{2}w_q)\prod_{j\in N}\sqrt{\hat{p}_j(1-\hat{p}_j)} \tag{4.21}$$

As any attentive reader will notice, both the expressions in Equations 4.20 and 4.21 are almost identical to their counterparts for the known accuracy case in Equations 3.8 and 3.10. The only difference is that the ground-truth accuracies p_j are replaced with $\hat{p}_j$. Since this difference only informs how the weights $\boldsymbol{w}$ (respectively $\hat{\boldsymbol{w}}$) are created, our discussion on bounding the quantity $\sum_X \mathbb{I}(h<0)+\frac{1}{2}\mathbb{I}(h=0)$ for the known accuracy case remains valid for $\hat{h}$ too. The same can be said for $\sum_X \mathbb{I}(h>0)+\frac{1}{2}\mathbb{I}(h=0)$ when $y=-1$. As a consequence, we can use Equations 4.20 and 4.21 to extend the result of Theorems 3.1 and 3.4 to the unknown accuracy case as shown in the present corollary. □

The results in Corollary 4.3 form the basis for our following analysis of the UNI and WB policies.

4.2.2 Classification error under the UNI policy

As we do in the previous chapter, we start our analysis from the simplest collection strategy: the uniform (UNI) policy. With respect to the known accuracy case (see Section 3.2.2), the behaviour of this policy remains the same. That is, we use a round-robin allocation strategy to assign the same number of predictors to each item. More formally, we define the UNI policy as follows:

$$\hat{\pi}_{uni}(t) = \underset{i\in M\setminus M^{t-1}_{a(t)}}{\operatorname{argmin}}\left(|N_i^{t-1}|\right) \tag{4.22}$$

where ties are resolved in lexicographic order.

Barring any degenerate cases, the UNI policy assigns $R = T/|M|$ predictors per item, rounded to the nearest integer. Since the choice of predictors does not depend either on their accuracy, or on their output, we can analyse this behaviour as $|M|$ separate instances of the single-item bounds presented in Section 4.2.1. In order to make our presentation simpler, we assume here that the estimates $\hat{\boldsymbol{p}}$ are unbiased, and thus we refer to the bounds in Corollary 4.3 as shown in the following discussion. If this is not the case, it is just a matter of taking the expectation over both $\hat{\boldsymbol{p}}$ and $\boldsymbol{p}$ of the bounds in Theorems 4.1 and 4.2 instead (Manino et al., 2019a).

Corollary 4.4. *Given a population of predictors with unknown accuracy $p_j \sim f_p$ and corresponding estimates $\hat{p}_j = \mathbb{E}_{p_j|O_j,\boldsymbol{g}}(p_j)$ after observing the output O_j of each predictor $j \in N$ on the trials $\boldsymbol{g}$, a prior on the positive class $q \equiv \mathbb{P}(y_i = +1)$, and R predictors per item, the probability of a classification error by the UNI policy under the plug-in aggregator is upper bounded by:*

$$\mathbb{P}(\hat{y} \neq y | f_p, \boldsymbol{g}, q) \leq \sqrt{q(1-q)}\Big(2\mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\big(\sqrt{\hat{p}_j(1-\hat{p}_j)}\big)\Big)^R \tag{4.23}$$

and lower bounded by:

$$\mathbb{P}(\hat{y} \neq y | f_p, \boldsymbol{g}, q) \geq 0.73\sqrt{q(1-q)}\mathbb{E}_{\hat{\boldsymbol{p}}'|f_p,\boldsymbol{g}}\Big(\exp\big(-\frac{1}{2}\|\hat{\boldsymbol{w}}'\|_2\big)\prod_{j=1}^{R} 2\sqrt{\hat{p}_j(1-\hat{p}_j)}\Big) \tag{4.24}$$

for any item $i \in M$, where $\hat{\boldsymbol{p}}'$ is a vector of length R and $\hat{\boldsymbol{w}}'$ contains its corresponding weights $\hat{w}_j = \log\big(\hat{p}_j/(1-\hat{p}_j)\big)$.

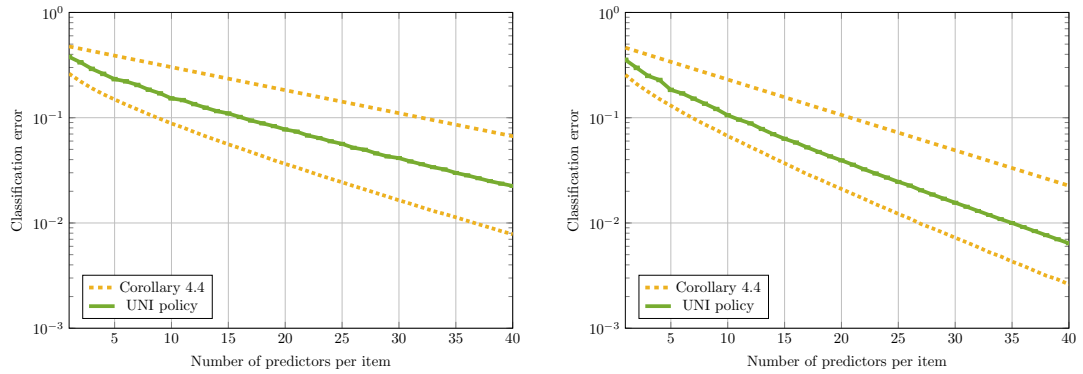
Proof. Corollary 4.3 provide us with bounds on the probability of a classification error given a specific set of predictors with output O_j on the trials $\boldsymbol{g}$, and corresponding estimated accuracies $\hat{\boldsymbol{p}}$. Since we know the underlying distribution f_p of the ground-truth accuracies $\boldsymbol{p}$, we can also compute the probability of observing a specific output O_j. Then, we can use the chain rule to derive the probability of an error for a random set of predictors as follows:

$$\mathbb{P}(\hat{y} \neq y | f_p, \boldsymbol{g}, q) = \mathbb{E}_{O|f_p,\boldsymbol{g}}\big(\mathbb{P}(\hat{y} \neq y | O, \boldsymbol{g}, q)\big) \tag{4.25}$$

Now, we know that the accuracy p_j and the output O_j of each predictor j is independently extracted. As a consequence the estimates $\hat{\boldsymbol{p}}$ are independent as well, and the result in the corollary follows. □

As for the known accuracy case in Section 3.2.2, both bounds in Corollary 4.4 share the same term $2\sqrt{\hat{p}_j(1-\hat{p}_j)}$. This term becomes dominant as R grows large, making the upper and lower bound match asymptotically.

We can see this phenomenon at play in the results of our empirical experiments in Figure 4.2. There, we run the UNI policy on synthetic data using the same settings of the corresponding results for the known accuracy case in Section 3.2.2. Moreover, we test the policy both with high and low uncertainty estimates $\hat{\boldsymbol{p}}$. In the former case, we test the accuracy of each predictor on just $|G| = 10$ trials, whereas in the latter we use an order of magnitude more ($|G| = 100$). Note how the UNI policy benefits from more accurate estimates $\hat{\boldsymbol{p}}$, as for each R its prediction error in Figure 4.2(b) is lower than that in Figure 4.2(a). In fact, as we increase the number of trials and $\hat{\boldsymbol{p}} \to \boldsymbol{p}$, the performance of the UNI policy, and the bounds in Corollary 4.4 converge to the known accuracy case portrayed in Figure 3.3. This is not an isolated phenomenon, as it happens for all the policies we analyse in this chapter.



(a) $|G| = 10$ trials per predictor

(b) $|G| = 100$ trials per predictor

FIGURE 4.2: Comparison between the empirical classification error under the UNI policy, the bounds in the existing literature and our results. The dashed line is the upper bound, the dotted one is the lower bound.

As a final note, there exists a closed form of the upper bound in Corollary 4.4 for a Beta prior. While it is of marginal interest here, it becomes more important in Chapter 5. For this reason, we include it separately in Appendix F.

4.2.3 Classification error under the WB policy

In the known accuracy case (see Section 3.2.3), we discussed how the UNI policy can be inefficient as it ignores the *quality* of the predictors in its allocation decisions. On the contrary, an optimal non-adaptive policy can achieve a lower prediction error with the same resources by distributing the predictors according to their accuracy. More formally, such policy can choose the composition of the subsets N_i for each item $i \in M$ so as to minimise the corresponding error bound.

When translating this idea to the present setting, we encounter a key issue. Without knowledge of the ground-truth accuracies $\boldsymbol{p}$, we are forced to trust the estimates $\hat{\boldsymbol{p}}$. For instance, we cannot use the general bounds in Theorems 4.1 and 4.2 because they contain

the term p_j. On the contrary, we can only assume that the estimates $\hat{p}_j$ are unbiased, and let our optimal non-adaptive policy minimise over the corresponding results in Corollary 4.3.

With this in mind, we can give an expression of the asymptotic performance of such an optimal policy. We do so in the following corollary:

Corollary 4.5. *Given a population of predictors with unknown accuracy $p_j \sim f_p$ and corresponding estimates $\hat{p}_j = \mathbb{E}_{p_j|O_j,\boldsymbol{g}}(p_j)$ after observing the output O_j of each predictor $j \in N$ on the trials $\boldsymbol{g}$, prior on the positive class $q \equiv \mathbb{P}(y_i = +1)$, and a large number of predictors per item R, the probability of a classification error by an optimal non-adaptive policy under the plug-in aggregator is:*

$$\mathbb{P}(\hat{y} \neq y|f_p, \boldsymbol{g}, q) = \exp\left(\frac{R}{2}\mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\Big(\log\big(4\hat{p}_j(1-\hat{p}_j)\big)\Big) + o\big(\sqrt{R}\big)\right) \tag{4.26}$$

for any item $i \in M$

Proof. This proof follows the same outline as the proof of Corollary 3.8. Accordingly, we can rewrite the bounds in Corollary 4.3 in exponential form as follows:

$$\mathbb{P}(\hat{y} \neq y|O, \boldsymbol{g}, q) = \exp\left(\frac{1}{2}\sum_{j\in N_i}\log\big(4\hat{p}_j(1-\hat{p}_j)\big) + o\big(\sqrt{|N_i|}\big)\right) \tag{4.27}$$

Then, we claim that an optimal non-adaptive policy would try to redistribute the predictors in such a way that the sum in Equation 4.27 has the same value for each item $i \in M$. We denote the total amount that we can distribute during the collection process as follows:

$$\hat{S}_{tot} = \frac{L|N|}{2}\mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\Big(\log\big(4\hat{p}_j(1-\hat{p}_j)\big)\Big) \tag{4.28}$$

where we assume that $|M_j| = L$ for each predictor j. In the best case scenario, the optimal policy manages to split $\hat{S}_{tot}$ evenly over M. Thus, $L|N|/|M| = R$ and we get the result in the corollary. □

The difference between the result in Corollary 4.5 and the corresponding result for the UNI policy in Corollary 4.4 is that now the expectation over $\hat{p}_j$ lies outside of the logarithm. According to Jensen's inequality, this guarantees that the optimal non-adaptive policy has a larger decay in the probability of error as R increases, since:

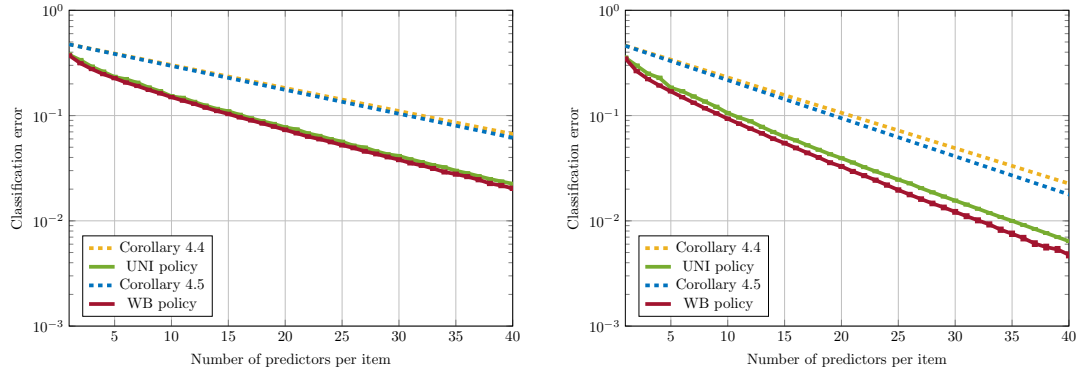
$$\mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\Big(\log\big(2\sqrt{\hat{p}_j(1-\hat{p}_j)}\big)\Big) \leq \log\Big(\mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\big(2\sqrt{\hat{p}_j(1-\hat{p}_j)}\big)\Big) \tag{4.29}$$

However, note that this optimal policy is only an abstract construct. In order to have a practical implementation of it, we turn again to the greedy strategy depicted in Section 3.2.3 for the known accuracy case. There, we define a weight balancing (WB) policy

that aims to distribute the quantity S_{tot} as evenly as possible. We can apply the same approach to $\hat{S}_{tot}$ in the following way:

$$\hat{\pi}_{wb}(t) = \underset{i \in M \setminus M_{a(t)}^{t-1}}{\operatorname{argmax}} \Big(\sum_{j \in N_i^{t-1}} \log\left(4\hat{p}_j(1-\hat{p}_j)\right) \Big) \tag{4.30}$$

In order to get a qualitative understanding of the improvement the WB policy brings over the UNI policy, we compare their performance on synthetic data. The results in Figure 4.3 show that the advantage of the WB policy is minimal, similarly to the known accuracy case in Section 3.2.3. This is true both for the empirical performance, as well as the theoretical bound. As noted for the experiments in Figure 4.2, the behaviour of the policies converges to the known accuracy case (see Figure 3.4) as the estimates $\hat{\boldsymbol{p}}$ become more accurate (larger number of trials).



(a) $|G| = 10$ trials per predictor

(b) $|G| = 100$ trials per predictor

FIGURE 4.3: Comparison between the empirical classification error under the WB and UNI policies and their corresponding upper bounds.

All things considered, the WB policy is usually not worth the added complexity. If the circumstances allow for a more complex collection strategy, one of the adaptive policies we analyse in the next Section 4.3 would provide better performance.

4.3 Adaptive collection policies

In this section, we analyse the performance of adaptive collection policies. Borrowing our definition from the known accuracy case (see Section 3.3), an adaptive policy is a collection strategy that chooses the next item i to label depending on all the data X^t collected so far. As a consequence, the behaviour of these policies tend to be more complex than that of non-adaptive policies. Fortunately, the tools we introduce for the known accuracy case translate well to the present scenario.

Still, there are a few technicalities that we need to cover. For this reason, we first introduce some general results on classifying a single item in Section 4.3.1. Then, we

use these to analyse the US policy in Section 4.3.2. Finally, we prove the equivalence between the US and IG policies in Section 4.3.3. For the reasons described in Section 3.3.4, we drop the LOS policy from our discussion.

4.3.1 Achieving a target classification error on a single item

As per Section 4.2.1, before dealing with the specifics of the individual adaptive policies, we derive some general results for a single item. This allows us to reduce the complexity of our analysis and use milder assumptions. However, in this section we reverse the argument we use for non-adaptive policies. That is, we fix the target classification error $\hat{p}_e$ we want and assume that we can collect as much data we need to achieve it. In contrast with our analysis for the known accuracy case (see Section 3.3.1), the target $\hat{p}_e$ is the *reported* probability of a classification error. This is because at runtime we only have access to the estimated log-odds $\hat{\boldsymbol{z}}$, and we do not know the actual probability of an error.

With this in mind, we can put an upper bound on the expected number of predictors we need to reach $\hat{p}_e$ as follows:

Theorem 4.6. *Given a population of predictors with unknown accuracy $p_j \sim f_p$, corresponding estimates $\hat{p}_j \sim f_{\hat{p}}$ and prior on the positive class $q \equiv \mathbb{P}(y_i = +1)$, the expected number $n \equiv |N|$ of predictors we need to lower the reported classification error below $\hat{p}_e$ is upper bounded by:*

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}},y|q,\hat{p}_e}(n) < \frac{-\hat{w}_e + (1-2q)w_q}{\mathbb{E}_{p_j,\hat{p}_j}\big((2p_j-1)\hat{w}_j\big)} + 1 \tag{4.31}$$

where $\hat{w}_e$, w_q and w_j are the log-odds of $\hat{p}_e$, q and $\hat{p}_j$ respectively.

Proof. This proof follows the same outline as the corresponding one for the known accuracy case in Theorem 3.9. First, let us define $\hat{z}^0 = w_q$ as the starting point of the random walk over the estimated log-odds. Similarly, let us define $\hat{z}^n$ as its final value after collecting the labels of $n \equiv |N|$ predictors. By definition we have $\hat{z}^n \notin (\hat{w}_e, -\hat{w}_e)$, as we collect labels until the value of $\hat{z}^t$ reaches or crosses the threshold. In this regard, the expected number of predictors we need to do so depends on the ground-truth class y as follows:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}},y|q,\hat{p}_e}(n) = q\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=+1,\hat{p}_e}(n) + (1-q)\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=-1,\hat{p}_e}(n) \tag{4.32}$$

At the same time, we can link the expected value of n to the average drift of the random walk. In fact, according to Wald's equation (Wald, 1944), each step $\hat{s}_j = x_j\hat{w}_j$ contributes in expectation to the value of $\hat{z}^n$ as follows:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y,\hat{p}_e}(\hat{z}^n) = \mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y,\hat{p}_e}(n)\mathbb{E}_{x_j,p_j,\hat{p}_j|y}(\hat{s}_j) + w_q \tag{4.33}$$

Moreover, we can bound the expected value of $\hat{z}^n$ with the threshold $\hat{w}_e$ as follows. First, recall that the steps $\hat{s}_j$ are independent. As a consequence, we have:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y,\hat{p}_e}(\hat{z}^n) = \mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y,\hat{p}_e}(\hat{z}^{n-1}) + \mathbb{E}_{x_j,p_j,\hat{p}_j|y}(\hat{s}_j) \tag{4.34}$$

Second, we know that $\hat{z}^{n-1} \in (\hat{w}_e, -\hat{w}_e)$, since $\hat{z}^{n-1}$ are the log-odds just before crossing the threshold. By combining this observation with Equation 4.34, we get the following inequalities:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=+1,\hat{p}_e}(\hat{z}^n) < -\hat{w}_e + \mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}(\hat{s}_j) \tag{4.35}$$

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=-1,\hat{p}_e}(\hat{z}^n) > \hat{w}_e + \mathbb{E}_{x_j,p_j,\hat{p}_j|y=-1}(\hat{s}_j) \tag{4.36}$$

Next, we can use Equation 4.33 to eliminate the variable $\hat{z}^n$ from Equations 4.35 and 4.36:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=+1,\hat{p}_e}(n)\mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}(\hat{s}_j) < -\hat{w}_e + \mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}(\hat{s}_j) - w_q \tag{4.37}$$

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=-1,\hat{p}_e}(n)\mathbb{E}_{x_j,p_j,\hat{p}_j|y=-1}(\hat{s}_j) > \hat{w}_e + \mathbb{E}_{x_j,p_j,\hat{p}_j|y=-1}(\hat{s}_j) - w_q \tag{4.38}$$

Finally, we notice that the expected value of the step $\hat{s}_j$ changes its sign, but not its magnitude, when we swap $y = +1$ with $y = -1$. Thus, we can plug Equations 4.37 and 4.38 into Equation 4.32, and compute the following bound:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}},y|q,\hat{p}_e}(n) < \frac{-\hat{w}_e + \mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}(\hat{s}_j) - w_q(2q-1)}{\mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}(\hat{s}_j)} \tag{4.39}$$

The bound in the theorem follows from writing the expected value of $\hat{s}_j = x_j\hat{w}_j$ in explicit form. □

Deriving a lower bound is more difficult. The main issue is that we need a relationship between the estimated log-odds $\hat{\boldsymbol{z}}$ and the probability of a classification error in our proof. For this reason, we introduce the extra assumption that the estimates of the predictors' accuracy $\hat{\boldsymbol{p}}$ are unbiased. According to our discussion in Section 4.2.1, this assumption suffices to restore to relationship we need, and establishes the following result:

Theorem 4.7. *Given a population of predictors with unknown accuracy $p_j \sim f_p$, corresponding estimates $\hat{p}_j = \mathbb{E}_{p_j|O_j,\boldsymbol{g}}(p_j)$ after observing the output O_j of each predictor $j \in N$ on the trials $\boldsymbol{g}$, and prior on the positive class $q \equiv \mathbb{P}(y_i = +1)$, the expected number $n \equiv |N|$ of predictors we need to lower the reported classification error below $\hat{p}_e$ is lower bounded by:*

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}},y|q,\hat{p}_e}(n) < \frac{-\hat{w}_e + (1-2q)w_q - 0.56 - \mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\big(|\hat{w}_j|\big)}{\mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\big((2\hat{p}_j - 1)\hat{w}_j\big)} \tag{4.40}$$

where $\hat{w}_e$, w_q and w_j are the log-odds of $\hat{p}_e$, q and $\hat{p}_j$ respectively.

Proof. In this proof we reuse some of the arguments from the corresponding result for the known accuracy case in Theorem 3.10, and the upper bound in Theorem 4.6. We begin by establishing a lower bound on the expected value of n when the ground-truth class is $y = +1$. By definition, we know that the magnitude of $\hat{z}^n$ is larger or equal to the threshold $\hat{w}_e$. Thus:

$$\begin{aligned} -\hat{w}_e &\leq \mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=+1,\hat{p}_e}\big(|\hat{z}^n|\big) \\ &= \mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=+1,\hat{p}_e}(\hat{z}^n) - 2\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=+1,\hat{p}_e}\big(\hat{z}^n\mathbb{I}(\hat{z}^n < 0)\big) \end{aligned} \tag{4.41}$$

where the last equality comes from the definition of absolute value. Now, we can substitute Equation 4.33 into Equation 4.41, which yields the following inequality:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=+1,\hat{p}_e}(n)\mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}(\hat{s}_j) \geq -\hat{w}_e - w_q + 2\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=+1,\hat{p}_e}\big(\hat{z}^n\mathbb{I}(\hat{z}^n < 0)\big) \tag{4.42}$$

Furthermore, we can apply the same arguments we use in Equations 3.48 and 3.49 to the last addend in Equation 4.42. In doing so, we get the following inequality:

$$\begin{aligned} \mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=+1,\hat{p}_e}\big(\hat{z}^n\mathbb{I}(\hat{z}^n < 0)\big) &\geq \hat{w}_e\mathbb{P}(\hat{z}^n \leq \hat{w}_e|y = +1) + \mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}\big(\hat{s}_j\mathbb{I}(\hat{s}_j < 0)\big) \\ &\geq -0.28 - \frac{1}{2}\mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}\big(|\hat{s}_j|\big) \end{aligned} \tag{4.43}$$

since the assumptions of the present theorem ensures that $\mathbb{P}(\hat{z}^n \leq \hat{w}_e) = \text{sig}(\hat{w}_e)$, and thus the same arguments of Equation 3.50 apply here (see discussion in Section 4.1). Then, by substituting Equation 4.43 into Equation 4.41 we have a lower bound on n for $y = +1$:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=+1,\hat{p}_e}(n) < \frac{-\hat{w}_e - w_q - 0.56 - \mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}\big(|\hat{s}_j|\big)}{\mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}(\hat{s}_j)} \tag{4.44}$$

The corresponding bound for $y = -1$ can be derived in a similar fashion. First, we exchange the definition of absolute value in Equation 4.41 for the following equivalent expression:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=-1,\hat{p}_e}\big(|\hat{z}^n|\big) = \mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=-1,\hat{p}_e}(-\hat{z}^n) + 2\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=-1,\hat{p}_e}\big(\hat{z}^n\mathbb{I}(\hat{z}^n > 0)\big) \tag{4.45}$$

Second, we combine Equation 4.45 with Equation 4.33 to derive the following inequality:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=-1,\hat{p}_e}(n)\mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}(\hat{s}_j) \geq -\hat{w}_e + w_q - 2\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=-1,\hat{p}_e}\big(\hat{z}^n\mathbb{I}(\hat{z}^n > 0)\big) \tag{4.46}$$

since the expected value of $\hat{s}_j$ changes in sign but not magnitude with y. Finally, we bound the expected value of a positive $\hat{z}^n$ with the same procedure shown in Equation 4.43 (up to a change in sign), which yields the following result:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y=-1,\hat{p}_e}(n) < \frac{-\hat{w}_e + w_q - 0.56 - \mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}\big(|\hat{s}_j|\big)}{\mathbb{E}_{x_j,p_j,\hat{p}_j|y=+1}(\hat{s}_j)} \tag{4.47}$$

Then, we notice that we can write the expected value of the steps $\hat{s}_j = x_j\hat{w}_j$ in terms of

the estimates $\hat{p}_j$ only. This is because we have $\mathbb{E}_{x_j,\hat{p}_j|y=+1}(x_j\hat{w}_j) = \mathbb{E}_{p_j,\hat{p}_j|y=+1}\big((2p_j - 1)\hat{w}_j\big)$, and $\mathbb{E}_{p_j|\hat{p}_j}(p_j) = \hat{p}_j$ as per the assumptions of the present theorem. Finally, by substituting Equations 4.44 and 4.47 into Equation 4.32 we get the result in the theorem. □

The results of Theorem 4.31 and 4.40 can be easily extended to the multi-item case. We do so in the next section, where we analyse the performance of the US policy.

4.3.2 Classification error under the US policy

The US policy can be easily implemented even if we only have access to the estimated posterior $\hat{\boldsymbol{z}}^t$. In fact, we can greedily choose the item whose log-odds are closest to zero:

$$\hat{\pi}_{us}(t) = \underset{i \in M \setminus M^{t-1}_{a(t)}}{\operatorname{argmin}} \big(|\hat{z}^t_i|\big) \tag{4.48}$$

where ties are broken in lexicographic order. Note that Equation 4.48 optimises over the *estimated* uncertainty. This differs from our definition in Section 3.3.2 where we have access to the actual posterior probability through the log-odds $\boldsymbol{z}^t$.

As a consequence, we lose the link between the value of the log-odds $\hat{z}^t_i$ and the probability of a classification error on that item. Such a link can be restored if we assume that all $\hat{p}$ are unbiased estimates of their ground-truth value p_j (see discussion in Section 4.2.1). In the following corollaries we work under this assumption, which allows us to derive very similar results to those for the known accuracy case in Section 3.3.2. However, bear in mind that the unbiasedness assumption is not strictly necessary. Milder assumptions that create a relationship between $\hat{z}^t_i$ and $\mathbb{P}(\hat{y}_i \neq y_i)$ can suffice. Since they are not needed for the discussion in this chapter, we explore them in Chapter 5.

At the same time, the runtime behaviour of the US policy does not depend on how the predictors' weights $\hat{\boldsymbol{w}}$ are computed. Accordingly, all of our consideration from Section 3.3.2 still applies. Additionally, we need to take into account the same non-ideality issues discussed in Section 3.4. With this in mind, we can bound its probability of a classification error as follows:

Corollary 4.8. *Given a population of predictors with unknown accuracy $p_j \sim f_p$ and corresponding estimates $\hat{p}_j = \mathbb{E}_{p_j|O_j,\boldsymbol{g}}(p_j)$ after observing the output O_j of each predictor $j \in N$ on the trials $\boldsymbol{g}$, a prior on the positive class $q \equiv \mathbb{P}(y_i = +1)$, a large number of items $|M| \to \infty$, and an average number of predictors per item R, the probability of a classification error by the US policy under the plug-in aggregator is upper bounded by:*

$$\mathbb{P}(\hat{y} \neq y | f_p, \boldsymbol{g}, q) \leq \exp\big(-(R-1)\mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\big((2\hat{p}_j - 1)\hat{w}_j\big) - (2q-1)w_q\big) \tag{4.49}$$

for all items $i \in M$.

Proof. Let us assume for the moment that we can collect as many data points T as we want. Under this assumption, we can run $|M|$ independent random walks on the estimated log-odds $\hat{z}_i$ of each item $i \in M$, until they all cross the threshold $\pm\hat{w}_e$ we choose. In this respect, Theorem 4.6 gives us the average number of predictors we need for each item. From there, we can derive the following expression:

$$-\hat{w}_e > \Big(\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}},y|q,\hat{p}_e}\big(|N_i|\big) - 1\Big)\mathbb{E}_{p_j,\hat{p}_j}\big((2p_j - 1)\hat{w}_j\big) + (2q-1)w_q \tag{4.50}$$

where, given the assumption of unbiased estimates $\hat{\boldsymbol{p}}$, we can simplify the expected value $\mathbb{E}_{p_j,\hat{p}_j}\big((2p_j-1)\hat{w}_j\big)$ to $\mathbb{E}_{\hat{p}_j}\big((2\hat{p}_j-1)\hat{w}_j\big)$ as we do in the proof of Theorem 4.7.

At the same time, when the number of items $|M|$ grows large, we know that the number of predictors we need on each item will converge to its expected value (for a thourough discussion on this, see the proof of Corollary 3.11). As a result, we can substitute $\mathbb{E}\big(|N_i|\big)$ in Equation 4.50, with the actual average number of predictors R we have access to. Furthermore, under the assumptions of this corollary, we can tie the value of the log-odds $\hat{z}_i^T$ at the end of the collection process, with the value of the threshold $\hat{w}_e$ as follows:

$$\begin{aligned}\mathbb{P}(\hat{y} \neq y|f_{\hat{p}}, q) &= \mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}},y|q}\Big(\operatorname{sig}\big(-|\hat{z}_i^T|\big)\Big)\\ &\leq \operatorname{sig}\big(-|\hat{w}_e|\big)\\ &\leq \exp(\hat{w}_e)\end{aligned} \tag{4.51}$$

The result in the present corollary follows from subsituting Equation 4.50 into Equation 4.51, with $\mathbb{E}\big(|N_i|\big) = R$. □

Corollary 4.9. *Given a population of predictors with unknown accuracy $p_j \sim f_p$ and corresponding estimates $\hat{p}_j = \mathbb{E}_{p_j|O_j,\boldsymbol{g}}(p_j)$ after observing the output O_j of each predictor $j \in N$ on the trials $\boldsymbol{g}$, a prior on the positive class $q \equiv \mathbb{P}(y_i = +1)$, a large number of items $|M| \to \infty$, and an average number of predictors per item R, the probability of a classification error by the US policy under the plug-in aggregator is lower bounded by:*

$$\begin{aligned}\mathbb{P}(\hat{y} \neq y|f_p, \boldsymbol{g}, q) \geq \exp\big(&-R\mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\big((2\hat{p}_j-1)\hat{w}_j\big) - (2q-1)w_q\\ &- 1.25 - \mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\big(|\hat{w}_j|\big)\big)\end{aligned} \tag{4.52}$$

for all items $i \in M$.

Proof. This proof follows the same outline as the proof of Corollary 3.12. Similarly to the proof of Corollary 4.8 we assume for the moment that we can collect as many data points T as we want. Under this condition, the techniques we used in Theorem 4.7 are valid. We use them to bound the expected value of the absolute log-odds $|\hat{z}_i^T|$ at the end of the collection process:

$$\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y_i,\hat{p}_e}\big(|\hat{z}_i^T|\big) \leq y_i\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}}|y_i,\hat{p}_e}(\hat{z}_i^T) + 0.56 + \mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\big(|\hat{w}_j|\big) \tag{4.53}$$

We also know that we can compute $\mathbb{E}(\hat{z}_i^T)$ according to Equation 4.33, and that we can take the expectation of Equation 4.53 over y_i. Together, these two operations yield the following result:

$$\begin{aligned}\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}},y_i|q,\hat{p}_e}\big(|\hat{z}_i^T|\big) \leq \mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}},y|q,\hat{p}_e}\big(|N_i|\big)\mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\big((2\hat{p}_j-1)\hat{w}_j\big) \\ + (2q-1)w_q + 0.56 + \mathbb{E}_{\hat{p}_j|f_p,\boldsymbol{g}}\big(|\hat{w}_j|\big)\end{aligned} \tag{4.54}$$

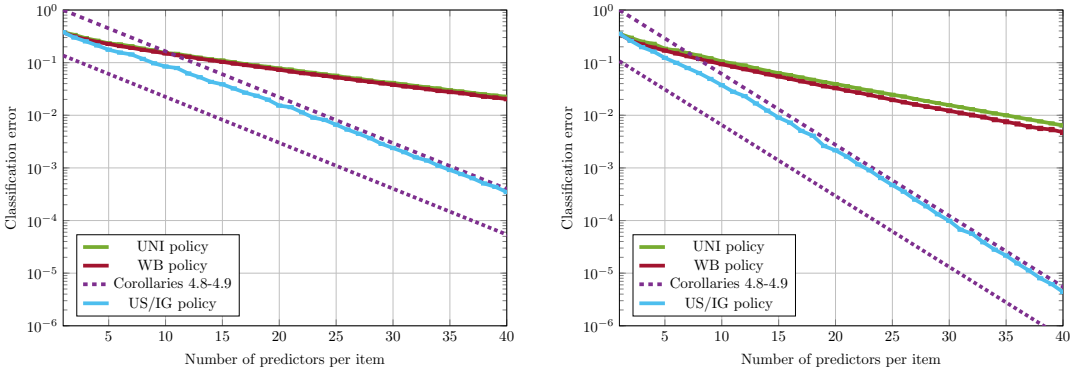
However, for the reasons explained in Corollary 3.11, for $|M| \to \infty$ the number of predictors per item converges to its expected value. Therefore, we can replace $\mathbb{E}\big(|N_i|\big)$ with the actual number of available predictors R in Equation 4.54. Additionally, given the assumptions in the present corollary, we have the following relationship between the probability of an error and the absolute log-odds $|\hat{z}_i^T|$:

$$\begin{aligned}\mathbb{P}(\hat{y} \neq y|f_{\hat{p}}, q) &= \mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}},y|q}\Big(\operatorname{sig}\big(-|\hat{z}_i^T|\big)\Big) \\ &\geq \operatorname{sig}\Big(-\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}},y|q}\big(|\hat{z}_i^T|\big)\Big) \\ &\geq \frac{1}{2}\exp\Big(-\mathbb{E}_{X,\boldsymbol{p},\hat{\boldsymbol{p}},y|q}\big(|\hat{z}_i^T|\big)\Big)\end{aligned} \tag{4.55}$$

The result in the present corollary follows from subsituting Equation 4.54 into Equation 4.55, with $\mathbb{E}\big(|N_i|\big) = R$. □

Corollaries 4.8 and 4.9 highlight two desired properties of the US policy. First, the two bounds match asymptotically in R as they share the same constant $\mathbb{E}_{\hat{p}_j}\big((2\hat{p}_j-1)\hat{w}_j\big)$. This is the average drift of the random walk on the log-odds $\hat{z}_i$, and thus ensures that the US policy is optimal. That is, there is no collection policy that can achieve a better asymptotic rate. Second, the two bounds converge to the corresponding results for the known accuracy case in Corollaries 3.11 and 3.12 as the estimates $\hat{\boldsymbol{p}}$ converge to their ground-truth value $\boldsymbol{p}$. This ensures that the US policy achieves similar performance to its known-accuracy counterpart when high-quality estimates $\hat{\boldsymbol{p}}$ are provided.

We test the latter property in a set of empirical experiments on synthetic data. In order to make the comparison easier, we use the same setting of the corresponding experiments for the known accuracy case in Section 3.3.2. Additionally, we run the same experiments twice: first in a high uncertainty regime, where the estimates $\hat{\boldsymbol{p}}$ are computed after only $|G| = 10$ trials, and then in the low uncertainty regime, with $|G| = 100$ instead. The results are reported in Figures 4.4(a) and 4.4(b) respectively. There, it is clear that having more accurate estimates $\hat{\boldsymbol{p}}$ improves the performance of all policies. Moreover, if we compare these two plots with Figure 3.6, we can see that for $|G| \to \infty$ we converge to the known accuracy case. Finally, the relative performance of all collection policy does not differ from the known accuracy case, even in the high uncertainty regime. We formally study this phenomenon in Section 4.4.



(a) $|G| = 10$ trials per predictor

(b) $|G| = 100$ trials per predictor

FIGURE 4.4: Comparison between the empirical classification error under the US/IG policies, and the bounds we derive in Corollaries 4.8 and 4.9. The dashed line is the upper bound, the dotted one is the lower bound. For reference we include the UNI and WB policies as well.

4.3.3 Classification error under the IG policy

The amount of information carried by each new data point x_{ij} we observe can be measured in terms of the Kullback-Leibler (KL) divergence (see Section 2.3.3). Similarly to our derivation for the known accuracy case in Section 3.3.3, we can apply the KL divergence to our setting as follows:

$$KL(\hat{\boldsymbol{z}}^{t+1} \parallel \hat{\boldsymbol{z}}^{t}) = \text{sig}(\hat{z}_i^t + x_{ij}\hat{w}_j)x_{ij}\hat{w}_j + \log\left(\frac{\text{sig}(-\hat{z}_i^t - x_{ij}\hat{w}_j)}{\text{sig}(-\hat{z}_i^t)}\right) \tag{4.56}$$

Of course, the value of the new data point x_{ij} can be observed only *after* we collect it. Therefore, our decision on which item i to collect it on must be based on the *expected* information gain of the future label. We can compute this expectation under the assumption that $\hat{p}_j$ is an unbiased estimate of the probability of observing a correct label. This is similar to what we do in Section 4.2.3 for the WB policy. With this assumption, $\hat{z}_i^t$ is an unbiased estimate of the log-odds z_i^t and the expectation of Equation 4.56 is the following:

$$\begin{aligned}\mathbb{E}_{x_{ij}|\hat{\boldsymbol{z}}^t,\hat{p}_j}\left(KL(\hat{\boldsymbol{z}}^{t+1} \parallel \hat{\boldsymbol{z}}^{t})\right) = \hat{w}_j(2\hat{p}_j - 1)\,\text{sig}(\hat{z}_i^t) - \log\left(\text{sig}(-\hat{z}_i^t)\right) \\ + \mathbb{P}(x_{ij} = +1|\hat{z}_i^t, \hat{p}_j)\log\left(\text{sig}(-\hat{z}_i^t - \hat{w}_j)\right) \\ + \mathbb{P}(x_{ij} = -1|\hat{z}_i^t, \hat{p}_j)\log\left(\text{sig}(-\hat{z}_i^t + \hat{w}_j)\right)\end{aligned} \tag{4.57}$$

Which leads to the following formal definition of the IG policy for the unknown accuracy case:

$$\hat{\pi}_{ig}(t) = \underset{i \in M \setminus M_{a(t)}^{t-1}}{\text{argmax}} \left(\mathbb{E}_{x_{ij}|\hat{\boldsymbol{z}}^t,\hat{p}_j}\left(KL(\hat{\boldsymbol{z}}^{t+1} \parallel \hat{\boldsymbol{z}}^{t})\right)\right) \tag{4.58}$$

where ties are broken in lexicographic order.

Now, the definition of the IG policy in Equation 4.58 almost matches the corresponding one for the known accuracy case (see Equation 3.69). The only difference is that we replaced the ground-truth accuracies $\boldsymbol{p}$ with their unbiased estimates $\hat{\boldsymbol{p}}$. As a result, our equivalence result with the US policy presented in Theorem 3.13 still applies:

Theorem 4.10. *For any values of the estimated log-odds $\hat{\boldsymbol{z}}^t$, and any predictor j with unbiased accuracy estimate $\hat{p_j} = \mathbb{E}_{p_j|O_j,\boldsymbol{g}}(p_j) \neq \frac{1}{2}$, the US and IG policies select the same item, that is $\hat{\pi}_{us}(t) = \hat{\pi}_{ig}(t)$.*

Proof. This proof follows the same outline of the corresponding result for the known accuracy case in Theorem 3.13. Recall that the US policy (see Equation 4.48) optimises over the function $|\hat{z}_i^t|$. Crucially, this function is even and has its global minimum in $\hat{z}_i^t = 0$. In order to prove our equivalence result, we need to show that Equation 4.57 has similar properties. The only difference is that the IG policy maximises rather than minimises over its objective funtion.

First, note that Equation 4.57 is even both in $\hat{z}_i^t$ and $\hat{w}_j$, as can be easily proven by substitution. Thus, in the following discussion we only focus on the positive quadrant, since the other ones have identical properties. Accordingly, it suffices to show that Equation 4.57 is monotonically decreasing in $\hat{z}_i^t > 0$. We do so by taking its derivative and proving that its sign is always negative (see Equations 3.70, 3.71, 3.72 and 3.73 for a similar derivation):

$$\begin{aligned} &\frac{d}{d\hat{z}_i^t}\Big\{\mathbb{E}_{x_{ij}|\hat{\boldsymbol{z}}^t,\hat{p}_j}\big(KL(\hat{\boldsymbol{z}}^{t+1} \parallel \hat{\boldsymbol{z}}^t)\big)\Big\} \\ &\qquad = (2\hat{p}_j - 1)\,\mathrm{sig}(\hat{z}_i^t)\,\mathrm{sig}(-\hat{z}_i^t)\left(\hat{w}_j + \log\left(\frac{\mathrm{sig}(-\hat{z}_i^t - \hat{w}_j)}{\mathrm{sig}(-\hat{z}_i^t + \hat{w}_j)}\right)\right) \end{aligned} \tag{4.59}$$

which can be proven to be negative for all $\hat{z}_i^t > 0$ and $\hat{w}_j \neq 0$ by contradiction in the same way as Equation 3.74. Finally, for $\hat{w}_j \neq 0$ the derivative in Equation 4.59 has its only zero in $\hat{z}_i^t = 0$, proving that the estimated expected information gain has a global maximum in that location. □

The result in Theorem 4.10 can be easily overlooked, given that it differs from its known-accuracy counterpart only slightly. However, there is one condition required for it to hold, which becomes crucial in the next Chapter 5. That is, we assume that the estimates $\hat{\boldsymbol{p}}$ are given, and do not depend on the set of data X we collect. As we progress further in our analysis of unknown accuracy models, we remove the aforementioned assumption, thus restoring the IG policy to its role of fully independent collection policy. An example of this is presented in Section 5.3.1. In contrast, throughout the present chapter the equivalence between the US and IG policies always holds, and therefore we can use the results in Corollaries 4.8 and 4.9 to analyse the performance of the latter.

4.4 Asymptotic policy comparison

Now that we have analysed each collection policy individually, it is time to compare their performance. As per Section 3.5 for the known accuracy case, we present here an *asymptotic* comparison. This is because, as the average number of items per predictor R grows large, the probability of an error exhibits the following exponential decay under all policies:

$$\mathbb{P}(\hat{y} \neq y) = \exp\big(- \hat{c}_\pi R + o(\sqrt{R})\big) \tag{4.60}$$

where the constant $\hat{c}_\pi$ depends on the particular collection policy we use. Thanks to the results in Corollaries 4.4, 4.8, 4.9 and Theorems 4.5, 4.10 we can quantify the value of such a constant for the UNI policy, an optimal non-adaptive policy and both US/IG adaptive policies as follows:

$$\hat{c}_{uni} = -\log\Big(2\mathbb{E}_{\hat{p}_j}\big(\sqrt{\hat{p}_j(1-\hat{p}_j)}\big)\Big) \tag{4.61}$$

$$\hat{c}_{nada} = -\frac{1}{2}\mathbb{E}_{\hat{p}_j}\Big(\log\big(4\hat{p}_j(1-\hat{p}_j)\big)\Big) \tag{4.62}$$

$$\hat{c}_{ada} = \mathbb{E}_{\hat{p}_j}\Big((2\hat{p}_j - 1)\log\Big(\frac{\hat{p}_j}{1-\hat{p}_j}\Big)\Big) \tag{4.63}$$

It is worth noting that the value of $\hat{c}_\pi$ for the policies analysed in this chapter is identical to the known accuracy case in so far as we replace p_j with its estimate $\hat{p}_j$ (see Equations 3.81, 3.82 and 3.83). As a consequence, we can expect the value of $\hat{c}_\pi$ to approach c_π as the estimates of the predictors' accuracy become more accurate. Additionally, we can derive a similar result to Theorem 3.15 about the advantage of using an adaptive policy:

Theorem 4.11. *Given a population of predictors with unknown accuracy $p_j \sim f_p$, corresponding estimates $\hat{p}_j = \mathbb{E}_{p_j|O_j,\boldsymbol{g}}(p_j)$ such that $\mathbb{P}(\hat{p}_j \neq \frac{1}{2}) > 0$, a large number of items $|M| \to \infty$, and a large number of predictors per item $R \to \infty$, the asymptotic rate of the US and IG policies is at least twice as large as that of any non-adaptive policy, that is $\hat{c}_{ada} \geq 2\hat{c}_{nada}$.*

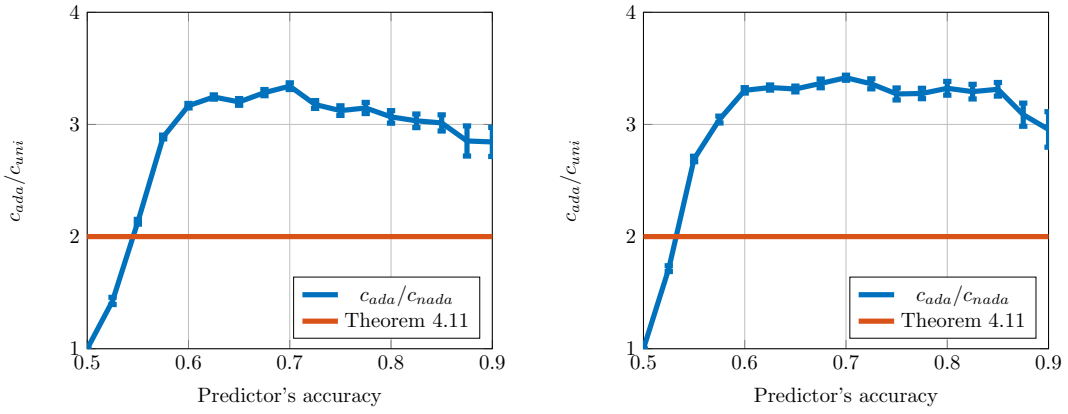
Proof. Thanks to Jensen's inequality, we know that $\hat{c}_{uni} \leq \hat{c}_{nada}$. Furthermore, a quick inspection of the arguments of the expected values in Equations 4.62 and 4.63 reveals that:

$$-\log\big(4\hat{p}_j(1-\hat{p}_j)\big) \leq (2\hat{p}_j - 1)\log\Big(\frac{\hat{p}_j}{1-\hat{p}_j}\Big) \qquad \text{for all} \qquad \hat{p}_j \in (0,1) \tag{4.64}$$

The result in the present theorem follows from these considerations and the monotonicity property of the expectation operator. □

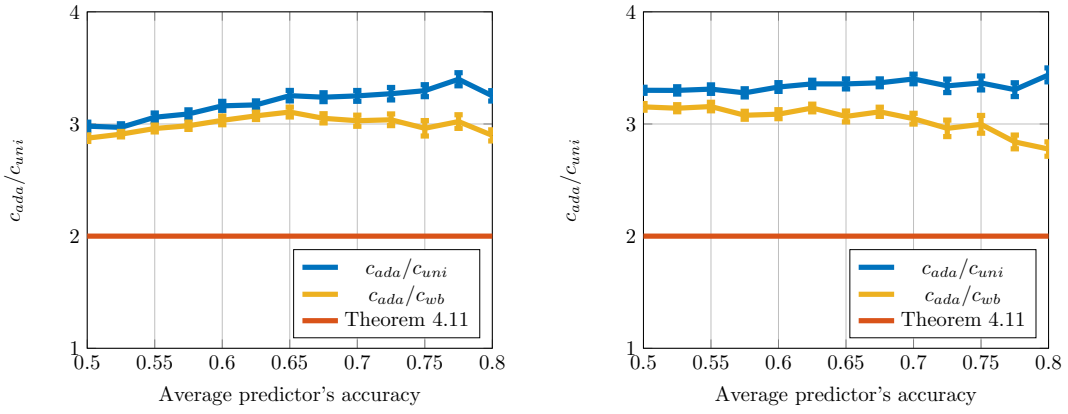
At the same time, Theorem 4.11 is based on the assumption that the estimates $\hat{p}_j$ are unbiased. In some instances, this can be impossible to guarantee, for example if we

have no prior information on the distribution of the predictors' accuracy f_p. In such circumstances, it is customary to replace the missing information with an uninformative prior, e.g. assuming a beta prior and setting $\alpha = 1, \beta = 1$ (Murphy, 2012). In order to explore what happens to the value of the ratio $\hat{c}_{ada}/\hat{c}_{nada}$ under such circumstances, we ran synthetic experiments with a variety of choices for f_p, and estimate the asymptotic ratio between the policies empirically. To make the comparison with the known accuracy case easier, we use the same settings of the corresponding experiments presented in Section 3.5.



(a) $L = 10$ labels per predictor. (b) $L = 100$ labels per predictor.

FIGURE 4.5: Empirical advantage of adaptive policies with identical predictors $p_j = \bar{p}$.



(a) $L = 10$ labels per predictor. (b) $L = 100$ labels per predictor.

FIGURE 4.6: Empirical advantage of adaptive policies with mixed predictors $p_j \sim \text{Uniform}(\bar{p} - 0.2, \bar{p} + 0.2)$.

In this regard, we present the result with homogeneous predictors in Figure 4.5, and those with mixed predictors in Figure 4.6. For both cases, we run experiments both in the high uncertainty regime, when the predictors' accuracy is tested on $|G| = 10$ trials, and the low uncertainty regime, when we have $|G| = 100$ trials instead. As expected, the more trials we have, the more the estimates $\hat{p}_j$ approach their ground-truth value

p_j, and thus the $\hat{c}_{ada}/\hat{c}_{nada}$ converges to its known accuracy counterpart. This makes Figures 4.5(b) and 4.6(b) very close to Figures 3.10(a) and 3.10(b).

On the contrary, in the high uncertainty regime we can see some small differences. In Figure 4.5(a) the difference between adaptive and non-adaptive policies is less evident for small $\bar{p}$, and thus the $\hat{c}_{ada}/\hat{c}_{nada}$ drops to lower values than Figure 4.5(b). This is because the predictors are almost random in that region, and thus the estimates $\hat{p}_j$ do not necessarily match the value of $\bar{p}$, leading to a less efficient adaptive allocation. A similar effect appears in Figure 4.6(a), even though the mixed population of predictors mitigates the phenomenon.

4.5 The estimate-predict tradeoff

Before closing the present discussion, we want to show how the results in this chapter, though theoretical, can have a concrete practical application. Recall that according to our assumptions in Section 4.1, each predictor outputs two distinct set of labels O_j and X_j. The first is used to estimate the accuracies $\hat{\boldsymbol{p}}$, the second to predict the classes of the items in M. However, until now we have taken for granted that the cardinality of O_j and X_j are given. What if we actually had the freedom to make this choice by ourselves?

In fact, it is not unimaginable to encounter a scenario where each predictor j is able to produce only $|O_j| + |X_j|$ labels *in total*. For example, in crowdsourcing the predictors are human workers that get hired to execute a fixed number of labelling tasks. In order not to waste the resources available to us, it is our job to decide how to optimally split the $|O_j| + |X_j|$ labels between the two sets.

Fortunately, this is an easy question to answer given the results presented in this chapter. Specifically, we define the following optimisation problem. Assume that each predictor provides the same number $S = |O_j| + |X_j|$ of labels in total (extending our results to a different number of labels per predictor is trivial). Then, for each collection policy presented in Sections 4.2 and 4.3 we can optimise over the respective upper bounds. This procedure splits S into a number of trials $|G| = S - R^*$ and a number of effective labels R^* that gives us the best guarantees over the classification error. Since the bounds presented in Sections 4.2 and 4.3 are close to the actual predictive error, such guarantees translate to an optimal performance.

As we would expect, the optimal split depends on all the parameters at play: the collection policy π we use, the total number of labels S, and the underlying accuracy distribution f_p of the predictors. At the same time, one interesting phenomenon emerges which is consistent across different choices of the aforementioned three parameters. We highlight it in the following loose statement:

Remark 4.12. For any collection policy π, and any population of predictors whose accuracy distribution f_p has both mean different from $\frac{1}{2}$ and small variance, the best allocation of resources is having zero trials, that is $R^* = S$, as long as S is not too large.

While Remark 4.12 can appear infuriatingly vague when compared with the other results in this chapter, it justifies a common practice in the crowdsourcing field. That is, oftentimes crowdsourcing pratictioners avoid employing any form of sophisticated statistical tool and revert to the basic majority voting rule (Snow et al., 2008; Barowy et al., 2012). In general, this works as long as the population of predictors is biased towards the ground-truth class y_i. Here, we can show that such a design choice is actually *optimal* if the conditions in Remark 4.12 apply.

In order to visualise why this is so, let us assume that $f_p \sim \text{Beta}(\alpha, \beta)$ and run the optimisation process for different choices of α and β. Thanks to the properties of the Beta distribution, we can force the probability density function to assume a large variety of shapes, as we show in Figure 4.7. Note that we choose a mean of $\alpha/(\alpha+\beta) = 0.6$ for all of them, but we progressively reduce the variance by increasing both α and β by the same factor. As a sidenote, if f_p has a mean of $\frac{1}{2}$, we are forced to spend some of the labels S on estimating the predictors' accuracy. In fact, running majority voting in such a scenario is impossible, as all predictors would receive a null weight $\hat{w}_j = 0$.

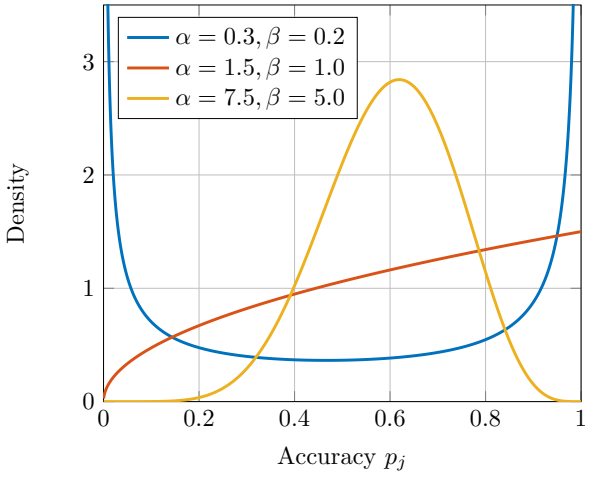


FIGURE 4.7: Three examples of the Beta distribution for different values of the parameters α and β.

We report the number of trials $|G|$ required for optimal performance in Table 4.1. The table explores all three parameter dimensions: the collection policy π, the total number of labels S, and the three accuracy distributions f_p portrayed in Figure 4.7. Three things must be noted. First, the choice of policy has little impact on the optimal split, with WB requiring slightly more trials and US/IG requiring slightly fewer. Second, the number of trials grows almost linearly with S, taking around 25% of the total number of labels. Third, as soon as the variance of f_p becomes small, it is not worth running any trial at all for small S. Instead, it is better to dedicate all resources to predict the

item classes, i.e. $R^* = S$ as highlighted in Remark 4.12.

	$\alpha = 0.3, \beta = 0.2$			$\alpha = 1.5, \beta = 1.0$			$\alpha = 7.5, \beta = 5.0$		
S	UNI	WB	US/IG	UNI	WB	US/IG	UNI	WB	US/IG
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
2	1	1	**0**	1	1	**0**	**0**	**0**	**0**
3	1	1	1	1	1	1	**0**	**0**	**0**
4	1	1	1	1	1	1	**0**	**0**	**0**
5	2	2	1	2	2	1	**0**	**0**	**0**
6	2	2	2	2	2	2	**0**	**0**	**0**
7	2	2	2	2	2	2	**0**	**0**	**0**
8	2	2	2	3	3	2	**0**	1	**0**
9	2	3	2	3	3	3	1	1	**0**
10	3	3	2	3	3	3	1	1	1
11	3	3	3	3	3	3	1	1	1
12	3	3	3	4	4	3	2	2	1
13	3	4	3	4	4	4	2	2	2
14	3	4	3	4	4	4	2	2	2
15	3	4	3	4	4	4	3	3	2
16	4	4	3	4	5	4	3	3	3
17	4	4	4	5	5	4	3	3	3
18	4	5	4	5	5	5	3	4	3
19	4	5	4	5	5	5	4	4	3
20	4	5	4	5	6	5	4	4	4

TABLE 4.1: Number of trials $|G|$ needed to get optimal guarantees given the predictors' productivity S.

Unfortunately, it is impossible to quantify the conditions under which Remark 4.12 is valid in general. Doing so requires knowing all of the three parameters π, S, f_p at play. For this reason, in a practical scenario any design choice has to be evaluated on a case by case basis.

4.6 Summary

In this chapter we extend the known accuracy model of Chapter 3 to account for predictors with unknown accuracy. We show that the majority of our results are reminiscent of the corresponding ones for the known accuracy case. If we had to summarise them in one sentence, we could say that, as long as the accuracy estimates $\hat{\boldsymbol{p}}$ are unbiased, all the results derived with known accuracy still apply. Furthermore, some of the results presented here are of independent interest: specifically, the theoretical improvement on the state of the art in Section 4.2.1 and the design choice guidelines in Section 4.5. Both can help a pratictioner make the correct design choices when deploying a multiple classifier system in the real world.

Against this background, the current chapter is as a stepping stone towards the next one. There, we remove any side-information from the model, whether it is the ground-truth accuracies $\boldsymbol{p}$ or the trial outcomes O, and deal with predictors of completely unknown properties. Importantly, such generalisation step does not make the results we present here invalid. In fact, we show there that as long as we can produce some estimates $\hat{\boldsymbol{p}}$ of the predictors' accuracy, the theoretical tools of this chapter remain applicable.

Chapter 5

Inferring the predictors' accuracy

Thanks to the solid theoretical foundations we lay in Chapters 3 and 4, we are now in the position to make our last step towards generality. Recall that in Chapter 3 we assume perfect knowledge of the predictors' accuracies $\boldsymbol{p}$. Conversely, in Chapter 4 we assume that the accuracies $\boldsymbol{p}$ are unknown, but we have access to their estimates $\hat{\boldsymbol{p}}$. Here, we let this last piece of information go, and rely exclusively on the dataset X to form our predicitons on the item classes.

This last scenario adheres closely to the original Dawid-Skene model (Dawid and Skene, 1979). There, the goal is to infer both the ability of a group of physicians and the correct diagnosis of their patients, given the opinion of each doctor on the patients' symptoms. With the advent of crowdsourcing, this model has found a new application: if we replace the physicians with crowdworkers, and the patients with a set of data-processing tasks, we are left with the same inference problem.

However, just solving this inference problem is not enough. As we show in the present chapter, most existing approaches to do so are computationally intensive. Clearly, this is not an issue in an offline setting, where the whole dataset is available for processing, and there is no time constraint. At the same time, some applications like crowdsourcing call for an online approach, where we need to make real-time inference decisions. Crucially, this is the case if we want to deploy an adaptive data collection strategy like the US policy, and enjoy its superior accuracy.

On a different note, the solution to the inference problem we study in this chapter is not unique. This is in stark constrast with the other variants of the Dawid-Skene model we study in Chapters 3 and 4, which are optimally solved by weighted majority voting and the plug-in aggregator respectively. A consequence of the presence of multiple inference solutions is that deriving general theoretical results is more challenging. As an example, the existing literature contains only two attempts of this kind (Karger et al., 2014; Gao et al., 2016). The rest take the form of algorithm-specific guarantees (see Section 2.2).

In this chapter we address both the aforementioned algorithmic and theoretical challenges. We begin in Section 5.1 by providing the details of the inference model we use throughout the whole chapter. Then, in Section 5.2 we introduce two new inference algorithms. The first is an efficient particle filter implementation of Monte Carlo sampling, and helps us define a baseline approach for probabilistic inference. The second is a novel approach based on approximate Bayesian inference, and provides good predictive accuracy while running several orders of magnitude faster than the state of the art.

Next, in Section 5.3 we focus on the theoretical challenge. Our contributions include a novel general bound on the accuracy of any inference algorithm, and several algorithm-specific bounds. All of these match the empirical performance of the respective algorithms very closely. Furthermore, these bounds cover both the UNI and US policies. The latter are the first result of this kind in the existing literature.

Finally, in Section 5.4 we run an empirical comparison of our algorithmic contributions with the state of the art. We do so both on synthetic and real datasets. Our results show that our algorithms are the most accurate among the existing Bayesian approaches. Moreover, they are the only ones exhibiting a consistent behaviour under the US policy, when the number of labels provided by each predictor is small. Lastly, we give a brief summary of all our contributions in Section 5.5.

5.1 A fully agnostic model

In this section we introduce our last model of multiple item classification. This model is the next natural step towards generality with respect o the other models we use in Chapters 3 and 4. Under this light, we keep many of the assumptions we use there (see Sections 3.1 and 4.1 for comparison), but remove any direct or side information about the accuracy of the individual predictors. As a consequence, our inference effort can only rely on the predictors' output over the original set of items.

Let us specify our setting in a more formal manner. As per the one-coin Dawid-Skene model we introduce in Sections 3.1 and 4.1, we have a set of M items with unknown ground-truth classes $y_i \in \{\pm 1\}$ and corresponding prior distribution $\mathbb{P}(y_i = +1) \equiv q$. Likewise, we have a pool of N predictors which become available multiple times according to a random sequence $\boldsymbol{a}$. At each timestep t, we assign the available predictor $j = a(t)$ to an item $i = \hat{\pi}(t)$ according to the policy $\hat{\pi}$, and receive the predictor's output label x_{ij}. After T timesteps the label collection process ends, and we use the resulting dataset $X = \{x_{ij}\}$ to form our final predictions $\hat{\boldsymbol{y}}$ over the item class vector $\boldsymbol{y}$.

At the same time, we have little information on the ground-truth accuracy $\mathbb{P}(x_{ij} = y_i) \equiv p_j$ of each predictor $j \in N$. Here, the only assumption we make is that the predictors are independent, and their accuracies $\boldsymbol{p}$ are extracted from an underlying

known distribution f_p. Together, this set of assumptions defines a generative model for the dataset X, which we summarise in graphical format in Figure 5.1.

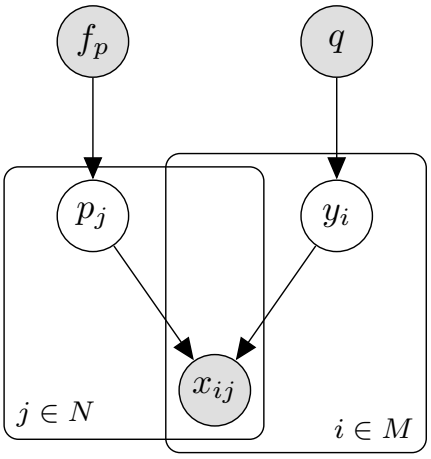


FIGURE 5.1: A graphical representation of the agnostic Dawid-Skene model.

With the background of Chapters 3 and 4.1, it is clear that the model in Figure 5.1 renews our research challenges. In fact, lacking any additional knowledge on the predictors' accuracies $\boldsymbol{p}$, the corresponding inference problem might even seem underdetermined at first glance. On the one hand, if we knew $\boldsymbol{p}$ like in Chapter 3, we would have all the tools we need to predict $\boldsymbol{y}$ efficiently. On the other hand, if we knew $\boldsymbol{y}$ beforehand, as we know the ground-truth classes $\boldsymbol{g}$ of the extra set of items G in Chapter 4.1, then we could estimate $\boldsymbol{p}$ easily. Here, we are tasked with doing both things at the same time, using the information in the dataset X alone.

Before moving to our contributions, let us point out that the challenges the present model poses are not only algorithmic. In fact, once we choose an existing algorithm or develop a new one, we are often left with no guarantees on how well such an algorithm will perform. Thus, we also need to address the corresponding theoretical challenge. To this end, after proposing our own algorithmic solutions in Section 5.2, we study their theoretical properties in Section 5.3.

5.2 Algorithmic contributions

The agnostic Dawid-Skene model we present in Section 5.1 has already received considerable attention in the machine learning literature. However, the existing algorithms (see Section 2.2) fail to address all the three requirements that underlie our present research effort. Namely, the need for computational speed, theoretical guarantees and good predictive performance in all settings.

Thus, we dedicate this section to our own take on this algorithmic challenge. Our contribution is divided in two parts. In the first (Section 5.2.1), we present our approach

to Bayesian probabilistic inference via Monte Carlo sampling. There, we show that a particle filter implementation is required for an efficient estimation of the posterior in an online setting. Furthermore, since the posterior distribution is strongly bimodal, we introduce a mirroring technique that allows the particles to jump out of the wrong mode with little computational cost. This first part also functions as an introduction to Bayesian inference on the Dawid-Skene model, and as such we often refer to it in the remainder of this chapter.

In the second part (Section 5.2.2), we present a novel algorithm, which we name Streaming Bayesian Inference for Crowdsourcing (SBIC). This algorithm employs a variational approximation of the posterior distribution, and updates it on the fly as new data becomes available. As a result, SBIC is not only computationally light, but it offers good theoretical guarantees as we detail later in Section 5.3.4.

5.2.1 Posterior estimation via Monte Carlo sampling

Bayesian probabilistic inference aims at estimating the posterior distribution on the latent variables of our generative model, and using this information to form a prediction on their ground-truth value. However, doing so on the Dawid-Skene model we present in Section 5.1 is not straightforward. A classic but computationally-expensive solution to this is Monte Calro sampling. Here, we show how to implement it efficiently.

First, let us derive an analytical expression for the posterior of the model in Figure 5.1. The probability of the evidence X given the latent variables $\boldsymbol{y}$ and $\boldsymbol{p}$ is the following:

$$\mathbb{P}(X|\boldsymbol{y},\boldsymbol{p}) = \prod_{x_{ij}\in X} p_j^{\mathbb{I}(x_{ij}=y_i)}(1-p_j)^{\mathbb{I}(x_{ij}\neq y_i)} \tag{5.1}$$

But, of course, we are interested in the opposite conditional probability. That is, we want the posterior on the latent variables $\boldsymbol{y}$ and $\boldsymbol{p}$ given the evidence X. For that, we need Bayes' formula and the prior distributions q and f_p:

$$\mathbb{P}(\boldsymbol{y},\boldsymbol{p}|X,q,f_p) \propto \mathbb{P}(X|\boldsymbol{y},\boldsymbol{p})\mathbb{P}(\boldsymbol{y}|q)\mathbb{P}(\boldsymbol{p}|f_p) \tag{5.2}$$

where the prior probability of the evidence $\mathbb{P}(X)$ in the denominator is hidden by the proportionality operator, and we use our independence assumption to split the joint probability $\mathbb{P}(\boldsymbol{y},\boldsymbol{p}|q,f_p)$ in its corresponding factors $\mathbb{P}(\boldsymbol{y}|q)$ and $\mathbb{P}(\boldsymbol{p}|f_p)$.

Now, what we really care about is the marginal probability on the class of each item i. If we could compute it, we would form our final predictions as follows:

$$\hat{y}_i = \operatorname*{argmax}_{\ell\in\{\pm 1\}} \Big\{\mathbb{P}(y_i=\ell|X,q,f_p)\Big\} \tag{5.3}$$

which is also known as the maximum-a-posteriori (MAP) estimator.

As will become apparent in a few lines, computing the argument of the argmax operator in Equation 5.3 is unfeasible. Still, we can make a big step towards that goal by taking Equation 5.2 and *marginalising away* the predictors' accuracies $\boldsymbol{p}$:

$$\begin{aligned}\mathbb{P}(\boldsymbol{y}|X,q,f_p) &= \int_{\boldsymbol{p}} \mathbb{P}(\boldsymbol{y},\boldsymbol{p}|X,q,f_p)d\boldsymbol{p} \\ &\propto \mathbb{P}(\boldsymbol{y}|q)\int_{\boldsymbol{p}} \mathbb{P}(X|\boldsymbol{y},\boldsymbol{p})\mathbb{P}(\boldsymbol{p}|f_p)d\boldsymbol{p}\end{aligned} \tag{5.4}$$

and then we can substitute the closed-form expression for $\mathbb{P}(X|\boldsymbol{y},\boldsymbol{p})$ from Equation 5.1, and recall that the predictors are independent, which yields the following:

$$\mathbb{P}(\boldsymbol{y}|X,q,f_p) \propto \mathbb{P}(\boldsymbol{y}|q)\prod_{j\in N}\int_0^1 p_j^{d(X_j,\boldsymbol{y})}(1-p_j)^{e(X_j,\boldsymbol{y})}\mathbb{P}(p_j|f_p)dp_j \tag{5.5}$$

where $d(X_j,\boldsymbol{y}) = \sum_{x_{ij}\in X_j}\mathbb{I}(x_{ij}=y_i)$ is the number of labels from predictor j that match the candidate class vector $\boldsymbol{y}$, and $e(X_j,\boldsymbol{y}) = \sum_{x_{ij}\in X_j}\mathbb{I}(x_{ij}\neq y_i)$ is the number of labels that contradict it.

For most prior distributions f_p, we can only compute a numerical approximation of the integral in Equation 5.5. A notable exception is the Beta distribution which, as the conjugate prior of our Bernoulli predictors, yields a closed form solution. In the our discussion we make this extra assumption, and state that $f_p = \text{Beta}(\alpha,\beta)$ with known parameters, or *pseudo-counts*, α and β. As a consequence we can rewrite Equation 5.5 as follows:

$$\mathbb{P}(\boldsymbol{y}|X,q,\alpha,\beta) \propto \mathbb{P}(\boldsymbol{y}|q)\prod_{j\in N} B\big(d(X_j,\boldsymbol{y})+\alpha, e(X_j,\boldsymbol{y})+\beta\big) \tag{5.6}$$

where $B(\bullet,\bullet)$ is the Beta function[1], and we let the proportionality operator absorb the denominator $B(\alpha,\beta)$.

Even in these favourable conditions, we cannot directly compute the marginal $\mathbb{P}(y_i|X,q,f_p)$ we need for the MAP estimate in Equation 5.3. This is because we would have to sum over an exponential number of possible candidate class vectors $\boldsymbol{y}$ that contain a specific y_i as follows:

$$\mathbb{P}(y_i=\ell|X,q,f_p) = \sum_{\boldsymbol{y}:y_i=\ell}\mathbb{P}(\boldsymbol{y}|X,q,f_p) \tag{5.7}$$

which rapidly becomes unfeasible as the set M grows larger than a dozen items.

Thus, we can resort to the Monte Carlo sampling framework, and approximate Equation 5.7 on a small sample of all possible candidates $\boldsymbol{y}$. In order to build an accurate and efficient approximation, this sample of class vectors, which we denote with the multiset

[1]The Beta function has the following convenient definition: $B(u,v) = \int_0^1 x^{u-1}(1-x)^{v-1}dx$

$Y = \{\boldsymbol{y}_k\}$, has to satisfy the following two properties. First, we need to sample the elements of Y according to the posterior $\mathbb{P}(\boldsymbol{y}|X, q, f_p)$. With this property, we can build the following unbiased estimator of the marginal:

$$\mathbb{P}(y_i = \ell|X, q, f_p) \approx \frac{1}{|Y|}\sum_{k=1}^{|Y|}\mathbb{I}(y_{ik} = \ell) \tag{5.8}$$

Second, we need to define an update mechanism for Y as new data becomes available. This property is crucial in the online setting, for instance when used in conjunction with the US policy, as we need to re-estimate the posterior after acquiring each data point in X. Of course, the simplest approach is to extract Y anew for every X^t, but such a method is very demanding computationally. In the next Sections 5.2.1.1 and 5.2.1.2, we show how an appropriately-built particle filter can satisfy both properties.

5.2.1.1 Sequential Monte Carlo with a bimodal posterior

The problem of tracking the evolution of a probability distribution over time is often addressed by means of a particle filter (Cappé et al., 2007). The core idea is to have a set of particles, in our case the elements of Y, and let them randomly evolve over time by mutating some of their elements. If done correctly, the particles will be able to track the main modes of the probability distribution.

Here, we do not have a truly evolving probability distribution. Instead, the underlying generative probabilistic model is fixed (see Figure 5.1), but our knowledge of it improves as the dataset X^t contains more and more data points. In this sense, we can treat the posterior distribution over the latent variables $\boldsymbol{y}$ and $\boldsymbol{p}$ as our probability distribution that evolves in t.

This framework is called Sequential Monte Carlo (Chopin, 2002), and was first proposed to speed up Monte Carlo probabilistic inference on large datasets. A high-level overview of the method is the following:

1. Initialise the multiset of particles Y according to the prior;
2. Augment the current dataset X^t with the new observations;
3. Compute an importance weight w_k for each particle $\boldsymbol{y}_k \in Y$ given the new evidence;
4. Sample a new set of particles from Y with replacement according to the weights;
5. Add a local perturbation to the new particles to reduce the number of duplicates;
6. Go to 2 unless all the data has been processed already.

where the importance weighting in Step 3, and the resampling in Step 4 allow the set of particles to drift along with the posterior distribution of $\boldsymbol{y}$ given X^t. We give the details of our implementation in Section 5.2.1.2.

Still, a vanilla implementation of Sequential Monte Carlo on the Dawid-Skene model of Section 5.1 might fail spectacularly depending on the initial random choice of particles Y. As an example, in our experiments of Figure 5.2 we get state-of-the-art performance for most random initialisations, but around 99% error rate in a small minority of cases. Depending on the specific parameter settings of the experiments, this minority of failures is as large as 1.2% of the total runs. Thus, before dealing with the details of our algorithmic solution, let us explain what is the underlying issue.

The posterior distribution in Equation 5.5 is strongly bimodal. One way of showing it analytically, is to take the version with Beta priors in Equation 5.6 and choose $\alpha = \beta$. Let us also define $\bar{\boldsymbol{y}}$ as the opposite of a class vector $\boldsymbol{y}$, such that $\bar{y}_i = -y_i$ for all items $i \in M$. Then, we have the property that:

$$\mathbb{P}(\bar{\boldsymbol{y}}|X, q, \alpha, \alpha) = \mathbb{P}(\boldsymbol{y}|X, q, \alpha, \alpha) \tag{5.9}$$

since for any X_j we have $d(X_j, \bar{\boldsymbol{y}}) = e(X_j, \boldsymbol{y})$, and the Beta function is symmetric, i.e. $B(u, v) = B(v, u)$ for any $u, v > 0$.

In this extreme case, even with a very large dataset X, the posterior has two identical modes, one centred on the ground-truth class vector $\boldsymbol{y}$, and the other on the *opposite* vector $\bar{\boldsymbol{y}}$. Thus, an optimal inference algorithm will either converge to $\boldsymbol{y}$ or $\bar{\boldsymbol{y}}$ with 50% chance. More in general, when $\alpha \neq \beta$ or the prior is not Beta-distributed, we expect a stronger mode around the ground-truth class vector $\boldsymbol{y}$. However, there will always be a large secondary mode around $\bar{\boldsymbol{y}}$ that threatens to fool our inference algorithm into a completely wrong solution with 100% error rate.

Our contribution to the classic Sequential Monte Carlo framework is the addition of a *mirroring* stage after perturbing the particles (Step 5 above). The idea is to allow a particle $\boldsymbol{y}_k$ to flip to its opposite $\bar{\boldsymbol{y}}$ according to the posterior probability of the two alternatives. In this way, the set of particle cannot get stuck in the wrong mode of the posterior, no matter the random initialisation we use or the order in which we visit the dataset X.

Before delving into the details of this technique in the next Section 5.2.1.2, let us present a visualisation of its advantages in Figure 5.2. There we compare the empirical performance of a particle filter with and without the additional mirroring stage. The setting is similar to our experiments in Chapters 3 and 4. Namely, we fix the number of items to $|M| = 1000$, extract the predictors' accuracy from a Beta$(4, 3)$ distribution and collect their labels according to the US policy. In order to compute stable estimates, we repeat the experiments multiple times until we observe at least 250 classification errors

for each value of $R \in [1, 40]$. Note how the classification accuracy of the regular particle filter becomes more and more erratic as R increases. At the same time, for some lucky initialisation choices, the performance of the two algorithm matches. By introducing the mirroring stage, we can make sure that the predictive performance does not depend on the random initialisation of the particles.

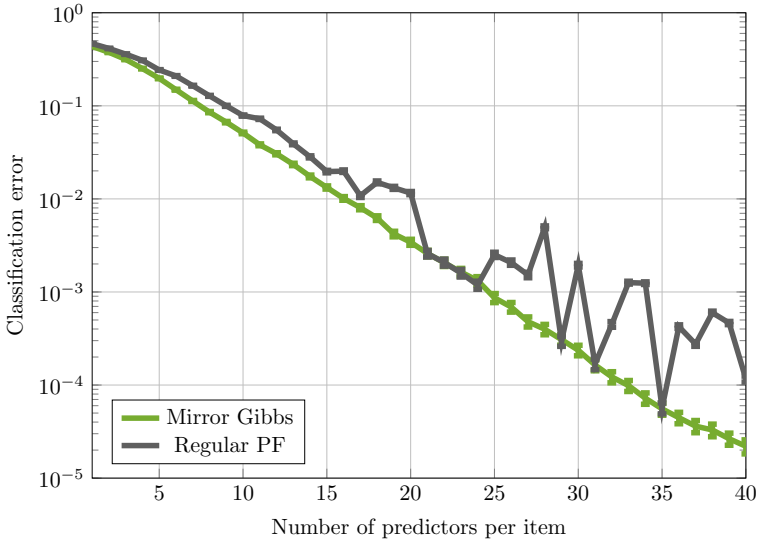


FIGURE 5.2: Comparison between the empirical classification error of Mirror Gibbs and a regular particle filter (PF) under the US policy.

5.2.1.2 Mirror Gibbs particle filter

In this section we go through the details of our implementation of Sequential Monte Carlo with an additional mirroring stage. Throughout our presentation we refer to the high-level stages of Sequential Monte Carlo listed in Section 5.2.1.1. For clarity, we also split the pseudocode accordingly, with a main routine in Algorithm 5.1, and corresponding subroutines in Algorithms 5.2 to 5.6.

First, let us discuss the initialisation procedure (Step 1). With no data observed yet, i.e. $X^0 = \emptyset$, we can extract the initial set of particles according to the prior q on the item classes only. We do so in Algorithm 5.2, where n_{part} is the required number of particles, and each particle $\boldsymbol{y}_k$ is represented as a column vector (thus y_{ik} is the ith entry of particle $\boldsymbol{y}_k$).

Second, we augment the dataset X^t one label at a time (Step 2) as shown in the loop of Algorithm 5.1 (Lines 4-14). However, recall that the particles of the current set Y are extracted according to the old posterior $\mathbb{P}(\boldsymbol{y}|X^{t-1}, q, f_p)$. Thus, after augmenting the old dataset X^{t-1} with the new label x_{ij}, we need to re-weight the importance of each particle $\boldsymbol{y}_k$ in the Monte Carlo estimate of Equation 5.8 (Step 3). We do so by

computing the following importance weight w_k for each particle $\boldsymbol{y}_k \in Y$:

$$\begin{aligned} w_k &= \frac{\mathbb{P}(\boldsymbol{y}_k|X^t, q, f_p)}{\mathbb{P}(\boldsymbol{y}_k|X^{t-1}, q, f_p)} \\ &= \frac{\int_0^1 p_j^{d(X_j^t, \boldsymbol{y}_k)}(1-p_j)^{e(X_j^t, \boldsymbol{y}_k)}\mathbb{P}(p_j|f_p)dp_j}{\int_0^1 p_j^{d(X_j^{t-1}, \boldsymbol{y}_k)}(1-p_j)^{e(X_j^{t-1}, \boldsymbol{y}_k)}\mathbb{P}(p_j|f_p)dp_j} \end{aligned} \tag{5.10}$$

where the expression for the posterior comes from Equation 5.5, and depends only on the output of the predictor $j = a(t)$ available at the current time step t. If we assume a Beta prior $f_p = \text{Beta}(\alpha, \beta)$ on the predictors like in Equation 5.6, we obtain the following simpler expression:

$$w_k = \frac{B\big(d(X_j^t, \boldsymbol{y}_k) + \alpha, e(X_j^t, \boldsymbol{y}_k) + \beta\big)}{B\big(d(X_j^{t-1}, \boldsymbol{y}_k) + \alpha, e(X_j^{t-1}, \boldsymbol{y}_k) + \beta\big)} \tag{5.11}$$

which is the one that appears in Algorithm 5.3.

Third, we resample the set of particles Y (Step 4). This stage ensures that particles with low importance weights are weeded out of our sample set. Conversely, particles closer to the main modes of the posterior distribution have a higher chance of survival. Since we sample with replacement, we often end up with multiple copies of particles in this second category. Algorithm 5.4 outlines the pseudocode for this resampling stage.

Fourth, the Sequential Monte Carlo recipe requires adding some local noise to the new set of particles (Step 5). The noise has both the effect of eliminating the duplicates, and helping the set of particles track the evolution of the posterior distribution. We achieve this second goal via Gibbs' sampling: given a particle $\boldsymbol{y}_k$ and an item i, we randomly flip the entry y_{ik} according to the posterior of the two alternatives. If we denote the modified particle with $\boldsymbol{y}'_k$, the corresponding posterior probability can be computed as follows:

$$\begin{aligned} \mathbb{P}(\boldsymbol{y}'_k|X, q, f_p) =& \mathbb{P}(\boldsymbol{y}_k|X, q, f_p)\frac{\mathbb{P}(y'_{ik}|q)}{\mathbb{P}(y_{ik}|q)} \\ & \prod_{j \in N_i^t} \frac{\int_0^1 p_j^{d(X_j^t, \boldsymbol{y}'_k)}(1-p_j)^{e(X_j^t, \boldsymbol{y}'_k)}\mathbb{P}(p_j|f_p)dp_j}{\int_0^1 p_j^{d(X_j^t, \boldsymbol{y}_k)}(1-p_j)^{e(X_j^t, \boldsymbol{y}_k)}\mathbb{P}(p_j|f_p)dp_j} \end{aligned} \tag{5.12}$$

where the correction factor from the posterior of $\boldsymbol{y}_k$ depends on the prior on the modified item i, and only the output of the predictors who labelled i. We use a version of Equation 5.12 with Beta prior in Algorithm 5.5 Lines 4-7.

Standard Gibbs' sampling (Murphy, 2012) requires repeating this procedure for every entry $i \in M$. In our experiments we found that choosing a random subset of entries $I \subset M$ is not only more efficient computationally, but also yields more accurate predictions. The latter is likely due to the random ordering, which helps the underlying Markov

Algorithm 5.1 MirrorGibbs

Input: dataset X, availability a, policy π, # particles n_{part}, # flips n_{flip}, priors q, α, β
Output: predictions $\hat{\boldsymbol{y}}$

$X^0 \leftarrow \emptyset$
$Y \leftarrow \text{InitialiseParticles}(q, n_{part})$
$\mu_i \leftarrow \frac{1}{|Y|}\sum_{k=1}^{|Y|} \mathbb{I}(y_{ki} = +1)$ for all $i \in M$
for $t = 1$ **to** T **do**
    $i \leftarrow \pi(t)$
    $j \leftarrow a(t)$
    $X^t \leftarrow X^{t-1} \cup x_{ij}$
    for $k = 1$ **to** $|Y|$ **do**
        $w_k \leftarrow \text{ImportanceWeight}(X^{t-1}, x_{ij}, \boldsymbol{y}_k, \alpha, \beta)$
    $Y \leftarrow \text{ResampleParticles}(Y, \{w_k\})$
    $Y \leftarrow \text{RejuvenateParticles}(Y, X^t, n_{flip}, q, \alpha, \beta)$
    extract an integer $k \in \left[1, |Y|\right]$ uniformly
    $\boldsymbol{y}_k \leftarrow \text{MirrorParticle}(\boldsymbol{y}_k, X^t, q, \alpha, \beta)$
    $\mu_i \leftarrow \frac{1}{|Y|}\sum_{k=1}^{|Y|} \mathbb{I}(y_{ki} = +1)$ for all $i \in M$
return $\hat{y}_i = 2\mathbb{I}(\mu_i > \frac{1}{2}) - 1$, $\forall i$

Algorithm 5.2 InitialiseParticles

Input: prior q, number of particles n_{part}
Output: set of particles Y

$Y \leftarrow \emptyset$
for $k = 1$ **to** n_{part} **do**
    for all $i \in M$ **do**
        extract $y_{ik} \in \{\pm 1\}$ such that $\mathbb{P}(y_{ik} = +1) = q$
    $Y \leftarrow Y \cup \boldsymbol{y}_k$
return Y

Algorithm 5.3 ImportanceWeight

Input: old dataset X, new label x_{ij}, particle $\boldsymbol{y}_k$, priors α and β
Output: importance weight w_k

$d_j \leftarrow \sum_{x_{hj} \in X_j} \mathbb{I}(x_{hj} = y_{hk})$
$e_j \leftarrow \sum_{x_{hj} \in X_j} \mathbb{I}(x_{hj} \neq y_{hk})$
$d'_j \leftarrow d_j + \mathbb{I}(x_{ij} = y_{ik})$
$e'_j \leftarrow e_j + \mathbb{I}(x_{ij} \neq y_{ik})$
return $w_k = \frac{B(d'_j+\alpha, e'_j+\beta)}{B(d_j+\alpha, e_j+\beta)}$

Algorithm 5.4 ResampleParticles

Input: old set of particles Y, corresponding weights $\{w_k\}$
Output: new set of particles Y'

$Y' \leftarrow \emptyset$
while $|Y'| < |Y|$ **do**
    extract an integer $k \in \big[1, |Y|\big]$ according to $\{w_k\}$
    $Y' \leftarrow Y' \cup \{\boldsymbol{y}_k\}$
return Y'

Algorithm 5.5 RejuvenateParticles

Input: set of particles Y, dataset X, number of flips n_{flip}, priors q, α and β
Output: rejuvenated set of particles Y

for $k = 1$ **to** $|Y|$ **do**
    extract uniformly $I \subseteq M$ such that $|I| = n_{flip}$
    for all $i \in I$ **do**
        $\boldsymbol{y}'_k \leftarrow (y_{0k}, \ldots, -y_{ik}, \ldots, y_{|M|k})$
        $w'_k \leftarrow \mathbb{P}(y'_{ik}|q) \prod_{j \in N_i} B\big(d(X_j, \boldsymbol{y}'_k) + \alpha, e(X_j, \boldsymbol{y}'_k) + \beta\big)$
        $w_k \leftarrow \mathbb{P}(y_{ik}|q) \prod_{j \in N_i} B\big(d(X_j, \boldsymbol{y}_k) + \alpha, e(X_j, \boldsymbol{y}_k) + \beta\big)$
        replace $\boldsymbol{y}_k \leftarrow \boldsymbol{y}'_k$ with probability $\frac{w'_k}{w'_k + w_k}$
return Y

Algorithm 5.6 MirrorParticle

Input: particle $\boldsymbol{y}$, dataset X, priors q, α and β
Output: mirrored particle $\boldsymbol{y}$

$\bar{w} \leftarrow \mathbb{P}(-y_{ik}|q) \prod_{j \in N} B\big(e(X_j, \boldsymbol{y}) + \alpha, d(X_j, \boldsymbol{y}) + \beta\big)$
$w \leftarrow \mathbb{P}(y_{ik}|q) \prod_{j \in N} B\big(d(X_j, \boldsymbol{y}) + \alpha, e(X_j, \boldsymbol{y}) + \beta\big)$
replace $\boldsymbol{y} \leftarrow (-y_0, -y_1, \ldots, -y_{|M|})$ with probability $\frac{\bar{w}}{\bar{w} + w}$
return $\boldsymbol{y}$

chain converge to the posterior distribution faster. In Algorithm 5.5 we regulate the cardinality of the subset I with the input variable n_{flip}.

Finally, we add a mirroring stage at the end of the iteration (see Algorithm 5.1 Lines 12-13). In this stage we select a single particle $\boldsymbol{y}_k$ from the set Y at random, and flip the values of all its entries y_{ik} at the same time. The mirrored particle $\bar{\boldsymbol{y}}_k$ is then accepted according to its posterior probability (see Algorithm 5.6). Note that comparing the posteriors of $\bar{\boldsymbol{y}}_k$ and $\boldsymbol{y}_k$ (Lines 1-2) is easier than the corresponding operation for a Gibbs' step in Equation 5.12. In fact, we do not need to count the number of correct and incorrect labels twice, since $d(X_j, \bar{\boldsymbol{y}}_k) = e(X_j, \boldsymbol{y}_k)$ and viceversa.

Our experiments show that adding this mirroring stage to a vanilla Sequential Monte Carlo implementation solves the issues discussed in Section 5.2.1.1 with negligible computational cost. Moreover, it enables us to compare any other inference algorithm with a direct estimation of the posterior. This is crucial for our experiments in Section 5.4, where we compare the predictive accuracy of both the state-of-the-art algorithms we introduce in Section 2.2 and the novel Streaming Bayesian Inference for Crowdsourcing algorithm in Section 5.2.2.

5.2.2 Streaming Bayesian Inference for Crowdsourcing

In Section 5.2.1 we showed that computing the marginal posterior distribution over the class y_i of each item $i \in M$ is challenging. There, we tackled the problem by Monte Carlo sampling, and proposed an efficient way to estimate it. Still, Monte Carlo techniques, no matter how efficient, require a large number of samples to reduce the variance in their estimates. Our mirror particle filter method (see Section 5.2.1.2) is not different. Thus, if we want to push the computational speed of our algorithmic solution further, we need a different approach.

Here, we resort to approximate Bayesian inference and introduce a new algorithm, which we call Streaming Bayesian Inference for Crowdsourcing (SBIC).[2] This algorithm is built around an analytical approximation of the posterior distribution, which allows for a certain degree of flexibility in its implementation. Consequently, we propose two variants of SBIC: the first, which we call Fast SBIC, prioritises computational speed, while the second, Sorted SBIC, prioritises predictive accuracy. Before presenting the details of these two variants in Sections 5.2.2.1 and 5.2.2.2, let us introduce the main ideas behind the SBIC algorithm.

According to mean-field variational principles, the posterior distribution in Equation 5.2 can be approximated as the product of some independent factors $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ (Liu et al.,

[2]While the SBIC algorithm was developed with crowdsourcing in mind, it applies to the general Dawid-Skene model this chapter is based on (see Section 5.1).

2012):

$$\mathbb{P}(\boldsymbol{y}, \boldsymbol{p}|X, q, f_p) \approx \prod_{i \in M} \mu_i(y_i) \prod_{j \in N} \nu_j(p_j) \tag{5.13}$$

where each factor μ_i corresponds to an item $i \in M$, and each factor ν_j to a predictor $j \in N$. The advantage of this factorisation of the posterior is that computing the marginal probability of y_i (see Equation 5.7) becomes trivial since:

$$\mathbb{P}(y_i = \ell|X, q, f_p) \approx \mu_i^t(\ell) \tag{5.14}$$

However, we still need to define a procedure to compute the value of the factors $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ in Equation 5.13. This is crucial since we want our approximation to be as close as possible to the true value of $\mathbb{P}(\boldsymbol{y}, \boldsymbol{p}|X, q, f_p)$. The classic approach requires the minimisation of the Kullback-Leibler divergence (KL) between the two sides of Equation 5.13, thus turning the inference problem into the following optimisation problem (Murphy, 2012):

$$\boldsymbol{\mu}, \boldsymbol{\nu} = \underset{\boldsymbol{\mu}', \boldsymbol{\nu}'}{\operatorname{argmin}} \left\{ KL\Big(\prod_{i \in M} \mu_i'(y_i) \prod_{j \in N} \nu_j'(p_j) \Big\| \mathbb{P}(\boldsymbol{y}, \boldsymbol{p}|X, q, f_p) \Big) \right\} \tag{5.15}$$

Previous work on the Dawid-Skene model (Liu et al., 2012) uses block coordinate descent to solve the optimisation problem in Equation 5.15. That is, from an initial guess on the item factors, we can iteratively refine the value of $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ until convergence. Under the additional assumption that $f_p = \text{Beta}(\alpha, \beta)$, we can do so by computing the optimal distribution of the predictors given the item factors $\boldsymbol{\mu}$:

$$\nu_j(p_j) \sim \text{Beta}\Big(\sum_{i \in M_j} \mu_i(x_{ij}) + \alpha, \sum_{i \in M_j} \mu_i(-x_{ij}) + \beta \Big) \tag{5.16}$$

which is still a Beta distribution thanks to the properties of the KL divergence. Similarly, given the predictor factors $\boldsymbol{\nu}$, we can optimise over the items as follows:

$$\begin{aligned} \mu_i(y_i) &\propto \prod_{j \in N_i} \exp\Big(\mathbb{E}\big(\log(p_j) \big) \Big)^{\mathbb{I}(x_{ij} = y_i)} \exp\Big(\mathbb{E}\big(\log(1 - p_j) \big) \Big)^{\mathbb{I}(x_{ij} \neq y_i)} \\ &\approx \prod_{j \in N_i} \mathbb{E}(p_j)^{\mathbb{I}(x_{ij} = y_i)} \big(1 - \mathbb{E}(p_j) \big)^{\mathbb{I}(x_{ij} \neq y_i)} \end{aligned} \tag{5.17}$$

where the last expression is the first-order Taylor approximation, which yields less extreme values in the interval $(0, 1)$. Note that, while Equations 5.16 and 5.17 are the solutions of locally-convex optimisation problems, the global objective from Equation 5.15 is not convex in general. Usually this is not a problem, and the corresponding approximate mean-field algorithm (AMF) reaches the global optimum reliably (Liu et al., 2012). However, in our experiments of Section 5.4 we show that it can exhibit a more unstable behaviour when used in an online setting.

Our work diverges from this iterative approach in that we use a streaming method to

optimise the factors $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. That is, instead of processing the whole dataset multiple times to compute the estimates in Equations 5.16 and 5.17, we make a single optimisation step for each data point in X. On the one hand, this yields quicker updates to $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, which allows us to run SBIC more efficiently. On the other hand, this yields a tractable random walk over the value of the factors $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, which allows us to derive strong theoretical guarantees on the predictive accuracy of SBIC (see Section 5.3.4).

More in detail, the core mechanism of the SBIC algorithm is the following. First, we assume again that the prior on each predictor's accuracy is $f_p = \text{Beta}(\alpha, \beta)$. Second, we initialise the item factors $\boldsymbol{\mu}^0$ to their respective prior q, that is:

$$\mu_i^0(+1) = q \qquad \text{and} \qquad \mu_i^0(+1) = 1 - q \tag{5.18}$$

Then, for each timestep t, we update the factor ν_j^t that corresponds to the currently available predictor $j = a(t)$. As per the approximate mean-field updates in Equation 5.16, the new factor is still Beta-distributed. Furthermore, we are only interested in its expectation $\bar{p}_j^t = \mathbb{E}\{\nu_j^t\}$, as this is the value that informs the first-order approximation in Equation 5.17. As a consequence, we have:

$$\bar{p}_j^t = \frac{\sum_{i \in M_j^{t-1}} \mu_i^{t-1}(x_{ij}) + \alpha}{|M_j^{t-1}| + \alpha + \beta} \tag{5.19}$$

where M_j^{t-1} is the set of items labelled by predictor j up to time t.

Next, we update the factor μ_i^t that corresponds to the item $i = \pi(t)$ labelled by the current predictor $j = a(t)$. Recall that Equation 5.17 is a product of N_i terms, one per each label x_{ij} that is cast on item i. Thus, we can easily update μ_i^t iteratively in the following way:

$$\mu_i^t(y_i) \propto \begin{cases} \mu_i^{t-1}(y_i)\bar{p}_j^t & \text{if } x_{ij} = y_i \\ \mu_i^{t-1}(y_i)(1 - \bar{p}_j^t) & \text{if } x_{ij} \neq y_i \end{cases} \tag{5.20}$$

At the end of the data collection process, we can inspect the final value of the factors $\boldsymbol{\mu}^T$, and use the MAP estimates as our predictions on the item classes as follows:

$$\hat{y}_i = \underset{\ell \in \{\pm 1\}}{\text{argmax}} \left\{ \mu_i^T(\ell) \right\} \tag{5.21}$$

We summarise the high-level structure of the SBIC algorithm in Figure 5.3. There, we unroll the graphical model representation of Figure 5.1, and show the dependency of each factor node on the corresponding data. Specifically, note how we update the vector of item factors $\boldsymbol{\mu}^t$ given the current predictor factor ν_j^t and data point x_{ij}^t. Similarly, in order to compute each new predictor factor ν_j^t, we only need the corresponding predictor's past output X_j^{t-1} and the previous item factors $\boldsymbol{\mu}^{t-1}$.

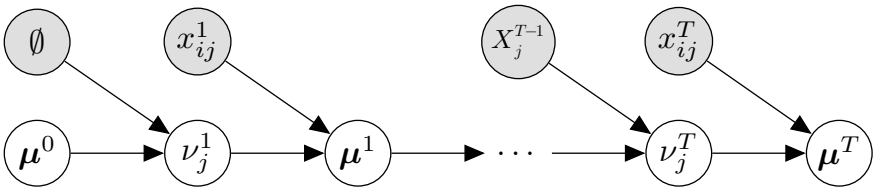


FIGURE 5.3: Unrolled graphical representation of the SBIC algorithm. The prior nodes f_p and q are omitted for simplicity.

All things considered, the SBIC algorithm falls under the umbrella of the Streaming Variational Bayes framework (Broderick et al., 2013). This is because, at each timestep t we trust our current approximations $\boldsymbol{\mu}^t$ and $\boldsymbol{\nu}^t$ to be close to the exact posterior $\mathbb{P}(\boldsymbol{y}, \boldsymbol{p}|X^t, q, f_p)$, and we use their values to inform the next local updates. However, even though the general structure of SBIC matches the framework proposed in (Broderick et al., 2013), the specific definition of our updates matters. In fact, Equations 5.19 and 5.20 allow us to derive strong theoretical guarantees for the SBIC algorithm in Section 5.3, whereas the only current results for the Streaming Variational Bayes framework in (Manoel et al., 2017) apply to a different setting.

Similarly, the SBIC algorithm can be interpreted as a form of constrained variational inference with implicit constraints. That is, instead of having an explicit alteration of the KL objective in Equation 5.15, the optimisation process is implicitly constrained by the local updates defined by Equations 5.19 and 5.20. Whether it is possible to express these implicit constraints in explicit form or not is an open question.

Finally, the sequential nature of the SBIC algorithm means that its output is deeply influenced by the order in which we process the labels in the dataset X. At the same time, the structure of the algorithm does not mandate for any specific ordering of the dataset X. Thus, by altering this ordering, we have the opportunity of optimising SBIC for different objectives. In this light, we present a variant of SBIC that prioritises computational speed in Section 5.2.2.1, and another variant that prioritises predictive accuracy in Section 5.2.2.2.

5.2.2.1 Fast SBIC

We can optimise SBIC for computational speed by keeping the natural ordering of the dataset X, represented by the predictor availability vector $\boldsymbol{a}$ and the policy π. The resulting algorithm, which we call Fast SBIC, makes use of the following computational tricks.

First, we express the value of each item factor μ_i^t in terms of its log-odds z_i^t. Accordingly,

Algorithm 5.7 FastSBIC

Input: dataset X, availability a, policy π, prior θ
Output: predictions $\hat{\boldsymbol{y}}$

$z_i^0 = \log\big(q/(1-q)\big), \quad \forall i \in M$
for $t = 1$ **to** T **do**
  $i \leftarrow \pi(t)$
  $j \leftarrow a(t)$
  $\bar{p}_j^t \leftarrow \frac{\sum_{i \in M_j^{t-1}} \mathrm{sig}(x_{ij} z_i^{t-1}) + \alpha}{|M_j^{t-1}| + \alpha + \beta}$
  $z_i^t \leftarrow z_i^{t-1} + x_{ij} \log\big(\bar{p}_j^t/(1-\bar{p}_j^t)\big)$
  $z_{i'}^t \leftarrow z_{i'}^{t-1}, \quad \forall i' \neq i$
return $\hat{y}_i = \mathrm{sign}(z_i^T), \quad \forall i$

Algorithm 5.8 SortedSBIC

Input: dataset X, availability a, policy π, prior θ
Output: final predictions $\hat{\boldsymbol{y}}^T$

$s_i^k = \log(q/(1-q)), \forall i \in M, \forall k \in M$
for $t = 1$ **to** T **do**
  $i \leftarrow \pi(t)$
  $j \leftarrow a(t)$
  for all $k \in M : k \neq i$ **do**
    $\bar{p}_j^k \leftarrow \frac{\sum_{h \in M_j^{t-1} \setminus k} \mathrm{sig}(x_{hj} s_h^k) + \alpha}{|M_j^{t-1} \setminus k| + \alpha + \beta}$
    $s_i^k \leftarrow s_i^k + x_{ij} \log(\bar{p}_j^k/(1-\bar{p}_j^k))$
  $z_i^t = \log(q/(1-q)), \forall i \in M$
  for $u = 1$ **to** t **do**
    $i \leftarrow \pi(u)$
    $j \leftarrow a(u)$
    $\bar{p}_j^i \leftarrow \frac{\sum_{h \in M_j^{t-1} \setminus i} \mathrm{sig}(x_{hj} s_h^i) + \alpha}{|M_j^{t-1} \setminus i| + \alpha + \beta}$
    $z_i^t \leftarrow z_i^t + x_{ij} \log(\bar{p}_j^i/(1-\bar{p}_j^i))$
return $\hat{y}_i = \mathrm{sign}(z_i^T), \forall i \in M$

the update in Equation 5.20 and the initialisation in Equation 5.18 become respectively:

$$z_i^t = \log\left(\frac{\mu_i^t(+1)}{\mu_i^t(-1)}\right) = z_i^{t-1} + x_{ij}\log\left(\frac{\bar{p}_j^t}{1-\bar{p}_j^t}\right) \tag{5.22}$$

and

$$z_i^0 = \log\left(\frac{q}{1-q}\right) \tag{5.23}$$

which has both the advantage of converting the chain of products into a summation, and removing the need for normalisation.

Second, we can use the value of the log-odds $\boldsymbol{z}^{t-1}$ at each timestep t to compute the expected predictor's accuracy in Equation 5.19 as follows:

$$\bar{p}_j^t = \frac{\sum_{i \in M_j^{t-1}} \text{sig}(x_{ij} z_i^{t-1}) + \alpha}{|M_j^{t-1}| + \alpha + \beta} \tag{5.24}$$

Thanks to the additive nature of Equation 5.22, we can quickly update the log-odds $\boldsymbol{z}^t$ as we observe new labels. More in detail, we report the full pseudocode of Fast SBIC in Algorithm 5.7. Initially, we set $\boldsymbol{z}^0$ to its prior value in Line 1. Then, we estimate the expected accuracy of the available predictor $j = a(t)$ in Line 5, and add its contribution to the logg-odds on item i in Line 6. At the end of the main iteration, we compute the final predictions by selecting the MAP estimates $\hat{\boldsymbol{y}}$ in Line 8.

This algorithm runs in $O(TL)$ time, where the factor T comes from the main iteration in Lines 2-7, and the maximum number of labels per predictor $L = \max_j \left(|M_j|\right)$ comes from computing each predictor's accuracy in Line 5. As a result, Fast SBIC is particularly effective in an online setting, e.g. when used in conjunction with the adaptive policies US and IG, since it takes only $O(L)$ operations to update its estimates after processing every new label x_{ij}. In Section 5.4.4 we show that its empirical computational speed is on par with the quick majority voting rule, while delivering more than an order of magnitude higher predictive accuracy.

5.2.2.2 Sorted SBIC

When more computational resources are available, we have the opportunity of trading off some of the computational speed of Fast SBIC in exchange for better predictive accuracy. As we mentioned at the end of Section 5.2.2, we can do so by reordering the labels from the dataset X. We call the resulting algorithm Sorted SBIC, and we list its pseudocode in Algorithm 5.8.

The intuition behind this algorithm is the following. When running Fast SBIC, the factors $\boldsymbol{\mu}^t$ and $\boldsymbol{\nu}^t$ are very close to their prior values in the first few rounds. As time passes, two things change. First, we gather more information in the form of additional

data points. Second, we run more update steps on the factors $\boldsymbol{\mu}^t$ and $\boldsymbol{\nu}^t$, which pushes them closer and closer to their respective ground-truth values. As a result, we get more accurate predictions on a specific item i when the corresponding subset of labels X_i is processed towards the end ($t \approx T$) rather than the beginning ($t \approx 0$) of the main iteration.

We can exploit this property by keeping a separate *view* of the log-odds $\boldsymbol{s}^k$ for each item $k \in M$ (see Line 1). Then, every time we observe a new label x_{ij} we update the views for all items k except the one we observed the label on (see Lines 5-7). We skip it because we want to process the corresponding label x_{ij} at the very end of the main iteration, since doing so improves the classification accuracy of the algorithm. Note that in Line 6 we compute a different estimate $\bar{p}_j^k$ for each item $k \neq i$. This is because we are implicitly running $|M|$ copies of Fast SBIC, where each copy (indexed by $k \in M$) can only see its correponding view of the item factors stored in $\boldsymbol{s}^k$.

Finally, we need to process all the labels we skipped. If we are running Sorted SBIC offline, we only need to do so once at the end of the collection process. Conversely, in an online setting we need to repeat the same procedure at each time step t. Lines 8-13 contain the corresponding pseudocode. Notice how we compute the estimates $\bar{p}_j^i$ by looking at all the items M_j^t labelled by predictor j except for item i itself. This is because we skipped the corresponding label x_{ij} in the past, and we are processing it right now.

The implementation of Sorted SBIC presented in Algorithm 5.8 runs in $O(|M|TL)$ time, which is a factor $|M|$ slower than Fast SBIC since we are implicitly running $|M|$ copies of it in parallel (see Lines 5-7). This means that Sorted SBIC is better suited for an offline setting where computational requirements are usually less of a concern. Still, the increase in predictive performance is usually worth the extra computational cost, as we illustrate in our experiments in Section 5.4.2. There, we show that Sorted SBIC achieves state-of-the-art accuracy, on par with Monte Carlo sampling methods.

5.3 Theoretical results

Let us move to our theoretical contribution now. As per Chapter 3 and 4, we are interested in bounding the probability of a classification error given the size of the dataset X, a collection policy π, and a choice of priors q, f_p. However, in contrast with our discussion therein, we have an extra dimension of complexity here: each inference algorithm exhibits a different predictive accuracy. As a result, we need to derive a separate set of bounds not only for each policy π, but also for each different algorithm.

Fortunately, all the results we present in this section are built on top of our contribution in Chapter 4. This means that we can reuse our analysis of classification behaviour

under the different policies, and the corresponding bounds. The only caveat is that this time the distribution of the estimated predictors' accuracies $\hat{\boldsymbol{p}}$ depends on the inference algorithm we use. As a consequence, our main theoretical effort is directed towards computing such distribution $f_{\hat{p}}$.

This fact is reflected in the structure of our material. Except for a brief discussion on the IG policy in Section 5.3.1, all of our results are rather organised by algorithm. Specifically, in Section 5.3.2 we derive a general lower bound on inference in the Dawid-Skene model. This bound is asymptotically tighter than the state-of-the-art result in (Gao et al., 2016) and, for the first time, it covers the US policies as well. Next, in Section 5.3.3 we adapt all the bounds of Chapter 4 to the majority voting algorithm. In doing so, we derive an upper bound on the UNI policy that is tighter than the corresponding result for the state-of-the-art KOS algorithm in (Karger et al., 2014). Finally, in Section 5.3.4 we bound the performance of Fast SBIC and Sorted SBIC from both sides.

5.3.1 A note on the IG policy

Recall that in Sections 3.3.3 and 4.3.3 we prove that the IG policy is equivalent to the US policy. While this is true under the variants of the Dawid-Skene models we analyse therein, the properties of the fully agnostic model of Section 5.1 yield a more complex picture. More specifically, our equivalence result between the US and IG policies remains valid only for a subset of inference algorithms.

To understand this, let us consider the objective of the IG policy: maximising the expected information gain at each timestep t. In order to compute this quantity we need two ingredients. First, the probability of observing each value of the next label $x_{ij} \in \{\pm 1\}$. Second, a measure of the information gain after observing it, i.e. the Kullback-Leibler divergence of the posterior on $\boldsymbol{y}$. With these two, we can write the following objective function:

$$\mathbb{E}_{x_{ij}}\Big(KL\big(X^{t-1} \cup \{x_{ij}\} \big|\big| X^{t-1}\big)\Big) \tag{5.25}$$

where we omit the priors q and f_p for simplicity.

In Chapters 3 and 4 we write a closed-form expression of Equation 5.25 and show that it has the same monotonicity properties of the uncertainty over $\boldsymbol{y}$. This is enough to prove that the US and IG policies are equivalent (see Theorems 3.13 and 4.10). However, in order to replicate this method we need the following conditions:

1. Access to the posterior over the ground-truth class y_i;
2. Access to the posterior over the predictor's accuracy p_j;
3. Independence of the latter from the value of the next label x_{ij}.

While the first two conditions are necessary for computing the probability of observing a specific value of x_{ij}, the third ensures that our estimate $\hat{p}_j$ of the predictor's accuracy does not change after the observation. The latter is crucial to compute the information gain of x_{ij} in closed form.

For some algorithms, all these three conditions are met. The list includes the simple majority voting algorithm and our streaming Bayesian method we introduce in Section 5.2.2. Thus, for these two algorithms we can cover the behaviour of the US and IG policies at the same time. In this light, we present the corresponding results in Section 5.3.3 and 5.3.4.

For other algorithms, this is not the case. Prominent examples are the KOS algorithm in (Karger et al., 2014), Monte Carlo sampling methods (see Section 5.2.1), the approximate mean-field inference (AMF) algorithm in Liu et al. (2012) and the expectation-maximisation (EM) algorithm (Dawid and Skene, 1979). In this respect, the first does not even have a well-defined posterior over the latent variables $\boldsymbol{p}$ and $\boldsymbol{y}$ (we explain how to work around this issue in Section 5.4.1). Conversely, the other three update their full posterior over $\boldsymbol{p}$ and $\boldsymbol{y}$ after observing a new label x_{ij}. As a consequence, the impact of each label on the information gain has a more unpredictable nature, which makes the theoretical analysis of the IG policy impractical.

Furthermore, running the IG policy with this last class of algorithms also poses a computational challenge. Namely, in order to compute the expected information gain in Equation 5.25, we need to first augment the current dataset X^{t-1} with the new label x_{ij}, and then update the posterior over $\boldsymbol{p}$ and $\boldsymbol{y}$. Without a closed-form solution, this operation must be repeated for all possible future labels x_{ij}. Clearly, such a procedure does not scale well with the number of items, since it requires $2|M|$ distinct inferences at each timestep t.

Finally, there is no noticeable improvement in empirical accuracy of this class of algorithms under the IG policy, as opposed as deploying the simpler US policy. We show an example of this in Figure 5.4, where we compare the performance of the AMF algorithm under the US and IG policies (similar results can be obtained for the EM algorithm). For consistency reasons, we use the same experimental setting as Chapters 3 and 4, where we extract the predictors' accuracy from a Beta$(4, 3)$ and each predictor labels $L_j = 10$ items. However, we fix the number of items to only $|M| = 100$ instead of $|M| = 1000$, since running the IG policy would be computationally prohibitive otherwise. For these reasons, we omit the IG policy from our theoretical discussion in the next Section 5.3.2, where we derive a general lower bound that applies to many of the aforementioned algorithms.

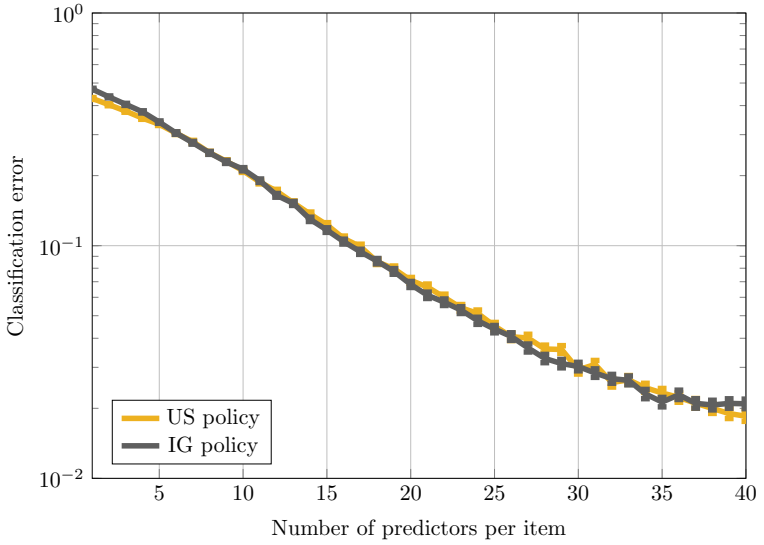


FIGURE 5.4: Comparison between the empirical classification error of the AMF algorithm under the US and IG policies.

5.3.2 General lower bounds on probabilistic inference

For the agnostic Dawid-Skene model of Section 5.1, the predictive accuracy is strongly linked to the inference algorithm we use. However, in most settings the current state-of-the-art algorithms exhibit similar empirical performance, as we show in Section 5.4. From a high-level perspective, this phenomenon suggests that deriving a general bound on the predictive accuracy of these algorithm should be possible.

At the same time, these kind of results are rare in the existing literature, with two notable exceptions. Karger et al. (2014) present a general bound, which proves that the probability of an error decays exponentially in the size of the dataset. Still, this bound is asymptotical and does not give any indication about the rate of exponential decay. Conversely, Gao et al. (2016) propose a method to compute such a rate of decay, which works under the assumption that our estimates of the predictors' accuracies are close to the ground-truth. Unfortunately, this assumption often yields a gross overestimation of the prediction accuracies, as we show in Remark 5.2. Overall, the main drawback of these two results is that they only cover the accuracy of inference under the UNI policy.

In this section we fill this gap in the literature and present our own general results for both the UNI and US policies. These come in the form of lower bounds on the accuracy of inference under the Dawid-Skene model of Section 5.1. Notably, they depend both on the size of the dataset and the number of labels we acquire from each predictor. Consequently, thay take into account the uncertainty on each predictor's accuracy, thus yielding useful estimates on the rate of decay of the prediction error.

The main argument we use is information-theoretic. More specifically, we define a hypothetical inference algorithm that has access to additional information. As a result,

its predictions are more accurate than those of any algorithm that works with the original information only. Furthermore, we define this hypothetical algorithm in such a way that it is possible to bound its predictive accuracy. Crucially, the lower bounds we obtain with this method are also valid for all the other inference algorithms.

Before introducing our hypothetical inference algorithm, let us revise the expression for the marginal probability on the item classes in Equation 5.7. The reason we are interested in it is because its value informs our final predictions $\hat{\boldsymbol{y}}$. Specifically, if we can compute the exact value of the marginal on y_i, then the corresponding maximum a posterior (MAP) estimate $\hat{y}_i$ is optimal. Here, we want to rewrite the closed-form expression for the marginal in Equation 5.7 as follows:

$$\begin{aligned}\mathbb{P}(y_i = \ell | X, q, f_p) &= \int_{\boldsymbol{p}} \mathbb{P}(y_i = \ell, \boldsymbol{p} | X, q, f_p) d\boldsymbol{p} \\ &= \int_{\boldsymbol{p}} \mathbb{P}(y_i = \ell | X_i, q, \boldsymbol{p}) \mathbb{P}(\boldsymbol{p} | X_{-i}, q, f_p) d\boldsymbol{p}\end{aligned} \tag{5.26}$$

While the first equality is obviously true, the second requires more consideration. In particular, the second probability is conditional on the subset $X_{-i} \subseteq X$ of all the labels cast on items different from i. Instead, a straightforward application of the chain rule would require conditioning on the full dataset X. We can prove that Equation 5.26 is equal to the marginal in Equation 5.7 by expanding this second term:

$$\begin{aligned}\mathbb{P}(\boldsymbol{p} | X_{-i}, q, f_p) &= \sum_{\boldsymbol{y}_{-i}} \mathbb{P}(\boldsymbol{y}_{-i}, \boldsymbol{p} | X_{-i}, q, f_p) \\ &\propto \sum_{\boldsymbol{y}_{-i}} \mathbb{P}(X_{-i} | \boldsymbol{y}_{-i}, \boldsymbol{p}) \mathbb{P}(\boldsymbol{y}_{-i} | q) \mathbb{P}(\boldsymbol{p} | f_p) \\ &\propto \sum_{\boldsymbol{y}_{-i}} \mathbb{P}(\boldsymbol{y}_{-i} | q) \prod_{j \in N} p_j^{d(X_j, \boldsymbol{y}_{-i})} (1 - p_j)^{e(X_j, \boldsymbol{y}_{-i})} \mathbb{P}(p_j | f_p)\end{aligned} \tag{5.27}$$

where $\boldsymbol{y}_{-i}$ is a vector assigning a class to each item $i' \neq i$, and the functions $d(X_j, \boldsymbol{y}_{-i})$ and $e(X_j, \boldsymbol{y}_{-i})$ count the number of respectively correct and incorrect labels provided by predictor j. Now, if we substitute Equation 5.27 into Equation 5.26 we get:

$$\begin{aligned}\mathbb{P}(y_i = \ell | X, q, f_p) \propto \sum_{\boldsymbol{y}_{-i}} \mathbb{P}(\boldsymbol{y}_{-i} | q) \int_{\boldsymbol{p}} \mathbb{P}(y_i = \ell | X_i, q, \boldsymbol{p}) \\ \left(\prod_{j \in N} p_j^{d(X_j, \boldsymbol{y}_{-i})} (1 - p_j)^{e(X_j, \boldsymbol{y}_{-i})} \mathbb{P}(p_j | f_p) \right) d\boldsymbol{p} \\ \propto \sum_{\boldsymbol{y}_{-i}} \mathbb{P}(\boldsymbol{y} | q) \prod_{j \in N} \int_0^1 p_j^{d(X_j, \boldsymbol{y})} (1 - p_j)^{e(X_j, \boldsymbol{y})} \mathbb{P}(p_j | f_p) dp_j\end{aligned} \tag{5.28}$$

which is equal to Equation 5.7 once we consider the expression for the posterior on $\boldsymbol{y}$ in Equation 5.5.

Understanding Equation 5.27 is the key here, because it reveals the underlying structure

of the marginal on y_i. That is, we can first compute the posterior on the latent variables $\boldsymbol{p}$ without considering the data on item i, and then use this posterior to form our prediction on y_i. In this light, the accuracy of our prediction y_i depends on how informative the posterior on $\boldsymbol{p}$ is.

In order to derive our bound, we need to build a better inference mechanism. Since Equation 5.27 already represents the exact marginal over y_i, our only option is to introduce more information in the system. We do so by assuming that we have access to the ground-truth classes $\boldsymbol{y}_{-i}$ of all the other items $i' \neq i$. In this setting, we can compute the marginal over y_i as follows:

$$\mathbb{P}(y_i = \ell | X, q, f_p, \boldsymbol{y}_{-i}) = \int_{\boldsymbol{p}} \mathbb{P}(y_i = \ell | X_i, q, \boldsymbol{p}) \mathbb{P}(\boldsymbol{p} | X_{-i}, f_p, \boldsymbol{y}_{-i}) d\boldsymbol{p} \tag{5.29}$$

where the posterior over $\boldsymbol{p}$ is now simpler than the one in Equation 5.27:

$$\begin{aligned} \mathbb{P}(\boldsymbol{p} | X_{-i}, f_p, \boldsymbol{y}_{-i}) &\propto \mathbb{P}(X_{-i} | \boldsymbol{y}_{-i}, \boldsymbol{p}) \mathbb{P}(\boldsymbol{p} | f_p) \\ &= \prod_{j \in N} p_j^{d(X_j, \boldsymbol{y}_{-i})} (1 - p_j)^{e(X_j, \boldsymbol{y}_{-i})} \mathbb{P}(p_j | f_p) \end{aligned} \tag{5.30}$$

Together, Equations 5.29 and 5.30 yield the following expression for the marginal on y_i:

$$\mathbb{P}(y_i = \ell | X, q, f_p, \boldsymbol{y}_{-i}) \propto \mathbb{P}(y_i | q) \prod_{j \in N} \int_0^1 p_j^{d(X_j, \boldsymbol{y})} (1 - p_j)^{e(X_j, \boldsymbol{y})} \mathbb{P}(p_j | f_p) dp_j \tag{5.31}$$

which appears very similar to the one in Equation 5.28, except for the missing summation over $\boldsymbol{y}_{-i}$. Most importantly, Equation 5.31 can be rewritten in a more familiar form by reintroducing the concept of estimates of the predictor's accuracy $\hat{\boldsymbol{p}}$:

$$\mathbb{P}(y_i = \ell | X, q, f_p, \boldsymbol{y}_{-i}) \propto \mathbb{P}(y_i | q) \prod_{j \in N_i} \hat{p}_j^{\mathbb{I}(x_{ij} = y_i)} (1 - \hat{p}_j)^{\mathbb{I}(x_{ij} \neq y_i)} \tag{5.32}$$

where each estimate $\hat{p}_j$ is the probability of observing a correct label from predictor j, given its previous answers on the items $i' \neq i$:

$$\hat{p}_j = \frac{\int_0^1 p_j^{d(X_j, \boldsymbol{y}_{-i}) + 1} (1 - p_j)^{e(X_j, \boldsymbol{y}_{-i})} \mathbb{P}(p_j | f_p) dp_j}{\int_0^1 p_j^{d(X_j, \boldsymbol{y}_{-i})} (1 - p_j)^{e(X_j, \boldsymbol{y}_{-i})} \mathbb{P}(p_j | f_p) dp_j} \tag{5.33}$$

Thanks to this, we can finally describe how our hypothetical inference algorithm works. Formally, for each item $i \in M$ the algorithm has access to the ground-truth vector $\boldsymbol{y}_{-i}$. Thus, it can compute the marginal on y_i according to Equations 5.32 and 5.33. Moreover, it can compute a set of hypothetical prediction $\hat{\boldsymbol{y}}^H$ by taking the MAP estimates of the marginal:

$$\hat{y}_i^H = \underset{\ell \in \{\pm 1\}}{\operatorname{argmax}} \Big\{ \mathbb{P}(y_i = \ell | X, q, f_p, \boldsymbol{y}_{-i}) \Big\} \tag{5.34}$$

By construction, this set of predictions is more accurate than the predictions any other inference algorithm can produce without the knowledge of $\boldsymbol{y}_{-i}$.

With this in mind, we can finally prove the result in the following theorem:

Theorem 5.1. *For any inference algorithm on the agnostic one-coin Dawid-Skene model, the lower bounds in Corollaries 4.4 and 4.9 hold, with the following distribution of estimates of the predictor's accuracy:*

$$f_{\hat{p}}(p) = \frac{1}{|T|} \sum_{j \in N} L_j \sum_{d=0}^{L_j - 1} \mathbb{P}(d | L_j - 1, f_p) \delta\Big(p - \hat{p}_j(d, L_j - 1, f_p)\Big) \tag{5.35}$$

where $\delta(\bullet)$ is Dirac's delta, $\hat{p}_j$ is computed as in Equation 5.33 with d correct answers in the past and $e = L_j - d - 1$ incorrect ones, and the probability of d successes out of L' trials is:

$$\mathbb{P}(d | L', f_p) = \binom{L'}{d} \int_0^1 p_j^d (1 - p_j)^{L' - d} \mathbb{P}(p_j | f_p) dp_j \tag{5.36}$$

Proof. Let us focus on the predictions of the hypothetical inference algorithm in Equation 5.34. These are the MAP estimates given the marginal posterior on each item i. More in detail, the corresponding marginal in Equation 5.32 fits the framework we study in Chapter 4: that is, the posterior on y_i is computed independently given an estimate of the predictor's accuracy $\hat{\boldsymbol{p}}$, the label cast on item i, and the prior on the item classes q.

The only difference is in how the estimates $\hat{\boldsymbol{p}}$ are computed. In Chapter 4 we have an extra set of items G with known ground-truth classes $\boldsymbol{g}$. Here, we have access to the ground-truth classes $\boldsymbol{y}_{-i}$ of the other items $i' \neq i$ instead. Similarly, in Chapter 4 we have an additional set of labels O the predictors cast on G, while here we have access to the set X_{-i} on the items M_{-i}.

These differences are just a matter of notation though. In fact, the bounds in Corollaries 4.4 and 4.9 hold for any vector of unbiased estimates $\hat{\boldsymbol{p}}$, which is still the case here (see Equation 5.33). Moreover, the results in these two theorems depend on the distribution $f_{\hat{p}}$ of the predictors' estimates, rather than the estimates themselves. Luckily, we can compute such a distribution in the following way.

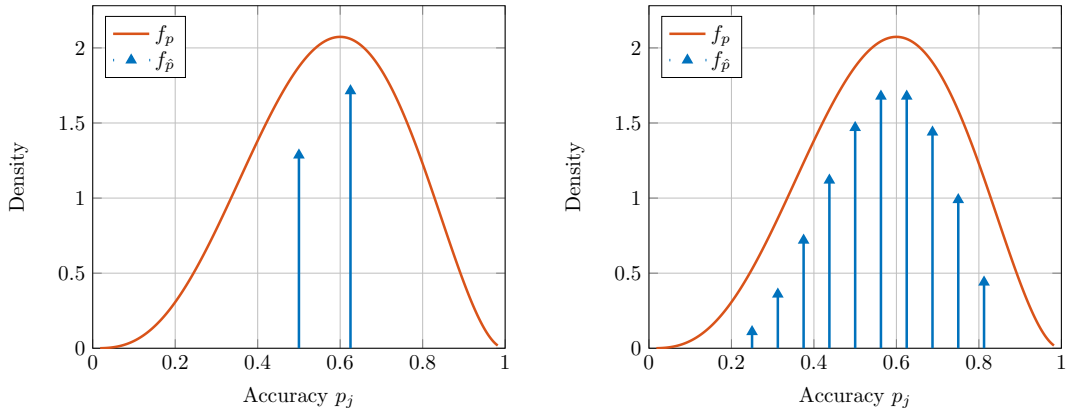
First, recall that each estimate $\hat{p}_j$ in Equation 5.33 depends on three quantities: the number of successes on the other items $d = d(X_j, \boldsymbol{y}_{-i})$, the number of failures $e = e(X_j, \boldsymbol{y}_{-i})$, and the prior $p_j \sim f_p$. Note that the number of failures can be computed as $e = L_j - d - 1$, where $L_j = |X_j|$ is the total number of labels provided by predictor j, and we ignore the label they cast on item i. Then, the probability of observing d successes out of $L_j - 1$ trials is shown in Equation 5.36.

Second, for each d we need to place a spike in the (discrete) probability distribution $f_{\hat{p}}$ (see Equation 5.35). The spikes are Dirac's deltas centred on the corresponding value of

the predictor's estimate $\hat{p}_j$. Moreover, we weigh the contribution of each predictor by the number of labels L_j it provides, and normalise by the total number of labels T, so that the integral of the distribution sums up to one.

As we discuss throughout the present section, the predictive accuracy of the hypothetical inference algorithm in Equation 5.34 is provably superior to that of any other inference algorithm. Thus, the lower bounds in Corollaries 4.4 and 4.9 with the distribution $f_{\hat{p}}$ in Equation 5.35 are also valid for any inference algorithm on the agnostic Dawid-Skene model. □

Theorem 5.1 shows the link between the amount of data we have on each predictor and the resulting classification accuracy. In fact, the more data we have, the better we can estimate each predictor's accuracy. In turn, this improves the accuracy of our predictions $\hat{\boldsymbol{y}}$. We give a visual explanation on why this is the case in Figure 5.5: note how for a larger value of L_j the distribution of estimates $f_{\hat{p}}$ matches the underlying distribution f_p more closely. In the limit for $L_j \to \infty$, the two distribution overlap and the lower bounds in Corollaries 4.4 and 4.9 reduce to their corresponding versions with known predictors' accuracies $\boldsymbol{p}$ (see Corollaries 3.7 and 3.12).



(a) $L_j = 2$ labels per predictor.

(b) $L_j = 10$ labels per predictor.

FIGURE 5.5: Comparison between the posterior distribution $f_{\hat{p}}$ on the predictor's accuracy in Theorem 5.1, and an example of prior distribution $f_p = \text{Beta}(4, 3)$.

The latter case is the one explored in (Gao et al., 2016) under the UNI policy, where it is shown that most state-of-the-art algorithms do reach this limit regime eventually. However, we argue that ignoring the uncertainty in the estimates $\hat{\boldsymbol{p}}$ when L_j is small can lead to serious overestimation of the predictive accuracy:

Remark 5.2. Theorem 5.1 yields a larger asymptotical term for the UNI policy than the corresponding result in (Gao et al., 2016):

$$\mathbb{P}\left(\hat{y}_i \neq y_i | \hat{\boldsymbol{p}} \to \boldsymbol{p}, q = \frac{1}{2}\right) \leq \exp\Big(- \big(1 + o(1)\big) \sum_{j \in N_i} \log\big(2\sqrt{p_j(1-p_j)}\big)\Big) \tag{5.37}$$

Proof. Before dealing with the details of our proof, let us point out two things. First, the result in Equation 5.37 is an upper bound only, as no corresponding lower bound is given in (Gao et al., 2016). Second, the result in Equation 5.37 is based on the assumption that $\hat{\boldsymbol{p}} \to \boldsymbol{p}$, where the mismatch error is absorbed into the $o(1)$ term. Here, we are interested in the setting where such an assumption does not hold, and the upper bound in Equation 5.37 becomes invalid.

Referring back to Corollary 4.4, we can see that the crux of our argument lies in the expected value of the square root, as this is the term that dominates asymptotically. More formally, we can restate our claim in the following form:

$$\mathbb{E}_{\hat{p}_j \sim f_{\hat{p}}}\Big(\sqrt{\hat{p}_j(1-\hat{p}_j)}\Big) \geq \mathbb{E}_{p_j \sim f_p}\Big(\sqrt{p_j(1-p_j)}\Big) \tag{5.38}$$

where the left-hand side comes from our bounds in Corollary 4.4, and the right-hand side is the same term under the assumption that $\hat{\boldsymbol{p}} \to \boldsymbol{p}$ like in Equation 5.37.

Intuitively, the inequality in Equation 5.38 is always satisfied since $\sqrt{p(1-p)}$ is a concave function, and $f_{\hat{p}}$ is an approximation of f_p that tends to put more mass close to the prior expected value. However, this fact can only be proven numerically. A notable exception is when $L_j = 1$, and $f_{\hat{p}}(p) = \delta\big(p - \bar{p}\big)$ as per Theorem 5.1. In this case, all predictors' estimates $\hat{\boldsymbol{p}}$ are equal to the prior expected value $\bar{p} = \mathbb{E}_{p_j \sim f_p}(p_j)$, and the following inequality holds because of concavity:

$$\sqrt{\bar{p}\big(1-\bar{p}\big)} \geq \mathbb{E}_{p_j \sim f_p}\Big(\sqrt{p_j(1-p_j)}\Big) \tag{5.39}$$

□

We show in Section 5.4 how the lower bound in Theorem 5.1 compares to the empirical performance of the state-of-the-art algorithms. Moreover, we prove in Section 5.3.4 that the Sorted SBIC algorithm matches this bound asymptotically.

5.3.3 Upper and lower bounds on majority voting

If we want bounds that are even closer to the actual predictive performance, we need to abandon generality and focus on one single inference algorithm at a time. In this section, we focus on the accuracy of majority voting under both the UNI and US policies. Despite the simplicity of this algorithm, the bounds we derive turn out to be quite informative. As an example, the upper bound on the UNI policy is tighter than the corresponding result for the state-of-the-art KOS algorithm (Karger et al., 2014) in most settings. In this light, the majority voting results we present here constitute a useful baseline for any future theoretical work.

Let us begin with the UNI policy. In order to bound the accuracy of majority voting in

this context, we go back to our result in Corollary 4.4, which depends on the distribution of the estimates $\boldsymbol{p}$. Here, the domain of such a distribution reduces to a single value $\bar{p}_j = \mathbb{E}_{p_j \sim f_p}(p_j)$, thus yielding the following corollary:

Corollary 5.3. *Given a population of predictors with unknown accuracy $p_j \sim f_p$, a prior on the positive class $q \equiv \mathbb{P}(y_i = +1)$, and R predictors per item, the probability of a classification error by the majority voting algorithm under the UNI policy is upper bounded by:*

$$\mathbb{P}(\hat{y}_i \neq y_i | f_p, q) \leq \exp\left(\frac{1}{2}\log\big(q(1-q)\big) + \frac{R}{2}\log\big(4\bar{p}_j(1-\bar{p}_j)\big)\right) \tag{5.40}$$

and lower bounded by:

$$\mathbb{P}(\hat{y}_i \neq y_i | f_p, q) \geq \exp\left(\frac{1}{2}\log\big(q(1-q)\big) + \frac{R}{2}\log\big(4\bar{p}_j(1-\bar{p}_j)\big) - \sqrt{R}|\bar{w}_j| - 0.32\right) \tag{5.41}$$

for all $i \in M$, where $\bar{p}_j = \mathbb{E}_{p_j \sim f_p}(p_j)$ and $\bar{w}_j = \log\big(\bar{p}_j/(1-\bar{p}_j)\big)$.

Proof. Take Equations 4.23 and 4.24 in Corollary 4.4, and substitute $\hat{p}_j = \bar{p}_j$ for all $j \in N$. Since all the estimates $\hat{\boldsymbol{p}}$ are identical, the expectation over $f_{\hat{p}}$ disappears. Finally, the constant term in front of the lower bound can be propagated inside the exponential by taking into account that:

$$\log\big(1 - 2\exp(-2)\big) \approx -0.3156297512 \tag{5.42}$$

which concludes the proof. □

From a purely theoretical perspective, the results in Corollary 5.3 have an intrinsic weakness. That is, they take into account only the average accuracy $\bar{p}_j$ of the predictors. For instance, if we compare the lower bound in Equation 5.41 with the general case in Theorem 5.1, we notice that they are equivalent only when each predictor provides us with only one label, i.e. $L_j = 1$. For any other case, we expect a more powerful inference algorithm to achieve better accuracy than majority voting.

At the same time, the bounds in Corollary 5.3 do give us a general indication on how well any other inference algorithm can perform, albeit in a very conservative way. In this sense, they can be tighter than some other celebrated bounds on state-of-the-art algorithms. A prominent example is the following:

Remark 5.4. The upper bound in Equation 5.40 of Theorem 5.3 is tighter than the following upper bound over the KOS algorithm proposed in (Karger et al., 2014):

$$\mathbb{P}\left(\hat{y}_i \neq y_i | f_p, q = \frac{1}{2}\right) \leq 2\exp\left(-\frac{R}{32}\mathbb{E}_{p_j \sim f_p}\Big((2p_j - 1)^2\Big)\right) \tag{5.43}$$

for any distribution which satisfy the following property:

$$\mathbb{E}_{p_j \sim f_p}\Big((2p_j - 1)^2\Big) < -16 \log\big(4\bar{p}_j(1-\bar{p}_j)\big) \tag{5.44}$$

Proof. If we inspect the asymptotic term in Equations 5.40 and 5.43, the former yields a quicker exponential decay than the latter when the inequality in Equation 5.44 is satisfied. □

With the risk of repeating ourselves, we want to stress the following idea. Remark 5.4 does not imply that majority voting performs better than the KOS algorithm. In practice, the opposite is often true, even for instances of f_p that do not satisfy Equation 5.44 (see Section 5.4). Instead, Remark 5.4 informs us that, just by analysing the behaviour of majority voting, we get a theoretical result which is stronger than the corresponding analysis for the KOS algorithm.

Let us now bound the accuracy of majority voting under the US policy. As for Corollary 5.3, we derive the following bounds as an extension of our results from Chapter 4:

Corollary 5.5. *Given a population of predictors with unknown accuracy $p_j \sim f_p$, prior on the positive class $q \equiv \mathbb{P}(y_i = +1)$, a large number of items $|M| \to \infty$, and an average of R predictors per item, the probability of a classification error by the majority voting algorithm under the US and IG policies is upper bounded by:*

$$\mathbb{P}(\hat{y} \neq y | f_p, q) \leq \exp\Big(-(R-1)\big((2\bar{p}_j - 1)\bar{w}_j\big) - (2q-1)w_q \Big) \tag{5.45}$$

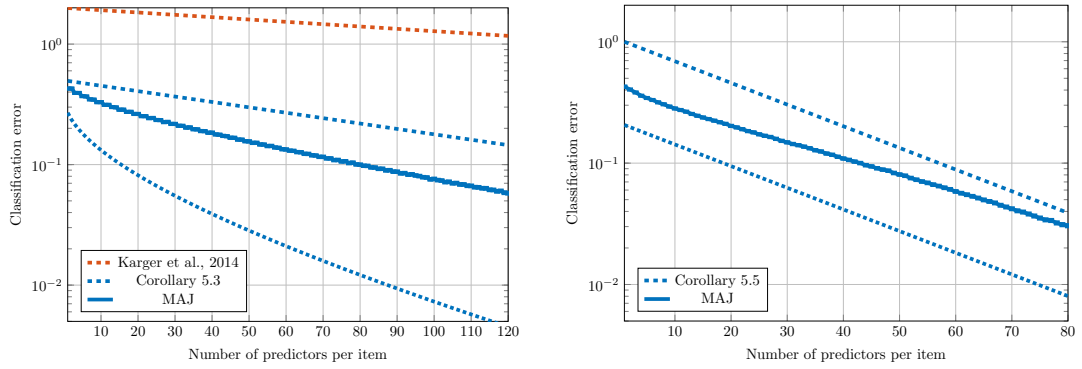
and lower bounded by:

$$\mathbb{P}(\hat{y} \neq y | f_p, q) \geq \exp\Big(-R\big((2\bar{p}_j - 1)\bar{w}_j\big) - (2q-1)w_q - 1.25 - |\bar{w}_j| \Big) \tag{5.46}$$

for all $i \in M$, where $\bar{p}_j = \mathbb{E}_{p_j \sim f_p}(p_j)$, $\bar{w}_j = \log\big(\bar{p}_j/(1-\bar{p}_j)\big)$ and $w_q = \log\big(q(1-q)\big)$.

Proof. Take the upper and lower bounds in Corollaries 4.8 and 4.9 respectively. Substitute $\hat{p}_j = \bar{p}_j$ and $\hat{w}_j = \bar{w}_j$ for all predictors $j \in N$. Since these are identical for all predictors, the expectation operator disappears thus concluding our proof. □

We present a qualitative comparison between the theoretical bounds in Corollaries 5.3, 5.5 and the empirical performance of majority voting in Figure 5.6. The experimental setting is the same as the one we study in Chapters 3 and 4. Namely, we fix the number of items to $|M| = 1000$, the distribution of the predictors' accuracy to $p_j \sim \text{Beta}(4, 3)$, and let each predictor label $L_j = 10$ items. We repeat each experiments 1000 times under both the UNI and US/IG policies. Note how the upper bounds in Corollaries 5.3 and 5.5 are a good representation of the empirical performance of MAJ. Also, the upper bound for KOS algorithm in (Karger et al., 2014) is less tight in this setting, as we discuss in Remark 5.4.



(a) UNI policy.

(b) US and IG policies.

FIGURE 5.6: Comparison between the empirical performance of the majority voting (MAJ) algorithm, the existing result in (Karger et al., 2014), and our bounds in Corollaries 5.3 and 5.5. Dashed lines are the upper bounds, dotted lines are the lower bounds.

5.3.4 Upper and lower bounds on SBIC

We conclude our brief tour of algorithm-dependent bounds with a theoretical analysis of the SBIC algorithm. Before dealing with the details of our contribution, let us preface them with quick note on the research process that led to them. Originally, the SBIC algorithm we present in Section 5.2.2 was meant purely as a theoretical device. In fact, the goal was to design an inference method whose properties would allow for strong guarantees on its predictive accuracy. In doing so, we ended up with an algorithm that has good empirical performance too (see our experiments in Section 5.4 for details). Nonetheless, we still managed to fulfil that original goal, and thus we dedicate the present section to it.

A key property of the SBIC algorithm in all its variants is that the estimates $\hat{\boldsymbol{p}}$ of the predictors' accuracies only depends on past information. In other terms, for each timestep t, the current predictor's estimate $\hat{p}_j^t$ is independent from the value of the label x_{ij} we observe. In this sense, the SBIC algorithm belongs to the framework of Chapter 4, where the posterior on each item class y_i is computed using the data on that particular item X_i and the estimates $\hat{\boldsymbol{p}}$. The only difference here is that SBIC computes the estimates $\hat{\boldsymbol{p}}$ using the labels cast on the other items X_{-i} and the posterior on the respective classes $\boldsymbol{y}_{-i}$, rather than the observations O on the extra set of items G.

We can take advantage of this property by making an argument similar to the one in Section 5.3.2. That is, if we assume that the ground-truth classes $\boldsymbol{y}_{-i}$ on all the other items $i' \neq i$ are known, then the corresponding estimates $\hat{\boldsymbol{p}}$ yield more accurate predictions on y_i. With this assumption, we can easily compute the distribution of estimates $f_{\hat{p}}$, and plug it in the results of Chapter 4. The resulting lower bounds are listed in the following corollary:

Corollary 5.6. *Given a population of predictors with distribution* $f_p = Beta(\alpha, \beta)$,

the probability of a classification error by the Streaming Bayesian Inference for Crowdsourcing (SBIC) algorithm satisfies the lower bounds in Corollaries 4.4 and 4.9, with a distribution of estimates $f_{\hat{p}}$ that depends on the variant of SBIC we use. For Sorted SBIC we have the same distribution as in Theorem 5.1:

$$f_{\hat{p}}^{Sorted}(p) = \frac{1}{|T|} \sum_{j \in N} L_j \sum_{d=0}^{L_j - 1} \mathbb{P}(d | L_j - 1, \alpha, \beta) \delta\Big(p - \hat{p}_j(d, L_j - 1, \alpha, \beta)\Big) \tag{5.47}$$

where $\delta(\bullet)$ is Dirac's delta, L_j is the number of labels provided by predictor j, and d_j is the number of correct ones. For Fast SBIC we have the following instead:

$$f_{\hat{p}}^{Fast}(p) = \frac{1}{|T|} \sum_{j \in N} \sum_{L'=0}^{L_j - 1} \sum_{d=0}^{L'} \mathbb{P}(d | L', \alpha, \beta) \delta\Big(p - \hat{p}_j(d, L', \alpha, \beta)\Big) \tag{5.48}$$

where the probability of d successes out of L' trials is the same as Equation 5.36.

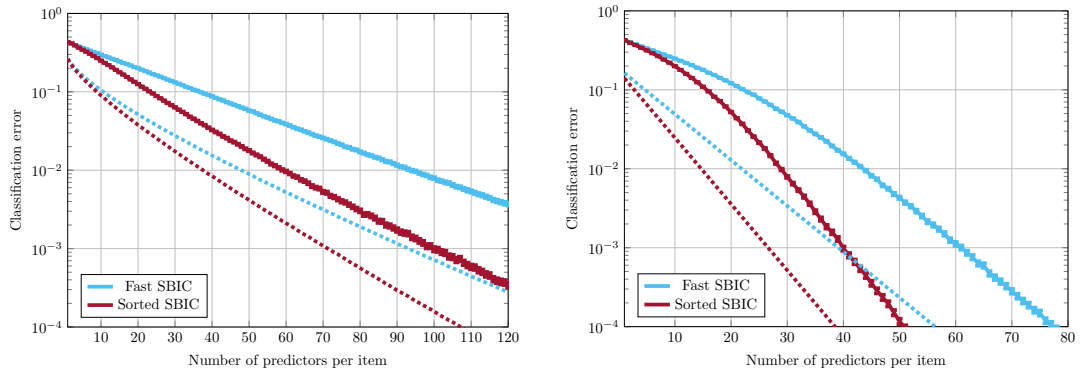
Proof. In a similar way as the proof of Theorem 5.1, we focus on the predictive accuracy of a hypothetical version of SBIC that, for each item $i \in M$, has access to the ground-truth $\boldsymbol{y}_{-i}$ on the other items $i' \neq i$. We know from our discussion in Section 5.3.2 that this hypothetical algorithm has provably better predictive accuracy. As a result, its probability of a classification error is always smaller than the one of SBIC.

What we are left with is deriving an expression for $f_{\hat{p}}$. For Sorted SBIC, we known that the final posterior on y_i is computed after observing all the data on the other items X_{-i}. Consequently, each corresponding estimate $\hat{p}_j$ is computed on the $L_j - 1$ past responses of predictor j on items $M \setminus i$. Since our hypotethical algorithm has access to the ground-truth classes of such items, the estimate $\hat{p}_j$ depends on the number of correct answers d out of $L_j - 1$ trials. This is the same case as our general result in Theorem 5.1, except for the additional assumption by the SBIC algorithm that $f_p = \text{Beta}(\alpha, \beta)$.

For Fast SBIC, the number of past responses L' each predictor is evaluated on increases during the data collection process. At the beginning each new predictor is tested on $L' = 0$ past answers, at the end on $L' = L - j - 1$ of them. Accordingly, we modify the expression for $f_{\hat{p}}$ in Equation 5.47 to account for the increasing value of L'. More specifically, in Equation 5.48 we sum over all values $L' \in [0, L_j - 1]$, instead of weighing the contribution of predictor j by its productivity L_j. □

We can compare the lower bounds in Corollary 5.6 with the actual empirical performance of SBIC by running both variants of the algorithm on synthetic data. We do so in Figure 5.7, where we choose the same setting of our experiments for the majority voting algorithm in Section 5.3.3. Specifically, we set the number of items to $|M| = 1000$, extract the predictors' accuracy according to $p_j \sim \text{Beta}(4, 3)$, and let the predictors label $L_j = 10$ items each. As the results in Figure 5.7 show, the lower bounds in Corollary

5.6 capture the differences between Fast SBIC and Sorted SBIC well. Moreover, they match the asymptotic decay in the error rate of both algorithms.



(a) UNI policy.

(b) US and IG policies.

FIGURE 5.7: Comparison between the lower bounds in Corollary 5.6 (dotted lines) and the empirical performance of the Fast SBIC and Sorted SBIC algorithms.

Computing an upper bound is more difficult, and thus we study the performance of the SBIC algorithm in its asymptotic regime. This regime emerges when enough data has been observed and the posterior on the item classes is converging towards its ground-truth values $\boldsymbol{y}$. We formalise this concept in the following way:

Definition 5.7. For any value of the item factors $\boldsymbol{\mu}^t$ at time t, define ϵ_1 and ϵ_2 as a pair of small positive numbers in $\mathbb{R}$ such that:

$$\frac{1}{|M|} \sum_{i \in M} \mathbb{I}\Big(\mu_i^t(y_i) < 1 - \epsilon_1\Big) < \epsilon_2 \tag{5.49}$$

where y_i is the ground-truth class of item i.

Note that Definition 5.7 is quite general, in the sense that there is a pair $\epsilon_1, \epsilon_2 \in [0, 1]$ that satisfies Equation 5.49 for any $\boldsymbol{\mu}^t$. At the same time, when ϵ_1 and ϵ_2 are small, this definition binds the value of $\boldsymbol{\mu}^t$ in a stronger way than a mere upper bound on the prediction error. More in detail, the latter is measured in terms of the 0-1 error loss $\frac{1}{|M|} \sum_{i \in M} \mathbb{I}\big(\mu_i^t(y_i) < 1/2\big) < \epsilon_2$. In contrast, Equation 5.49 requires both the probability of an error and the uncertainty ϵ_1 in our item factors $\boldsymbol{\mu}^t$ to be small. As a consequence, for small ϵ_1 and ϵ_2 the item factors $\mu_i^t(\ell)$ are close to the indicator function $\mathbb{I}(\ell = y_i)$ with high probability.

In order to derive our upper bound, we compare the estimates of the predictors' accuracy $\bar{\boldsymbol{p}}$ computed by SBIC and those computed by the hypothetical algorithm of Theorem 5.6. To avoid confusion, we denote the former with $\bar{p}(X_j, \boldsymbol{\mu})$ and the latter with $\bar{p}(X_j, \boldsymbol{y})$.

Under Definition 5.7, we have:

$$\begin{aligned}\epsilon &= \left|\bar{p}(X_j, \boldsymbol{\mu}) - \bar{p}(X_j, \boldsymbol{y})\right| \\ &= \left|\frac{\sum_{i \in M_j} \mu_i(x_{ij}) - \mathbb{I}(x_{ij} = y_i)}{|M_j| + \alpha + \beta}\right| \\ &= \left|\frac{\sum_{i \in M_j} \mu_i(-y_i)\big(\mathbb{I}(x_{ij} = -y_i) - \mathbb{I}(x_{ij} = y_i)\big)}{|M_j| + \alpha + \beta}\right| \\ &\leq \frac{\sum_{i \in M_j} \mu_i(-y_i)}{|M_j| + \alpha + \beta} \leq \frac{|M_j|(\epsilon_1 + \epsilon_2)}{|M_j| + \alpha + \beta} \leq \epsilon_1 + \epsilon_2\end{aligned} \tag{5.50}$$

In other terms, each estimate $\bar{p}(X_j, \boldsymbol{\mu})$ used by the SBIC algorithm is always in a ϵ neighbourhood of the optimal estimate $\bar{p}(X_j, \boldsymbol{y})$. If this neighbourhood is small at a given time t', we can prove that the item factors $\boldsymbol{\mu}^t$ for $t > t'$ converge to the ground-truth as we observe more and more data. We do so separately for the UNI and US policy:

Theorem 5.8. *Given a population of predictors with distribution $f_p = Beta(\alpha, \beta)$ and R predictors per item, the probability of a classification error of the SBIC algorithm under the UNI policy goes to zero as $R \to \infty$, if there exists a timestep t such that Definition 5.7 holds with ϵ_1, ϵ_2, and:*

$$\mathbb{E}_{X_j|\boldsymbol{y},\alpha,\beta}\left\{\max_{\hat{p} \in \left[\bar{p}(X_j,\boldsymbol{y}) \pm \epsilon\right]} \left[\left(\frac{\bar{p}(X_j, \boldsymbol{y})}{\hat{p}} + \frac{1 - \bar{p}(X_j, \boldsymbol{y})}{1 - \hat{p}}\right)\sqrt{\hat{p}(1 - \hat{p})}\right]\right\} < 1 \tag{5.51}$$

where $\epsilon = \epsilon_1 + \epsilon_2$ as in Equation 5.50.

Proof. Theorem 4.1 provides us with an upper bound on the probability of a classification error on a single item, given a plug-in aggregator and vectors of real and estimated accuracies $\boldsymbol{p}$ and $\hat{\boldsymbol{p}}$ respectively. In particular, we know that the probability of an error depends on each predictor j according to the following multiplicative factor:

$$g(p_i, \hat{p}_j) = \left(\frac{p_j}{\hat{p}_j} + \frac{1 - p_j}{1 - \hat{p}_j}\right)\sqrt{\hat{p}_j(1 - \hat{p}_j} \tag{5.52}$$

We can extend the result in Equation 5.52 to our current setting by taking the expectation of p_j with respect to f_p. In this way, we can derive the average contribution of a generic predictor, given the set of answers X_j it provided in the past:

$$\mathbb{E}_{X_j|\boldsymbol{y},\alpha,\beta}\Big(g(p_i, \hat{p}_j)\Big) = \left(\frac{\bar{p}(X_j, \boldsymbol{y})}{\hat{p}_j} + \frac{1 - \bar{p}(X_j, \boldsymbol{y})}{1 - \hat{p}_j}\right)\sqrt{\hat{p}_j(1 - \hat{p}_j} \tag{5.53}$$

where $\bar{p}(X_j, \boldsymbol{y}) = \mathbb{E}_{X_j|\boldsymbol{y},\alpha,\beta}(p_j)$ by definition.

Now, we need to consider two things. First, if the factor in Equation 5.53 is smaller than one, observing more labels is guaranteed to lower the probability of an error on average.

As the number of observations R grows to infinity, the probability of a classification will converge to zero. Second, the SBIC algorithm computes the value of $\hat{p}_j = \bar{p}(X_j, \boldsymbol{\mu})$ given both X_j and $\boldsymbol{\mu}$. We can remove the dependency on the latter from Equation 5.53 by using the theorem assumption that $\bar{p}(X_j, \boldsymbol{\mu}) \in [\bar{p}(X_j, \boldsymbol{y}) \pm \epsilon]$ for some timestep t. Since we are interested in proving convergence, we write the worst-case scenario where Equation 5.53 is maximised, yielding the result in Equation 5.52.

Finally, if the condition in the present theorem are met for a given timestep t, any subsequent observation at $t' > t$ will make $\mu_i^t(y_i)$ drift towards one for each $i \in M$. In turn, the value of $\epsilon = \epsilon_1 + \epsilon_2$ will converge to zero, yielding even smaller values for the expression in Equation 5.53. □

Theorem 5.9. *Given a population of predictors with distribution $f_p = Beta(\alpha, \beta)$, an average of R predictors per item, and a large number of items $|M| \to \infty$, the probability of a classification error of the SBIC algorithm under the US policy goes to zero as $R \to \infty$, if there exists a timestep t such that Definition 5.7 holds with ϵ_1, ϵ_2, and:*

$$\mathbb{E}_{X|\boldsymbol{y},\alpha,\beta}\left\{ \min_{\tilde{p} \in \left[\bar{p}_j(X,\boldsymbol{y})-\epsilon, \bar{p}_j(X,\boldsymbol{y})+\epsilon\right]} \left(\left(2\bar{p}_j(X,\boldsymbol{y}) - 1\right) \log\left(\frac{\tilde{p}}{1-\tilde{p}}\right)\right)\right\} > 0 \tag{5.54}$$

where $\epsilon = \epsilon_1 + \epsilon_2$ as in Equation 5.50.

Proof. Theorem 4.6 proves that the average number of predictors required to reach a target (reported) accuracy $1-p_e$ is finite. The key message is that each predictor changes the log odds on an item i by an additive random step. The corresponding random walk has the following average drift:

$$\mathbb{E}_{p_j \sim f_p, \hat{p}_j \sim f_{\hat{p}}}\left((2p_j - 1) \log\left(\frac{\hat{p}_j}{1-\hat{p}_j}\right)\right) \tag{5.55}$$

which has to be strictly positive.

As in the proof of Theorem 5.8, we extend the result of Equation 5.55 to our current setting by taking the expectation of p_j with respect to f_p explicitly. Accordingly, we write the average drift induced by a predictor j, given the set of answers X_j it provided in the past:

$$\mathbb{E}_{X_j|\boldsymbol{y},\alpha,\beta}\left((2\bar{p}(X_j,\boldsymbol{y}) - 1) \log\left(\frac{\hat{p}_j}{1-\hat{p}_j}\right)\right) \tag{5.56}$$

where $\bar{p}(X_j, \boldsymbol{y}) = \mathbb{E}_{X_j|\boldsymbol{y},\alpha,\beta}(p_j)$ by definition.

Unfortunately, the dependency of the SBIC estimates $\hat{p}_j = \bar{p}(X_j, \boldsymbol{\mu})$ on X and $\boldsymbol{y}$ is not straightforward. We solve this issue by considering the worst-case scenario. Since we are interested in proving that Equation 5.56 is positive, we minimise over all possible values of $\bar{p}(X_j, \boldsymbol{\mu}) \in [\bar{p}(X_j, \boldsymbol{y}) \pm \epsilon]$. The result of this operation is shown in Equation 5.54.

Crucially, if the conditions in the present theorem are met for a given timestep t, then we can reach any level of reported accuracy $\mu_i^{t'}(y_i) = 1 - p_e$ we want. Furthermore, the lower we set p_e and the more observations we collect from the predictors, the smaller the value of $\epsilon = \epsilon_1 + \epsilon_2$ will be. In turn, this ensures convergence of $\bar{p}$ to $\bar{p}(X_j, \boldsymbol{y})$, thus making the value of Equation 5.54 larger. Finally, even if p_e is only a reported accuracy, the fact that $\mu_i^{t'}(y_i) \to 1$ as $p_e \to 0$ guarantees that the probability of a classification error goes to zero. □

Theorems 5.8 and 5.9 provide us with sufficient conditions for the predictions of the SBIC algorithm to converge to the ground-truth. To summarise, if the item factors $\boldsymbol{\mu}$ are concentrated enough towards the item classes $\boldsymbol{y}$, then we can guarantee that further observations will make $\boldsymbol{\mu}$ converge to $\boldsymbol{y}$ even further. After this initial transient, we can study the rate of decay of the misclassification error with the same technique as in Corollary 5.6. The result are the following two asymptotical upper bounds:

Corollary 5.10. *Given that the conditions in Theorem 5.8 are met, the probability of a classification error by the Streaming Bayesian Inference for Crowdsourcing (SBIC) algorithm under the UNI policy is upper bounded by:*

$$\mathbb{P}(\hat{y}_i \neq y_i | \alpha, \beta, q) \leq \exp\Big(- R \log\Big(\mathbb{E}_{\hat{p}_j \sim f_{\hat{p}}}\big(2\sqrt{\hat{p}_j(1-\hat{p}_j)}\big)\Big) + o(R)\Big) \tag{5.57}$$

for all items $i \in M$, where the distribution $f_{\hat{p}}$ depends on the variant of SBIC we use, and is the same as in Corollary 5.6.

Similarly, given that the conditions in Theorem 5.9 are met, the probability of a classification error by SBIC under the US and IG policies is upper bounded by:

$$\mathbb{P}(\hat{y}_i \neq y_i | \alpha, \beta, q) \leq \exp\Big(- R \mathbb{E}_{\hat{p}_j \sim f_{\hat{p}}}\big((2\hat{p}_j - 1)\hat{w}_j\big) + o(R)\Big) \tag{5.58}$$

for all items $i \in M$, where $\hat{w}_j = \log\big(\hat{p}_j/(1-\hat{p}_j)\big)$ and $f_{\hat{p}}$ is in Corollary 5.6.

Proof. First, Theorems 5.8 and 5.9 ensure that, after a specific timestep t, the difference $\epsilon = \epsilon_1 + \epsilon_2$ between the estimates $\bar{p}(X_j, \boldsymbol{\mu})$ and $\bar{p}(X_j, \boldsymbol{y})$ in Equation 5.50 converges to zero. Thus, in this asymptotical regime, the SBIC algorithm behaves as its hypothetical counterpart in Corollary 5.6. Specifically, the distribution of estimates $f_{\hat{p}}$ is the same.

Second, before reaching this asymptotic regime, the approximated marginal posterior $\mu_i(y_i)$ on each item $i \in M$ may become biased with respect to its hypothetical counterpart $\mathbb{P}(y_i | X, \alpha, \beta, q, \boldsymbol{y}_{-i})$. In general, this bias increases for each contribution of the predictors $j \in N_i$, though such contributions become negligible as $R \to \infty$ since the estimates $\bar{p}_j$ converge to their unbiased version. This accounts for the $o(R)$ term in Equations 5.57 and 5.58.

Finally, the asymptotical term in the upper bounds of Corollaries 4.4 and 4.8 applies here. In fact, after accounting for the bias of the initial transient, each contribution of the predictors in the asymptotic regime reduces the probability of an error by the factor specified therein. □

Taken together, Corollaries 5.6 and 5.10 ensure that the asymptotic performance of SBIC follows an exponential decay in classification error as the number of predictors per item R increases. In particular, the rate of decay depends on the variant of the algorithm and the collection policy we use, as we illustrate in Figure 5.7. At the same time, the performance of the SBIC algorithm is also sensitive to the number of items L_j each predictor labels. We show this theoretically in Corollary 5.6, as our result there depends on L_j. For more empirical evidence on that, see our experiments in Section 5.4.2.

5.4 Experimental comparison

After presenting our algorithmic and theoretical contributions, it is now time check how well they hold up empirically. Indeed, we already present some empirical results in the corresponding Sections 5.2 and 5.3. However, they only serve the purpose of illustrating our main arguments therein. Here we make a more structured analysis.

In particular, we want to compare our take on the inference challenge of Section 5.2 with the existing algorithms in the current literature. For ease of reference, we list them again in the following Section 5.4.1, together with their implementation details and parameter settings. These algorithms span a wide range of techniques, from belief propagation to the method of moments and from expectation-maximisation to variational inference. Still, in many of our experiments they end up exhibiting very similar predictive performance.

In this regard, we organise our empirical results according to three research questions. First, we study the performance of all the inference algorithms on synthetic data (see Section 5.4.2). This gives us full control on the properties of the data we produce. Furthermore, it allows us to make a meaningful comparison between the empirical performance of the algorithms and our theoretical results of Section 5.3. Second, we repeat our analysis in Section 5.4.3 using real data taken from existing crowdsourcing experiments. In this way, we can measure how well the performance of these algorithms translates to a setting where their assumptions are not satisfied. Finally, in Section 5.4.4 we compare the computational requirements of the different inference methods.

5.4.1 List of algorithms and related settings

In this section we list the parameter settings and the implementation details of the inference algorithms we use in our experiments.[3] In order to have a representative sample of the existing literature, we include at least one state-of-the-art algorithm per each category we introduce in Section 2.2. More in detail, voting schemes are represented by majority voting (see Section 2.2.1.1), variational inference by both approximate mean-field and expectation-maximisation (see Section 2.2.2.1), belief propagation by message-passing (see Section 2.2.2.2) and the method of moments by triangular estimation (see Section 2.2.3.3). When possible, we use the parameter settings suggested by the original authors, or we run additional experiments to find the best configuration. All things considered, in our experiments we compare the following algorithms:

- **Majority voting (MAJ).** We use a straightforward implementation of majority voting. Under the US policy, we use the partial sum of votes $\sum_{j \in N_i} x_{ij}$ as an indication of uncertainty.
- **Approximate mean-field inference (AMF).** For experiments on fully-observed data or in conjunction with the UNI policy, we initialise the predictor estimates to their mean prior value $\bar{p} = \alpha/(\alpha + \beta)$ and run 50 iterations of the algorithm to ensure convergence. For adaptive settings in conjunction with the US policy, we run 4 iterations after collecting each new label x_{ij} to update the current estimates. At the end of the collection process we run 50 iterations from scratch. As for MC and SBIC, we use a matching prior $\alpha = 4, \beta = 3$ for synthetic data, a generic prior $\alpha = 2, \beta = 1$ for real-world data, and $q = \frac{1}{2}$ for all experiments.
- **Expectation-maximisation (EM).** This algorithm shares the same implementation as AMF. The only difference is that we use $\bar{\alpha} = \alpha - 1$ and $\bar{\beta} = \beta - 1$ for the predictors' prior. As explained in (Liu et al., 2012), this forces the algorithm to compute the mode rather than the mean of the posterior predictor accuracy.
- **Belief propagation with the spammer-hammer prior (KOS).** We implement the algorithm as a power law iteration with alternating steps $\boldsymbol{w} = X\boldsymbol{z}$ and $\boldsymbol{z} = X\boldsymbol{w}$, where $\boldsymbol{w} = (w_1, \ldots, w_{|N|})$ are the predictor weights and $\boldsymbol{z}$ are the item log-odds. This is the setup recommended by Karger et al. (2014) to achieve maximum performance, as opposed to the more theoretically-sound belief propagation algorithm. We initialise $\boldsymbol{z}$ to its majority voting value, and normalise the result at every iteration to prevent numerical explosion. Under the US policy, we pick the item with log-odds z_i closest to zero. For synthetic data (both under the UNI and US policies), we run only 5 power law iterations before producing the final estimates, as we found this yields better accuracy. For real-data we let the algorithm run for 100 iterations to reach convergence instead.

[3] The source code and related datasets are openly available (see Manino (2020)).

- **Triangular estimation (TE).** We use a straightforward implementation of the algorithm in (Bonald and Combes, 2017). No parameters are required to run this algorithm.

- **Mirror Gibbs particle filter (Mirror Gibbs).** We use a straightforward implementation of the algorithms we present in Section 5.2.1.2. For experiments under the US policy we set the number of particles to $n_{part} = 100$, and the number of rejuvenating flips per particle to $n_{flip} = 100$. For experiments under the UNI policy we set $n_{part} = 20$ and $n_{flip} = 20$, as this is enough to achieve state-of-the-art accuracy and we do not need to track the posterior during the collection phase. In general we found that, given the same value for the product $n_{part}n_{flip}$, the performance of the algorithm is very similar for any choice of n_{part} and n_{flip}. As for AMF and SBIC, we use a matching prior $\alpha = 4, \beta = 3$ for synthetic data, a generic prior $\alpha = 2, \beta = 1$ for real-world data, and $q = \frac{1}{2}$ for all experiments.

- **Streaming Bayesian Inference for Crowdsourcing (SBIC).** We use a straightforward implementation of the algorithms we present in Sections 5.2.2.1 and 5.2.2.2. As for MC and AMF, we use a matching prior $\alpha = 4, \beta = 3$ for synthetic data, a generic prior $\alpha = 2, \beta = 1$ for real-world data, and $q = \frac{1}{2}$ for all experiments.

Note that the TE algorithm requires a dense item-predictor matrix X to function properly. As a result, we exclude it from the experiments in which this assumption is not met, and its predictions degrade to mere random guesses.

5.4.2 Predictive accuracy on synthetic data

We begin our comparison between the different inference algorithms by measuring their performance on synthetic data. This choice allows us to test the algorithms when all the assumptions of the one-coin Dawid-Skene model of Section 5.1 are met. Furthermore, it gives us the opportunity to compare the empirical performance with the theoretical results in Section 5.3.

As in Chapter 3 and 4, we set the number of items to $|M| = 1000$, the item class prior to $q = 1/2$, and we extract the predictors' accuracy according to $p_j \sim \text{Beta}(\alpha, \beta)$. This represents a medium-sized instance, with a fairly varied population of predictors whose average accuracy is larger than $1/2$ but not too distant from it. Thanks to this, the synthetic data we generate contains a large number of incorrect labels, without which it is more difficult to appreciate the difference between the various inference algorithms.

We repeat each test for two values of $L_j \in \{10, 100\}$ for all $j \in N$. The former leaves a larger uncertainty in the individual accuracy of each predictor, as we have fewer data points to estimate it. The latter allows the inference algorithms to compute values of

$\hat{\boldsymbol{p}}$ that are closer to the ground-truth $\boldsymbol{p}$. Note that both these values induce an item-predictor matrix X that is too sparse for the TE algorithm to produce non-random predictions.

For each combination of algorithm, collection policy π, number of labels per predictor L_j and number of predictors per item R we run multiple experiments until we have at least 250 runs with one classification error. In this way the average result has low variance even for very large R. On this note, the error bars we report in all the figures of the present section are computed with the Agresti-Coull method set at 99% confidence (Brown et al., 2001), and their value is as small as 10^{-5} (which makes them barely visible in our plots).

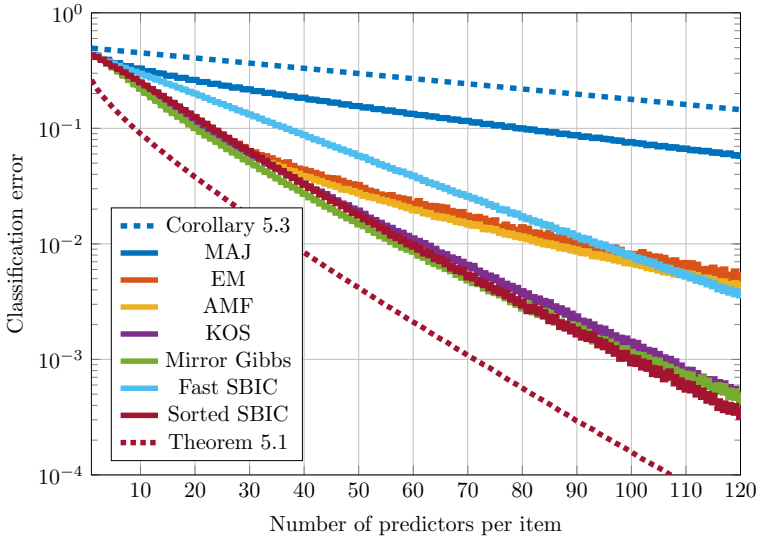
In Figure 5.8 we compare the performance of the inference algorithms under the non-adaptive UNI policy. For reference, we also plot the general lower bound of Theorem 5.1, and the upper bound on majority voting in Corollary 5.3. As suggested by these two bounds, all state-of-the-art algorithms outperform majority voting. Incidentally, this is not the case on real-world data as we show in Section 5.4.3.

Additionally, estimating the exact posterior with the Mirror Gibbs particle filter we present in Section 5.2.1.2 is only one of the most accurate approach. Interestingly, its performance is matched by Sorted SBIC, which is provably optimal from the theoretical standpoint, and the KOS algorithm. Conversely, Fast SBIC tends to suffer from a noticeable performance gap, which is smaller for $L_j = 100$. This is due to the fact that, with a large number of labels per predictor L_j, the incremental estimates of $\bar{p}_j$ do have a chance to converge close to the respective ground-truth p_j.

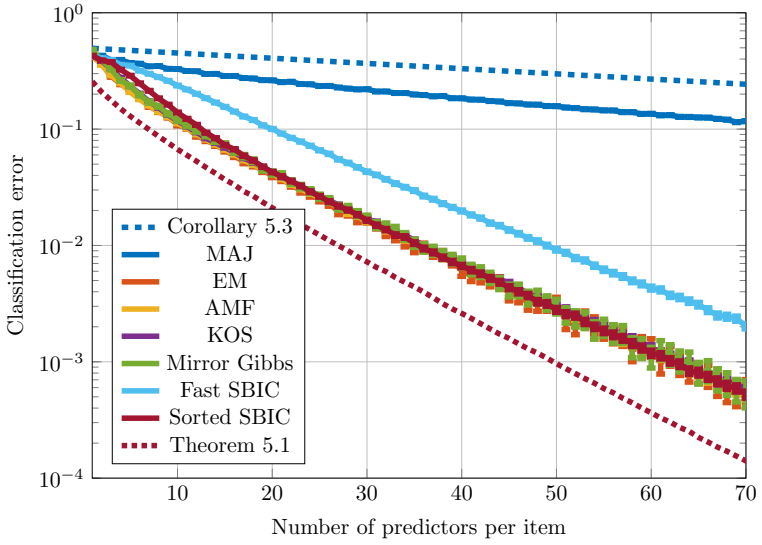
At the same time, the AMF and EM algorithms exhibit a weaker asymptotic behaviour with respect to the other state-of-the-art algorithms when $L_j = 10$. This is because they truly minimise the Kullback-Leibler divergence of their approximation (see Equation 5.15). As a consequence, they are unable to form unbiased estimates of the predictors' accuracy, and end up underestimating the uncertainty in their predictions. Fortunately, this phenomenon disappears for larger values of L_j, where their performance matches the one of the other algorithms.

Similarly, in Figure 5.9 we compare the performance of the inference algorithms under the adaptive US policy. As expected, all algorithms exhibit a smaller prediction error with respect to the UNI policy. However, some of them take advantage of this setting more efficiently. In this regard, only Sorted SBIC and Mirror Gibbs manage to guide the US policy well when $L_j = 10$, and achieve an accuracy that matches the theoretical limit of Theorem 5.1 asymptotically. Conversely, KOS performs better than Fast SBIC for small values of R, but exhibit a weaker asymptotical behaviour when $L_j = 10$.

Finally, the estimation problems in AMF and EM are exacerbated by the interaction with the US policy. With few labels per predictor, the over-optimistic predictions of



(a) $L_j = 10$ labels per predictor.

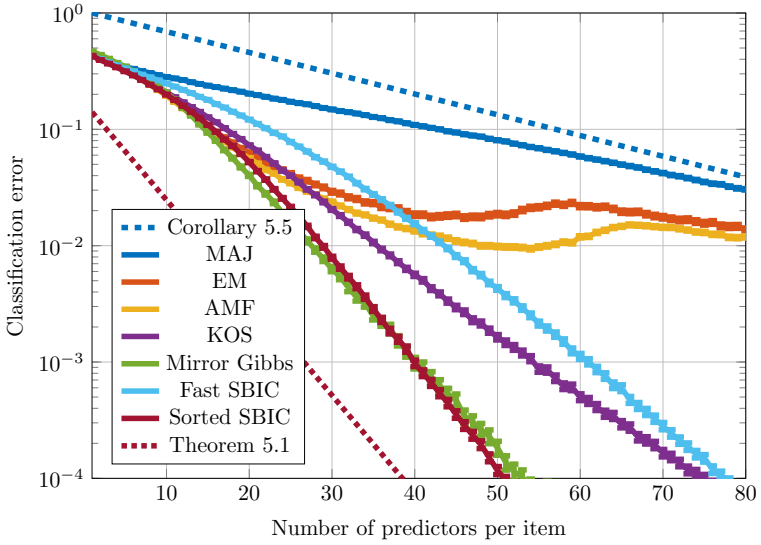


(b) $L_j = 100$ labels per predictor.

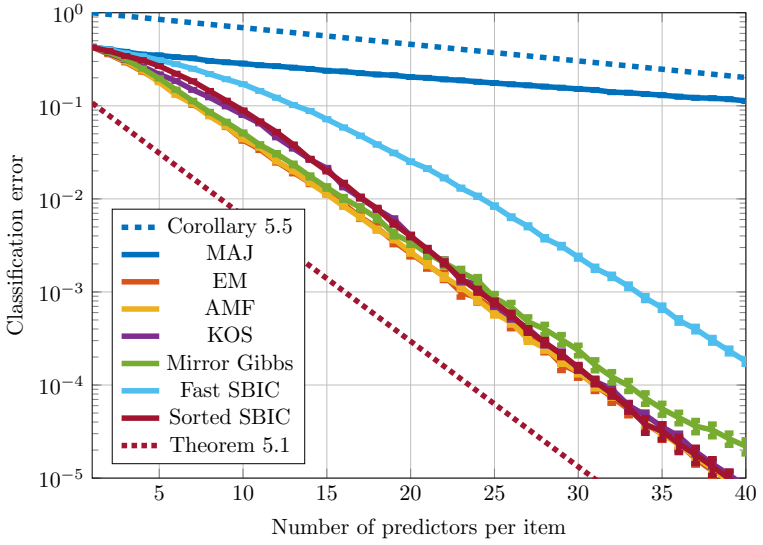
FIGURE 5.8: Prediction error on synthetic data under the UNI policy. For reference, we include our upper bound on majority voting and our general lower bound on probabilistic inference as well.

these two algorithms cause the US policy to collect more labels on the wrong items. As a result, the prediction performance reaches a plateau for $R > 40$. As with the UNI policy, these issues disappear as soon as more labels are available per each predictor.

In summary, all algorithms except from MAJ benefit from a larger number of labels per predictor L_j. On the contrary, all algorithms including MAJ benefit from the use of an adaptive policy like US. We answer the question of computational requirements for this second case in Section 5.4.4. On a different note, this section shows that the general lower bound we present in Theorem 5.1 is not an unattainable asymptote, but actually represents an achievable goal for an inference algorithm.



(a) $L_j = 10$ labels per predictor.



(b) $L_j = 100$ labels per predictor.

FIGURE 5.9: Prediction error on synthetic data under the US policy. For reference, we include our upper bound on majority voting and our general lower bound on probabilistic inference as well.

5.4.3 Predictive accuracy on real datasets

Our experiments in Section 5.4.2 are run under the assumption that the data is produced according to the generative model in Section 5.1. If that is not the case, the predictive accuracy of the different inference algorithms may be affected dramatically. In order to assess what could happen in such a scenario, in the present section we repeat our experiments on real datasets.

Specifically, we consider the 5 publicly available crowdsourcing dataset listed in Table 5.1, which come with binary annotations and ground-truth values (Manino, 2020). These

datasets are often used as a benchmark in the corresponding literature (Snow et al., 2008; Welinder et al., 2010; Lease and Kazai, 2011; Bonald and Combes, 2017). Crucially, they represent a diverse testbed for our inference algorithms. Notably, the density of the label set X ranges from a full matrix for the Birds dataset, to a very sparse one for the TREC dataset. Furthermore, since the data has been crowdsourced, we can expect each crowdworker to exhibit a complex predictive behaviour, rather different from the one-coin Dawid-Skene modelling assumption of a single parameter p_j.

Dataset	# Items	# Predictors	# Labels	Avg. L	Avg. R
Birds	108	39	4212	108	39
Ducks	240	53	9600	181	40
RTE	800	164	8000	49	10
TEMP	462	76	4620	61	10
TREC	711	181	2199	12	3

TABLE 5.1: Summary of the properties of five crowdsourcing datasets including the number of items, number of predictors, total number of data points, average number of predictors per item and average number of items per predictor.

The performance of the inference algorithms is reported in Table 5.2. There we run EM, AFM, Mirror Gibbs and SBIC with the generic prior $\alpha = 2$, $\beta = 1$ and $q = \frac{1}{2}$ as proposed in Liu et al. (2012). Additionally, we include the triangular estimation (TE) algorithm from Bonald and Combes (2017), since it outputs non-random predictions on most of the aforementioned datasets.

Dataset	MAJ	AMF	EM	KOS	MirrorGibbs	TE	FastSBIC	SortedSBIC
Birds	0.241	0.278	0.278	0.278	0.341	**0.194**	0.260	0.298
Ducks	**0.306**	0.412	0.412	0.396	0.412	0.408	0.400	0.405
RTE	0.100	0.075	**0.072**	0.491	0.079	0.257	0.075	**0.072**
TEMP	**0.057**	0.061	0.061	0.567	0.095	0.115	0.059	0.062
TREC	0.257	0.266	**0.217**	0.259	0.302	0.451	0.251	0.239

TABLE 5.2: Prediction error on the real-world datasets

Interestingly, the MAJ algorithm performs quite well and achieves the best score on the Ducks and TEMP datasets. This confirms the practitioner's knowledge that majority voting is a robust and viable algorithms in most crowdsourcing settings. Unsurprisingly, TE achieves its best score on the Birds dataset, which has a full item-predictor matrix X. On the contrary, its predictions are almost random on the TREC dataset, which has a low number of labels per predictor.

At the same time, both Mirror Gibbs and the two variants of SBIC match the performance of the other state-of-the-art Bayesian algorithms (EM, AFM), with Sorted SBIC achieving the best score on RTE. This similarity suggests that the predictive performance of all Bayesian approaches is defined by their common set of assumptions, rather than the specific inference algorithm. More importantly, the fact that Fast SBIC is al-

ways close to the other algorithms makes a strong case for its computationally efficient approach to variational Bayes.

5.4.4 Computational speed

As we show in our experiments of Section 5.4.2, all algorithms benefit from being run under the adaptive US policy. However, in order to deploy such a policy we need to make real-time decisions on how to allocate the predictors on the set of items M, and thus we rely on quick updates by the inference algorithm. In contrast, if we run these algorithms offline after all the data X has been collected (e.g. under the UNI policy), having a small time complexity becomes a less important requirement.

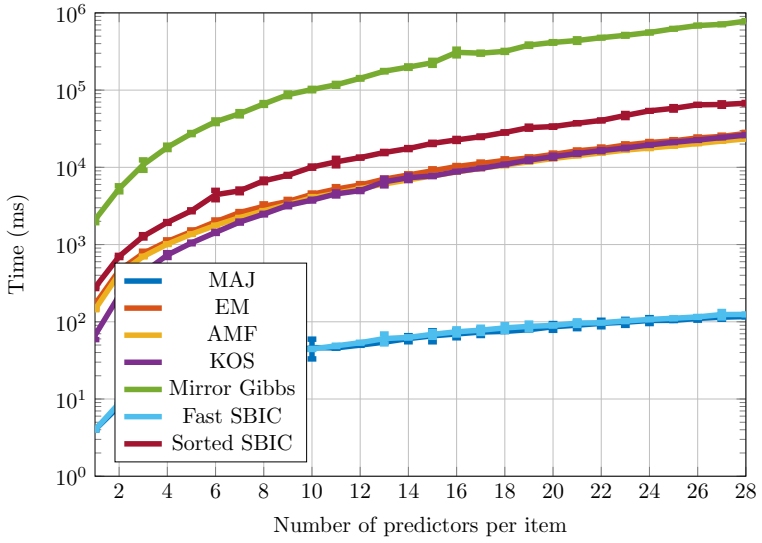
In this section we show that most existing algorithms have too large a computational requirements to do that. We achieve this by measuring the time required by the inference algorithms to produce their predictions under the US policy. For ease of comparison, we choose the same setting as our experiments on the synthetic data of Section 5.4.2. For each algorithm and number of predictors per item R, we run 10 simulations and report the average in Figure 5.10. The error bars correspond to the standard deviation.

Note how Fast SBIC matches MAJ in terms of computational speed, whereas all the other algorithms are orders of magnitude slower. This makes Fast SBIC a valid alternative to MAJ for the online setting, particularly because it can deliver superior predictive accuracy as we demonstrate in Section 5.4.2. Incidentally, Sorted SBIC runs faster in the setting with only $L = 10$ labels per predictor since its time complexity $O(|M|TL)$ depends on L. In theory the same applies to Fast SBIC, but since its time complexity is $O(TL)$ the difference is negligible.

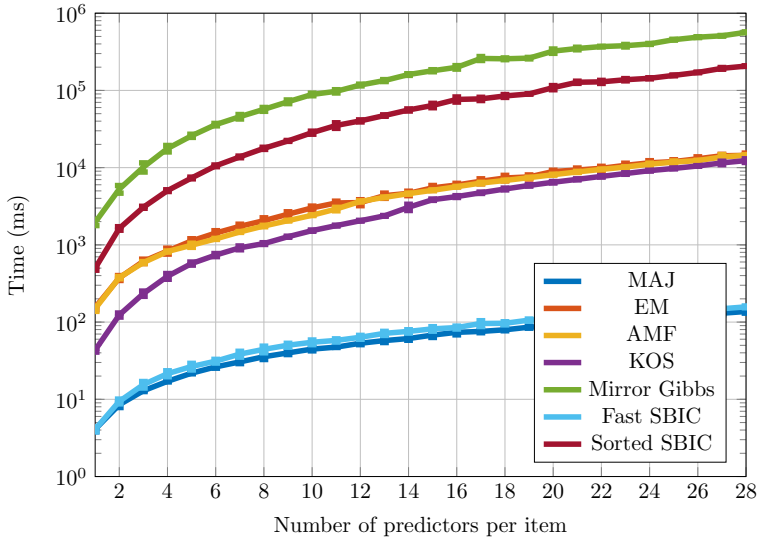
5.5 Summary

In this Chapter, we study the agnostic variant of the Dawid-Skene model, where we have no side information on the accuracy of the predictors. This renews all the challenges we present in Chapter 1. Accordingly, our effort is directed towards solving the aggregation problem in an efficient way and studying the performance of different data collection policies theoretically. Thus, let us highlight what we feel are the three main takeover messages.

First, the Bayesian framework allows us to write a closed-form solution of the inference problem. However, using this method directly is computationally infeasible. We can turn to Monte Carlo methods for estimating the exact posterior, but even our efficient Mirror Gibbs particle filter requires many samples to converge. Thus, we turn to approximate methods, and introduce Streaming Bayesian Inference for Crowdsourcing. This



(a) $L_j = 10$ labels per predictor.



(b) $L_j = 100$ labels per predictor.

FIGURE 5.10: Time required to complete a single inference run with $|M| = 1000$ items under the US policy.

algorithm, in its Fast SBIC variant, is an order or magnitude faster than the state of the art, while retaining good predictive performance.

Second, the Bayesian framework allows us to derive a general bound on the accuracy of any inference algorithm. Our experiments show that this bound is not just of theoretical interest, but is also representative of the empirical performance of state-of-the-art methods. Furthermore, we are able to show that one of our algorithms, Sorted SBIC, matches this bound asymptotically, thus proving its optimality.

Finally, we identify a common weakness of all existing inference algorithms. That is, their performance may degrade substantially once their underlying assumptions are not

met. In particular, we demonstrate this phenomenon on five different crowdsourcing datasets. In some cases, this issue can be mitigated by choosing more conservative priors for Bayesian methods. However, a broader analysis is needed to fully address this issue.

Chapter 6

Conclusions

Combining multiple classifiers to form accurate predictions is a powerful and widespread idea. From the first work on a mathematical justification of democracy (Marquis de Condorcet, 1785), to more modern applications like ensemble methods (Dietterich, 2000) and crowdsourcing (Howe, 2006), the drive to deploy and develop more advanced combinations of classifiers is unrelenting. Doing so requires solving two main problems: how to organise the individual predictors to produce useful data, and how to combine it together to reveal the desired ground-truth.

These data collection and data aggregation problems are partially addressed by the existing literature, as we discuss in Chapter 2. There, we introduce the popular Dawid-Skene model (Dawid and Skene, 1979) as a useful common ground to compare different techniques, and review the existing aggregation and collection algorithms. However, the existing body of knowledge is incomplete: we do not have a theoretical understanding on the difference between data collection policies, we do not have computationally efficient aggregation algorithms that can deliver state-of-the-art preformance in all settings, and we do not have strong guarantees on their accuracy.

Since filling these research gaps is not trivial, our approach is incremental. Thus, in Chapter 3 we begin by reducing the multiple classifier setting to an idealised version of it, where we know the accuracy of each individual predictor exactly. Thanks to this framework, we are able to introduce our random walk interpretation of the collection process, and prove bounds on the exponential tradeoff between the number of predictors and the classification accuracy. Furthermore, we are able to compare different collection policies theoretically for the first time. In this regard, our results include a proof of equivalence between the uncertainty sampling and information gain maximisation policies, and a proof of the superiority of adaptive policies over non-adaptive ones in terms of predictive accuracy.

While the results in Chapter 3 are interesting on their own, most applications of the multiple classifier setting feature predictors of unknown accuracy. In order to cover

these cases, in Chapter 4 we generalise our analysis to a model where the each predictor's accuracy can only be estimated. Fortunately, this step, while highly technical, does not change the nature of our results. Namely, the tradeoff between number of predictors and accuracy is still exponential, the uncertainty sampling and information gain maximisation policies are still equivalent, and all adaptive policies are still superior to non-adaptive ones. However, what changes is the slope of these exponential tradeoffs: the more uncertainty we have on the predictors' accuracy, the slower the decay in error rate is.

But what happens if we have no prior information about the individual predictors? In Chapter 5 we study this case, which we call the agnostic Dawid-Skene model. There, we assume that both the predictors' accuracy and the ground-truth classes of the items have to be inferred from the same dataset. This chicken-or-the-egg problem revives the question of data aggregation, and forces us to introduce two novel aggregation algorithms: Mirror Gibbs and Streaming Bayesian Inference for Crowdsourcing. These algorithms are computationally efficient, while delivering better than state-of-the-art in most settings. On a different note, the presence of a multitude of aggregation algorithms makes a theoretical analysis more challenging. Despite this, we manage to prove a general lower bound on the classification accuracy that greatly improves on the state of the art, and several other bounds on specific algorithms. Notably, we are the first to conduct such analysis for the adaptive uncertainty sampling policy too, as opposed to limiting ourselves to non-adaptive data collection policies.

Overall, in this thesis we make big steps in advancing our knowledge of the multiple classifier setting, and propose several improvements on the state of the art. However, despite these advances, many open questions remains. We outline them in the next Section 6.1, as they constitute exciting directions for future work.

6.1 Future work

In Chapter 1 we listed three main challenges to our research effort on the combination of multiple classifiers: the abundance of different existing models, the lack of theoretical results and the obstacles towards creating efficient collection and aggregation algorithms. In this thesis we lay the foundational work to overcome these challenges, but further steps are needed to broaden the scope of our results. In particular, we envision the following three research directions:

- **Multiclass model.** The natural extension of the binary classification case we study in this thesis is the multiclass Dawid-Skene model. In this scenario, which we briefly review in Section 2.1.2, the ground-truth label y_i of each item is allowed to take values in the interval $[1, k]$. Similarly, each predictor j is parametrised by

a confusion matrix P_j whose entries p^j_{xy} represent the conditional probability of reporting label x given that the ground-truth class of the item is y. Together, these extensions to the one-coin Dawid-Skene model induce some important theoretical differences. First, our random walk interpretation of the collection process takes place in a multidimensional space now. In fact, the marginal posterior probability on the item classes has $k-1$ degrees of freedom in general. For this reason, inferring the ground-truth in the multiclass case may be a harder computational problem, as suggested by Zhang et al. (2016). Second, we lose our equivalence result between the uncertainty sampling and information gain maximisation policies. This is because it may be more profitable to allocate a predictor j on items where she is expected to be more accurate, rather than always choosing the most uncertain task. In this regard, a theoretical analysis of data collection in the multiclass case could explain the empirical observations of Simpson and Roberts (2014).

- **Mismatching priors.** Most of our theoretical results in Chapters 4 and 5 are built on the assumption that the priors q and f_p are known exactly. At the same time, we see that the performance of Bayesian inference algorithms degrades in our experiments on real datasets when this assumption is not met (see Section 5.4.3). Can we extend our theoretical analysis to this case? In general, it is known that exact Bayesian approaches converge to the ground-truth as the size of the dataset increases (Murphy, 2012). However, in our setting we may not have many data points for each predictor. Furthermore, approximate Bayes algorithms like SBIC may suffer even more from a mismatching prior. On this subject, the current literature suggests two different approaches. On the empirical side, we can add an extra layer to our generative model by putting a prior on the prior (Kim and Ghahramani, 2012). Then, the inference algorithm will be able to learn q and f_p. On the theoretical side, we can deal with the generalisation risk of Bayesian approaches by constructing a PAC-Bayes bound (McAllester and Akinbiyi, 2013). Which of these techniques is the most appropriate for the multiple classifier setting is unknown.

- **Accuracy-speed tradeoff.** The two variants of Streaming Bayesian Inference for Crowdsourcing we introduce in Chapter 5 lie at the opposite ends of the efficiency spectrum. Fast SBIC maximises the computational speed, while Sorted SBIC maximises the predictive accuracy. A logical question is whether it is possible to sacrifice part of the computational efficiency of the former to acquire part of the predictive performance of the latter. This *Mixed SBIC* algorithm would be a much needed jack-of-all-trades inference method for the Dawid-Skene model. Furthermore, analysing its theoretical properties would shed some light on the tradeoff between predictive accuracy and computational speed that the multiple classifier setting exhibits.

Apart from this, we also aim to extend the Streaming Bayesian Inference for Crowdsourc-

ing algorithm to other variational inference problems. On the one side, our interested is fuelled by the recent work of Manoel et al. (2017), which proves the convergence of streaming Bayes methods under mild assumptions. On the other side, the streaming framework imposes an implicit regularisation in the number of data points in the training set. This is in contrast with classic regularisation (Murphy, 2012) which adds an explicit regulariser to the objective function. We believe that understanding the connection between the two types of regularisation will be a great source of inspiration for multiple algorithmic contributions.

Appendix A

A guide to the results in Gao et al. 2016

In this appendix, we guide the reader through the results presented in (Gao et al., 2016). Our aim is to show that the bounds in Equations 3.13 and 3.22 are indeed present in the aforementioned paper, even if somewhat clouded by the original notation. In order to make the comparison to our work easier, we employ here the same notation of Section 3.2.1, where we derive our bounds for the same setting. We will occasionally refer to the original notation when needed. Furthermore, we denote all the references to equations and theorems in (Gao et al., 2016) with the superscript (*).

We are interested here in in Corollary 3.1*, since it is built on the same one-coin Dawid-Skene model that we use in Section 3.2.1:

Corollary 3.1*. *Assume* $\max_{j\in N}\big(|\log(p_j)| \vee |\log(1-p_j)|\big) = o\big(RI(\boldsymbol{p})\big)$. *Then, we have:*

$$\mathbb{P}(\hat{y} \neq y) = \exp\big(-\big(1+o(1)\big)RI(\boldsymbol{p})\big) \tag{A.1}$$

where $I(\boldsymbol{p})$ *is defined as:*

$$I(\boldsymbol{p}) = -\frac{1}{R}\sum_{j\in N}\log\big(2\sqrt{p_j(1-p_j)}\big) \tag{A.2}$$

According to the authors, Corollary 3.1* can be proved by adapting the proof of Theorem 3.1* (see (Gao et al., 2016) and its supplementary material). The key is rewriting Equation 15* from its general form to the one coin case:

$$\begin{aligned}\mathbb{P}\left(\bar{s}_j = \frac{1}{2}\log\left(\frac{1-p_j}{p_j}\right)\right) &= p_j\\ \mathbb{P}\left(\bar{s}_j = \frac{1}{2}\log\left(\frac{p_j}{1-p_j}\right)\right) &= 1-p_j\end{aligned} \tag{A.3}$$

Note that $\bar{s}_j$ has the opposite sign of the step $s_j = x_j w_j$ that we use in Section 3.2.1. The authors of (Gao et al., 2016) state that we have a classification error when the following occurs:

$$\mathbb{P}(\hat{y} \neq y) = \mathbb{P}(\bar{h} > 0) \quad \text{where} \quad \bar{h} = \sum_{j \in N} \bar{s}_j \tag{A.4}$$

Now, the probability $\mathbb{P}(\bar{h} > 0)$ can be rewritten as follows (see (Gao et al., 2016) just below Equation 15*):

$$\begin{aligned} \mathbb{P}(\bar{h} > 0) &= \sum_{\bar{h}>0} \prod_{j \in N} \mathbb{P}(\bar{s}_j) \\ &= \sum_{\bar{h}>0} \prod_{j \in N} \frac{\mathbb{P}(\bar{s}_j) \exp(\bar{s}_j) B_{1/2}(p_j, 1-p_j)}{B_{1/2}(p_j, 1-p_j) \exp(\bar{s}_j)} \\ &= \sum_{\bar{h}>0} \exp(-\bar{h}) \prod_{j \in N} \mathbb{Q}_j(\bar{s}j) B_{1/2}(p_j, 1-p_j) \end{aligned} \tag{A.5}$$

where $B_{1/2}(p_j, 1-p_j) = 2\sqrt{p_j(1-p_j)}$ according to the authors' definition at the beginning of their proof of the lower bound part of their Theorem 3.1*, and similarly:

$$\mathbb{Q}_j\left(\bar{s}_j = \frac{1}{2} \log\left(\frac{1-p_j}{p_j}\right)\right) = \frac{\sqrt{p_j}\sqrt{1-p_j}}{B_{1/2}(p_j, 1-p_j)} = \frac{1}{2} \tag{A.6}$$

From here, we can easily derive the authors' upper bound. Note that $\exp(-\bar{h}) < 1$ for all $\bar{h} > 0$, and that the value of $\mathbb{Q}_j(\bar{s}_j)$ does not depend on $\bar{s}_j$. Therefore:

$$\mathbb{P}(\bar{h} > 0) < \prod_{j \in N} B_{1/2}(p_j, 1-p_j) \tag{A.7}$$

which is the result we report in this thesis as Equation 3.13. This result is represented in Corollary 3.1* by the asymptotic rate $\exp\big(-RI(\boldsymbol{p})\big)$ without the $o(1)$ term.

The authors' lower bound requires a little more work. Going back to Equation 15*, we can lower bound $\mathbb{P}(\bar{h} > 0)$ as follows:

$$\begin{aligned} \mathbb{P}(\bar{h} > 0) &> \sum_{0<\bar{h}<H} \exp(-H) \prod_{j \in N} \mathbb{Q}_j(\bar{s}_j) B_{1/2}(p_j, 1-p_j) \\ &= \exp(-H) \mathbb{Q}(0 < \bar{h} < H) \prod_{j \in N} B_{1/2}(p_j, 1-p_j) \end{aligned} \tag{A.8}$$

where $\mathbb{Q}$ is the joint probability of all $\mathbb{Q}_j$ with $j \in N$. If we turn to the supplemental material of (Gao et al., 2016), we find the following bound for $\mathbb{Q}$:

$$\mathbb{Q}(0 < \bar{h} < H) \geq \frac{1}{2} - \mathbb{Q}(\bar{h} \geq H) \geq \frac{1}{2} - \frac{\mathrm{Var}_{\mathbb{Q}}(\bar{h})}{H^2} \geq \frac{1}{4} \tag{A.9}$$

where the second to last inequality is Chebyshev's inequality, and the last one is derived by setting $H = 2\sqrt{\mathrm{Var}_{\mathbb{Q}}(\bar{h})}$. Putting all this pieces together, we can derive the authors'

lower bound:

$$\mathbb{P}(\bar{h} > 0) > \exp\big(-2\sqrt{\mathrm{Var}_{\mathbb{Q}}(\bar{h})}\big)\frac{1}{4}\prod_{j\in N} B_{1/2}(p_j, 1-p_j) \tag{A.10}$$

which is the result we report in the present paper as Equation 3.22. For reference, this result accounts for the presence of the $o(1)$ term in Corollary 3.1*.

Appendix B

Alternative bounds on the UNI policy under the WMV aggregator

In Section 3.2.2 we present matching lower and upper bounds on the classification error of the UNI policy under weighted majority voting. Those results are obtained by careful derivation of the probability of observing the data X, as detailed in Section 3.2.1. In this section we explore a different route, which involves the use of concentration inequalities on the value of the weighted sum $z = \sum_{j \in N} x_j w_j + w_q$. Our early results in (Manino et al., 2018) were derived in this fashion, but they are now superseded by the aforementioned bounds. We explain why in the following discussion.

Concentration inequalities bound the probability that a sum of random variables deviates from its expected value by more than a given margin. In this respect, we can apply them to the weighted sum z and bound the probability of having a positive or negative prediction $\hat{y}$. For simplicity, we assume that $q = \frac{1}{2}$ and thus w_q is zero in the following discussion. Furthermore, we arbitrarily fix the ground-truth label to $y = +1$, since it has no effect on our results given the above assumptions.

With this in mind, we can employ a classic concentration inequality by Hoeffding (1963), and derive the following upper bound:

Theorem B.1. *Given a population of predictors with accuracy $p_j \sim f_p$ such that the corresponding distribution of the steps $s_{ij} = x_{ij} w_j$ is sub-Gaussian with variance proxy $\sigma^2_{f_s}$, a prior on the positive class $q = \frac{1}{2}$, and R predictors per item, the probability of a classification error by the UNI policy under weighted majority voting is upper bounded by:*

$$\mathbb{P}(\hat{y}_i \neq y_i) \leq \exp\Big(- R \frac{\mathbb{E}_{p_j \sim f_p}\big((2p_j - 1)w_j\big)^2}{2\sigma^2_{f_s}} \Big) \tag{B.1}$$

for any item $i \in M$.

Proof. Under the assumption that each step $s_{ij} = x_{ij}w_j$ in the random walk over z_i is sub-Gaussian with variance proxy $\sigma^2_{f_s}$, Hoeffding's concentration inequality states that:

$$\mathbb{P}\big(z_i - \mathbb{E}_{X,\boldsymbol{p}}(z_i) \leq -\epsilon\big) \leq \exp\Big(-\frac{\epsilon^2}{2R\sigma^2_{f_s}}\Big) \tag{B.2}$$

where $R\sigma^2_{f_s}$ is the variance proxy of $z_i = \sum_{j \in N_i} s_j$ since the steps are i.i.d.

Now, the left-hand side of Equation B.2 is related to the probability of a misclassification. In fact, if we substitute $\epsilon = \mathbb{E}_{X,\boldsymbol{p}}(z_i)$ and we notice that $\mathbb{E}_{X,\boldsymbol{p}}(z_i) = R\mathbb{E}_{x_{ij},p_j}(s_{ij})$ we have:

$$\mathbb{P}(z_i \leq 0) \leq \exp\Big(-\frac{R\mathbb{E}_{x_{ij},p_j}(s_{ij})^2}{2\sigma^2_{f_s}}\Big) \tag{B.3}$$

Finally, the expected value of a step s_{ij} depends on the predictor's accuracy p_j alone. Given that by convention the ground-truth is $y_i = +1$, we have that $\mathbb{E}_{x_{ij},p_j}(s_{ij}) = \mathbb{E}_{p_j}\big((2p_j - 1)w_j\big)$, which yields the result in the theorem. □

Hoeffding's concentration inequality is a general result for a sum of random variables. As such, its estimates tend to be quite conservative. A well-known consequence (Hoeffding, 1963) is that for small values of R other concentration inequalities may provide tighter bounds. In our case, this holds true for the Chebyshev-Cantelli inequality (Cantelli, 1928), which translates to the following bound:

Theorem B.2. *Given a population of predictors with accuracy $p_j \sim f_p$ such that the corresponding steps $s_{ij} = x_{ij}w_j$ have variance $\sigma^2_{s_{ij}}$, a prior on the positive class $q = \frac{1}{2}$, and R predictors per item, the probability of a classification error by the UNI policy under weighted majority voting is upper bounded by:*

$$\mathbb{P}(\hat{y}_i \neq y_i) < \left(1 + R\frac{\mathbb{E}_{p_j \sim f_p}\big((2p_j - 1)w_j\big)^2}{\sigma^2_{s_{ij}}}\right)^{-1} \tag{B.4}$$

for any item $i \in M$.

Proof. We can prove this theorem using the same procedure of Theorem B.1. This time we use the Chebyshev-Cantelli concentration inequality (Cantelli, 1928), which states that:

$$\mathbb{P}\big(z_i - \mathbb{E}_{X,\boldsymbol{p}}(z_i) \leq -\epsilon\big) < \frac{\mathrm{Var}_{X,\boldsymbol{p}}(z_i)}{\mathrm{Var}_{X,\boldsymbol{p}}(z_i) + \epsilon^2} \tag{B.5}$$

By substituting $\epsilon = \mathbb{E}_{X,\boldsymbol{p}}(z_i)$ we get the following upper bound on the probability of a classification error:

$$\mathbb{P}(z \leq 0) < \left(1 + \frac{\mathbb{E}_{X,\boldsymbol{p}}(z_i)^2}{\mathrm{Var}_{X,\boldsymbol{p}}(z_i)}\right)^{-1} \tag{B.6}$$

Finally, we know that z_i is the sum of R i.i.d. random variables. Thus, its expected value is $\mathbb{E}_{X,\boldsymbol{p}}(z) = R\mathbb{E}_{x_{ij},p_j}(s_{ij})$ and its variance is $\mathrm{Var}_{X,\boldsymbol{p}}(z) = R\mathrm{Var}_{x_{ij},p_j}(s_{ij})$, which yields the result in the theorem. □

An example of the bounds in Theorems B.1 and B.2 for different values of R is presented in Figure B.1. There, we set the distribution of the predictors' accuracy to $f_p = \mathrm{Uniform}(0.4, 0.8)$. With this choice we represent a mixed population of predictors, with average accuracy 0.6. While similar results can be obtained for different choices of f_p, we must guarantee that the distribution of the steps s_{ij} is sub-Gaussian , otherwise the result in Theorem B.1 becomes invalid. For example, the Beta distribution we use throughout this all thesis does not satisfy this property.

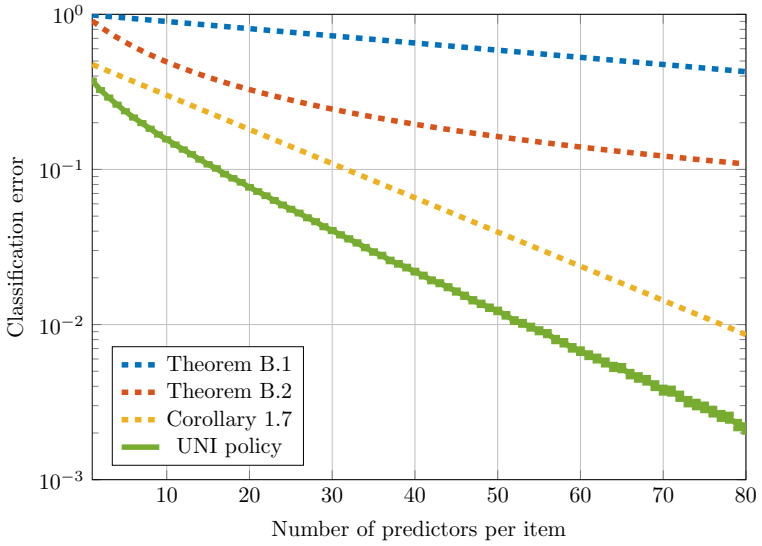


FIGURE B.1: Comparison between the the empirical classification error under the UNI policy and the upper bounds in Theorems B.1, B.2 and Corollary 3.7.

For reference, in Figure B.1 we plot the bound in Corollary 3.7 and the empirical performance of the UNI policy. Note how the two bounds in Theorems B.1 and B.2 are considerably less tight than our main result. Additionally, the bound in Theorem B.1 is far more conservative than the bound in Theorem B.2, despite being asymptotically tighter. This is the result of the aforementioned weakness in Hoeffding's concentration inequality (Hoeffding, 1963).

In summary, using concentration inequalities to study the performance of the UNI policy yields less useful bounds. Moreover, it is impossible to use the same techniques and derive a lower bound. We see a similar issue with the existing bounds on the performance of inference algorithms, as we discuss in Remark 5.4.

Appendix C

The number of steps in a bounded random walk have finite moments

This appendix covers a property that we need in the proofs of Theorems 3.11 and 3.12. There, we rely on the fact that the variance of the number of steps in a bounded random walk is finite. When the steps are i.i.d., like in our case, we can actually prove the stronger result that there always exist a finite exponential moment. What follows is based on a proof published on `math.stackexchange.com` by an anonymous user.[1]

Assume that we have a random walk with i.i.d. steps $s_j \in \mathbb{R}$, such that after t steps we reach the position $z^t = \sum_{j=1}^{t} s_j + z^0$, where z^0 is the starting point. Also, assume that the random walk stops as soon as z^t moves out of some interval (a, b), with finite $a, b \in \mathbb{R}$, and call $n : z^n \notin (a, b) \wedge z^{n-1} \in (a, b)$ the number of steps when this happens. If $s_j \neq 0$ with non-zero probability, n is almost surely finite, and there exists an $m \in \mathbb{N}^+$ such that the following is true:

$$\mathbb{P}(n > m | z^0 = z) \leq e^{-1} \tag{C.1}$$

for any starting point $z \in (a, b)$. However, we also know that the random walk is markovian, that is the final number of steps n only depends on the starting point z. In other terms, if we start the random walk in any z' and then reach z after $k \in \mathbb{N}^+$ steps, we have the following equality:

$$\mathbb{P}(n > m + k | z^k = z) = \mathbb{P}(n > m | z^0 = z) \tag{C.2}$$

Next, we can marginalise the left-hand side of Equation C.2 over all possible values of

[1] math.stackexchange.com/questions/351100/hitting-times-of-markov-chain-process-have-always-finite-moments, retrieved on March 14, 2020.

z, and get the following:

$$\begin{aligned}\mathbb{P}(n > m+k) &= \int_a^b \mathbb{P}(n > m+k, z^k = z)dz \\ &= \int_a^b \mathbb{P}(n > m+k | z^k = z)\mathbb{P}(z^k = z)dz \\ &= \int_a^b \mathbb{P}(n > m | z^0 = z)\mathbb{P}(z^k = z, n > k)dz\end{aligned} \tag{C.3}$$

where the second equality comes from the chain rule, and the third comes from Equation C.2. Note that adding the condition $n > k$ does not change anything, since we already know that we observed k steps to reach $z^k = z$, with $z \in (a, b)$.

Now, we can take advantage of Equation C.1 to bound the value of Equation C.3. We do so as follows:

$$\begin{aligned}\mathbb{P}(n > m+k) &\leq e^{-1} \int_a^b \mathbb{P}(z^k = z, n > k)dz \\ &= e^{-1}\mathbb{P}(n > k)\end{aligned} \tag{C.4}$$

This result proves that the probability of observing n steps before the random walk terminates decreases exponentially as n increases. If we set k to be a multiple of m, we get the following bound:

$$\mathbb{P}(n > cm) \leq \exp(-c) \tag{C.5}$$

for any $c \in \mathbb{N}$. Additionally, if we substitute $k = cm$ we get:

$$\mathbb{P}(n > k) \leq \exp\Big(-\frac{k}{m}\Big) \tag{C.6}$$

which bounds the upper tail of n. As a consequence, there always exists a positive $d < \frac{1}{m}$ such that the exponential moment $\mathbb{E}\big(\exp(dn)\big)$ is finite. A simple corollary is that the second moment $\mathbb{E}(n^2)$, which relates to the variance, is finite too.

Appendix D

Properties of the sigmoid function

In this appendix we list several properties of the sigmoid function which we use throughout this thesis. The sigmoid function is defined as the following monotonically increasing function on $\mathbb{R}$:

$$\text{sig}(z) \equiv \frac{1}{1+\exp(-z)} \qquad \text{for all} \qquad z \in \tag{D.1}$$

The output of the sigmoid function is bounded in $(0, 1)$ as shown in Figure D.1. As a consequence, this function is often used to map a real number to the probability domain, for instance in neural network architectures (Murphy, 2012). In reality, the sigmoid function is not just a generic mapping, but it has a strong probabilistic interpretation. In fact, it is the inverse of the logit function $\log\big(p/(1-p)\big)$ which maps a probability p to its corresponding log-odds z. This is easy to prove, because:

$$\log\Big(\frac{\text{sig}(z)}{1-\text{sig}(z)}\Big) = \log\Big(\frac{1}{\exp(-z)}\Big) = z \tag{D.2}$$

Thus, if we know the log-odds z, we can compute the corresponding probability as $p = \text{sig}(z)$. Moreover, we can exploit the fact that the sigmoid is an odd function to compute the probability of the opposite event as follows:

$$1 - p = 1 - \text{sig}(z) = \text{sig}(-z) \tag{D.3}$$

While these are the basic properties of the sigmoid function, in our discussion in Chapter 3 we use a few more advanced ones. These need some additional definitions (see Section 3.1). First, let us call z the log-odds corresponding to the posterior on an item class $y \in \{\pm 1\}$. In this respect, we can interpret $\text{sig}(z)$ as the posterior probability that $y = +1$. Second, let us assume that an independent predictor is about to provide us with a new label $x \in \{\pm 1\}$, which will be correct with probability p. Under these

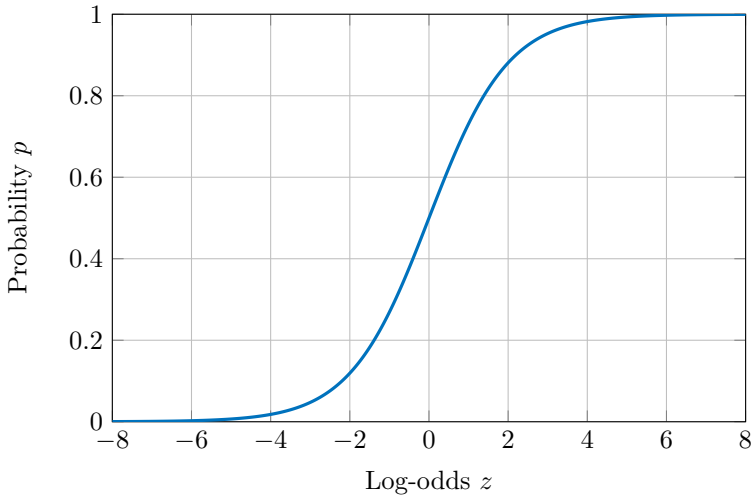


FIGURE D.1: The sigmoid function.

assumptions, we can write the probability of observing a specific x as follows:

$$\begin{aligned}\mathbb{P}(x|z,p) &= \sum_{y\in\{\pm 1\}} \mathbb{P}(y|z)\mathbb{P}(x|y,p) \\ &= \sum_{y\in\{\pm 1\}} \operatorname{sig}(yz)\operatorname{sig}(yxw)\end{aligned} \tag{D.4}$$

where $w = \log\big(p/(1-p)\big)$ are the log-odds of the preditor's accuracy p. Note how we take advantage of the property in Equation D.3 to write a closed form of the joint probability $\mathbb{P}(x, y|z, p) = \mathbb{P}(y|z)\mathbb{P}(x|y, p)$.

Finally, we can extend the result in Equation D.4 to cover a second property, which we make extensive use of in our analysis of the IG policy in Section 3.3.3. This relates the probability of observing a specific new label x with the state of the log-odds z after adding the new contribution xw:

$$\begin{aligned}\mathbb{P}(x|z,p)\operatorname{sig}(z+xw) &= \operatorname{sig}(z)\operatorname{sig}(xw)\Big(1+\frac{\operatorname{sig}(-z)\operatorname{sig}(-xw)}{\operatorname{sig}(z)\operatorname{sig}(xw)}\Big)\operatorname{sig}(z+xw) \\ &= \operatorname{sig}(z)\operatorname{sig}(xw)\Big(1+\exp(-z-xw)\Big)\operatorname{sig}(z+xw) \\ &= \operatorname{sig}(z)\operatorname{sig}(xw)\end{aligned} \tag{D.5}$$

where the first equality comes from the definition of $\mathbb{P}(x|z, p)$ in Equation D.4, the second from the relationship with the log-odds in Equation D.2, and the third from the definition of the sigmoid function in Equation D.1.

Appendix E

Existing bounds on a single item under the plug-in aggregator

The existing literature contains three different upper bounds on the probability of a classification error on a single item with the plug-in aggregator. However, these bounds are qualitatively different from our result in Theorem 4.1, and so a direct comparison is not straightforward. In this appendix, we translate them to our notation and present their properties.

Let us begin with the first bound proposed by Berend and Kontorovich (2015) as Theorem 11*.[1] The corresponding statement reads as follows:

Theorem E.1. *Let* $0 < \delta < 1$ *and* $0 < \epsilon < \min(5, 2\Phi/|N|)$ *where* $\Phi = \sum_{j \in N}(p_j - \frac{1}{2}) \log\big(p_j/(1-p_j)\big)$. *If the following holds:*

$$|O_j| \min(p_j, 1-p_j) \geq 3\left(\frac{\sqrt{4\epsilon+1}-1}{4}\right)^{-2} \log\left(\frac{4|N|}{\delta}\right) \qquad \textit{for all} \qquad j \in N \tag{E.1}$$

then we have:

$$\mathbb{P}\left(\hat{y} \neq y | \boldsymbol{p}, \hat{\boldsymbol{p}}, q = \frac{1}{2}\right) \leq \delta + \exp\left(-\frac{(2\Phi - \epsilon|N|)^2}{8\Phi}\right) \tag{E.2}$$

We can easily spot a major difference with respect to our result in Theorem 4.1: the bound in Equation E.2 contains a extra addend δ. This term can be explained as a measure of how much the estimates $\hat{\boldsymbol{p}}$ are close to their ground-truth values $\boldsymbol{p}$. We can see why in Equation E.1. The more observations $|O_j|$ we have on the trial items, the smaller we can choose δ without breaking the inequality therein. This means smaller values of δ correlate with a more accurate estimate $\hat{p}_j$.

[1] We use the superscript (*) to refer to theorems and equations in other papers.

A second difference is that the bound in Equation E.2 decreases according to Φ. This quantity measures how informative is the set of predictors N, and depends on the ground-truth accuracies $\boldsymbol{p}$. Adding more predictors to the set N guarantees a larger value of Φ, and consequently a stronger bound. However, as we add more and more predictors, Equation E.2 converges to its lowest value δ. Therefore, even with an infinite number of very informative predictors, this bound still predicts a non-zero probability of a classification error.

In contrast our result in Theorem 4.1 depends on the estimates $\hat{\boldsymbol{p}}$ and does not have a constant addend. As a consequence, it remains consistent and informative even with a large number of predictors, or noisy estimates $\hat{\boldsymbol{p}}$. In fact, the latter condition would induce a large value of δ here, making the bound in Equation E.2 meaningless.

Let us move now to the second bound proposed by Berend and Kontorovich (2015) as Theorem 13*. If we translate the original statement to our notation, we have:

Theorem E.2. *Choose any* $\delta \geq \sum_{j \in N} 1/\sqrt{|O_j|}$*, and let* E *be the event that:*

$$\exp\big(-\frac{1}{2}\sum_{j \in N}(\hat{p}_j - \frac{1}{2})\hat{w}_j\big) \leq \frac{\delta}{2} \tag{E.3}$$

then the following holds:

$$\mathbb{P}\big(E \wedge \hat{y} \neq y \big| \boldsymbol{p}, \hat{\boldsymbol{p}}, q = \frac{1}{2}\big) \leq \delta \tag{E.4}$$

Here, the quantity δ is again linked to the accuracy of the estimates $\hat{\boldsymbol{p}}$. In fact, when we can test each predictor $j \in N$ on a larger number of trials, the corresponding contribution $1/\sqrt{|O_j|}$ will be smaller. In contrast with the previous result in Theorem E.1, δ alone represent the upper bound on the probability of an error. The introduction of the event E in Equation E.3 is meant to ensure that the estimated version of Φ provides enough informative power.

Unfortunately, the bound in Theorem E.2 is not consistent in the number of predictors $|N|$. That is, the more predictors we have, the worse the bound becomes. To see why is it so, let us perform the following thought experiment. First, note that when event E fails, we are left without a bound. Thus, let us build the best-case scenario and assume that event E does always happen. Under this condition, we can minimise the upper bound in Equation E.4 by choosing $\delta = \sum_{j \in N} 1/\sqrt{|O_j|}$. Now, if we add one more predictor j^* to our current set N, the value of δ must increase by $1/\sqrt{|O_{j^*}|}$. Accordingly, the bound in Equation E.4 becomes less strong by the same amount.

Conversely, our result in Theorem 4.1 is consistent in the number of predictors $|N|$. That is, each new predictor makes our upper bound closer to zero. The only exception being when the estimate $\hat{p}_j$ is wildly different from its ground-truth value p_j.

Finally, let us consider the bound proposed by Gao et al. (2016) as Theorem 4.1*.

Similarly to the known accuracy case (see Appendix A), the results in this paper are somewhat obfuscated by the authors' notation. Thus, we present here a digested version of their bound, using our notation:

Theorem E.3. *Assume that, as the number of trials increases, i.e.* $|G| \to \infty$*, we have:*

$$\mathbb{P}\Big(\sum_{j \in N} \max\Big(|\log(\hat{p}_j) - \log(p_j)|, |\log(1-\hat{p}_j) - \log(1-p_j)| \Big) > \delta \Big) \to 0 \tag{E.5}$$

with $\delta = o\big(I(\boldsymbol{p})\big)$*, where* $I(\boldsymbol{p}) = \frac{1}{2} \sum_{j \in N} \log\big(4p_j(1-p_j)\big)$*. Then, we have:*

$$\mathbb{P}\Big(\hat{y} \neq y | \boldsymbol{p}, \hat{\boldsymbol{p}}, q = \frac{1}{2}\Big) \leq \exp\Big(-\big(1 + o(1)\big) I(\boldsymbol{p}) \Big) \tag{E.6}$$

At first glance, Equation E.6 seem to state that the probability of an error is asymptotically identical to the known accuracy case of Remark 3.2. In fact the quantity $\exp\big(I(\boldsymbol{p})\big)$ is nothing but the term $\prod_{j \in N} 2\sqrt{p_j(1-p_j)}$ therein. However, let us not forget that the condition imposed in Equation E.5 guarantees that the estimates $\hat{\boldsymbol{p}}$ are close to their ground-truth value $\boldsymbol{p}$. Under this light, the result in Theorem E.3 is not surprising: it merely states that the performance with accurate estimates $\hat{\boldsymbol{p}}$ is close to having the actual ground-truth $\boldsymbol{p}$ in place.

At the same time, when the condition in Equation E.5 is not met, the $o(1)$ term in Equation E.6 is not negligible anymore, and we are left without a bound. In contrast, our result in Theorem 4.1 holds for estimates $\hat{\boldsymbol{p}}$ of any kind, as long as they do not reach the extremes of the interval $(0, 1)$.

In summary, all three bounds in Theorems E.1, E.2 and E.3 are derived under some restrictive assumption. Because of that, they either exhibit some unappealing properties (non-convergence, non-consistency), or do not cover the general case (noisy estimates of $\boldsymbol{p}$). This fact, makes them qualitatively inferior to our bound in Theorem 4.1.

Appendix F

The Beta distribution as a prior on the predictors' accuracy

Throughout our discussion in Chapters 3 and 4 we assume to have some knowledge on the underlying distribution of the predictors' accuracy f_p. Still, except for some specific examples, we keep our results general and do not restrict f_p to any specific distribution. In this appendix we do the opposite, and show what happens when we constrain f_p to be a Beta distribution.

The reason for this choice is twofold. First, some of the theoretical results we present in Chapter 4 can be expressed in closed form when f_p is set to be a Beta distribution. Second, the Beta distribution is the conjugate prior of Bernoulli random variables. This makes it the most natural distribution to model the distribution of the predictors' accuracy p_j. In fact, such choice is standard in Bayesian modelling (Murphy, 2012) as it often yields an analytical form to otherwise intractable expressions.

As an example, let us take the upper bound on the performance of the UNI policy for the known accuracy case in Equation 3.27. If we assume that $f_p = \text{Beta}(\alpha, \beta)$ for any choice of the parameters $\alpha, \beta > 0$, the expectation over p_j at its core becomes:

$$\begin{aligned}\mathbb{E}_{p_j \sim f_p}\big(\sqrt{p_j(1-p_j)}\big) &\equiv \int_0^1 \sqrt{p_j(1-p_j)}\,\mathbb{P}(p_j|f_p)dp_j \\ &= \int_0^1 \sqrt{p_j(1-p_j)}\Big(\frac{p_j^{\alpha-1}(1-p_j)^{\beta-1}}{B(\alpha,\beta)}\Big)dp_j \\ &= \frac{B\big(\alpha+\frac{1}{2},\beta+\frac{1}{2}\big)}{B(\alpha,\beta)}\end{aligned} \tag{F.1}$$

which is a closed-form expression based on the beta function $B(a,b) \equiv \int_0^1 u^{a-1}(1-u)^{b-1}du$.

While the result in Equation F.1 gives us a taste of the analytical properties of the Beta

prior, its role really comes to the fore when we apply it to our bounds for the unknown accuracy case in Chapter 4. There, we assume that each predictor j is tested on a set of trial items G with known ground-truth $\boldsymbol{g}$. If we assume that $f_p = \text{Beta}(\alpha, \beta)$ we can write the probability of observing the outcomes O_j on G in closed form. Let us define the corresponding number of successes $v_j \equiv \sum_{k \in G} \mathbb{I}(o_{kj} = g_k)$. Then, we can write the following equivalence:

$$\mathbb{P}(O_j | f_p, \boldsymbol{g}) = \frac{B(v_j + \alpha, |G| - v_j + \beta)}{B(\alpha, \beta)} \tag{F.2}$$

In this regard, the quantities $v_j + \alpha$ and $|G| - v_j + \beta$ are often referred to as the *pseudo-counts* (Murphy, 2012), as they measure the (adjusted) number of successes and failures respectively. When computing a value for the estimate $\hat{p}_j$, these pseudo-counts are the only variables that matter. In fact, under our assumptions the expected value of the posterior on the ground-truth accuracy p_j is $\hat{p}_j = (v_j + \alpha)/(|G| + \alpha + \beta)$. As a result, we are often interested in the probability of observing a specific number of successes v_j. Since v_j depends solely on O_j, we can extend Equation F.2 and derive the following expression:

$$\mathbb{P}(v_j | f_p, |G|) = \binom{|G|}{v_j} \frac{B(v_j + \alpha, |G| - v_j + \beta)}{B(\alpha, \beta)} \tag{F.3}$$

where the binomial coefficient takes into account all the possible permutations of the successes v_j over the vector O_j.

Thanks to Equation F.3 we can derive the closed form of the expectations in the bounds of Corollaries 4.4, 4.5, 4.8, 4.9 and Theorem 4.7. In order of appearance, we have:

$$\mathbb{E}_{\hat{p}_j | f_p, \boldsymbol{g}}\big(\sqrt{\hat{p}_j(1 - \hat{p}_j)}\big) = \sum_{v_j=0}^{|G|} \mathbb{P}\big(v_j | f_p, |G|\big) \left(\frac{\sqrt{(v_j + \alpha)(|G| - v_j + \beta)}}{|G| + \alpha + \beta} \right) \tag{F.4}$$

$$\mathbb{E}_{\hat{p}_j | f_p, \boldsymbol{g}}\Big(\log\big(4\hat{p}_j(1 - \hat{p}_j)\big) \Big) = \sum_{v_j=0}^{|G|} \mathbb{P}\big(v_j | f_p, |G|\big) \log \left(\frac{4(v_j + \alpha)(|G| - v_j + \beta)}{(|G| + \alpha + \beta)^2} \right) \tag{F.5}$$

$$\mathbb{E}_{\hat{p}_j | f_p, \boldsymbol{g}}\big(|\hat{w}_j|\big) = \sum_{v_j=0}^{|G|} \mathbb{P}\big(v_j | f_p, |G|\big) \left| \log \left(\frac{v_j + \alpha}{|G| - v_j + \beta} \right) \right| \tag{F.6}$$

$$\begin{aligned} \mathbb{E}_{\hat{p}_j | f_p, \boldsymbol{g}}\big((2\hat{p}_j - 1)\hat{w}_j\big) = \sum_{v_j=0}^{|G|} & \mathbb{P}\big(v_j | f_p, |G|\big) \\ & \left(\frac{2v_j - |G| + \alpha - \beta}{|G| + \alpha + \beta} \right) \log \left(\frac{v_j + \alpha}{|G| - v_j + \beta} \right) \end{aligned} \tag{F.7}$$

Incidentally, the results in Equations F.3, F.4, F.5 and F.6 can be used to compute the value of the constants $\hat{c}_{uni}$, $\hat{c}_{nada}$ and $\hat{c}_{ada}$ we introduce in Section 4.4. For more examples of the use of the Beta distribution as a prior, see our discussion on the SBIC algorithm in Chapter 5.

Bibliography

Russell L. Ackoff. *A Concept of Corporate Planning*. Wiley-Interscience, 1969.

Susanne Albers. Better Bounds for Online Scheduling. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, pages 130–139, 1997.

Anima Anandkumar, Dean P. Foster, Daniel Hsu, Sham M. Kakade, and Yi-Kai Liu. A Spectral Algorithm for Latent Dirichlet Allocation. *Algorithmica*, 72(1):193–214, 2015.

Arash Asadpour and Amin Saberi. An Approximation Algorithm for Max-Min Fair Allocation of Indivisible Goods. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, pages 114–121, 2007.

A. Avidor, J. Sgall, and Y. Azar. Ancient and New Algorithms for Load Balancing in the L_p Norm. *Algorithmica*, 29:422–441, 2001.

Eyal Baharad, Jacob Goldberger, Moshe Koppel, and Shmuel Nitzan. Distilling the Wisdom of Crowds: Weighted Aggregation of Decisions on Multiple Issues. *Autonomous Agents and Multi-Agent Systems*, 22(1):31–42, 2011.

Daniel W. Barowy, Charlie Curtsinger, Emery D. Berger, and Andrew McGregor. AutoMan: A Platform for Integrating Human-based and Digital Computation. In *Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications*, pages 639–654, 2012.

Daniel Berend and Aryeh Kontorovich. A Finite Sample Analysis of the Naive Bayes Classifier. *Journal of Machine Learning Research*, 16(1):1519–1545, 2015.

Daniel Berend and Jacob Paroush. When is Condorcet's Jury Theorem valid? *Social Choice and Welfare*, 15(4):481–488, 1998.

Thomas Bonald and Richard Combes. A Minimax Optimal Algorithm for Crowdsourcing. In *Proceedings of the Thirtieth International Conference on Neural Information Processing Systems*, pages 4355–4363, 2017.

Leo Breiman. Bagging Predictors. *Machine Learning*, 24(3):123–140, 1996.

Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C Wilson, and Michael I Jordan. Streaming Variational Bayes. In *Proceedings of the Twenty-Sixth International Conference on Neural Information Processing Systems*, pages 1727–1735, 2013.

Lawrence D. Brown, T. Tony Cai, and Anirban DasGupta. Interval Estimation for a Binomial Proportion. *Statistical Science*, 16(2):101–133, 2001.

Michael Buhrmester, Tracy Kwang, and Samuel D. Gosling. Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data? *Perspectives on Psychological Science*, 6(1):3–5, 2011.

Francesco P. Cantelli. Sui Confini della Probabilità. *Atti del Congresso Internazionale dei Matematici*, 6(1):47—-59, 1928.

Olivier Cappé, Simon J. Godsill, and Eric Moulines. An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. *Proceedings of the IEEE*, 95(5): 899–924, 2007.

Xi Chen, Qihang Lin, and Dengyong Zhou. Optimistic Knowledge Gradient Policy for Optimal Budget Allocation in Crowdsourcing. In *Proceedings of the Thirtieth International Conference on Machine Learning*, pages 64–72, 2013.

Nicolas Chopin. A Sequential Particle Filter Method for Static Models. *Biometrika*, 89 (3):539–552, 2002.

Peter J. Coughlin. *Probabilistic Voting Theory*. Cambridge University Press, 1992.

Peng Dai, Mausam, and Daniel S. Weld. Artificial Intelligence for Artificial Artificial Intelligence. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 1153–1159, 2011.

Nilesh Dalvi, Anirban Dasgupta, Ravi Kumar, and Vibhor Rastogi. Aggregating Crowdsourced Binary Ratings. In *Proceedings of the 22Nd International Conference on World Wide Web*, pages 285–294, 2013.

Sanjoy Dasgupta. Analysis of a Greedy Active Learning Strategy. In *Proceedings of the Seventeenth International Conference on Neural Information Processing Systems*, pages 337–344, 2004.

Alexander P. Dawid and Allan M. Skene. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, 1979.

Thomas G. Dietterich. Ensemble Methods in Machine Learning. In *Proceedings of the International Workshop on Multiple Classifier Systems*, pages 1–15, 2000.

Laura Dietz. Directed Factor Graph Notation for Generative Models. Technical report, Max Planck Institute for Informatics, Saarbrücken, Germany, 2010.

Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G. Ipeirotis, and Philippe Cudré-Mauroux. The Dynamics of Micro-Task Crowdsourcing: The Case of Amazon MTurk. In *Proceedings of the 24th International Conference on World Wide Web*, pages 238–247, 2015.

Pinar Donmez, Jaime Carbonell, and Jeff Schneider. A Probabilistic Framework to Learn from Multiple Annotators with Time-Varying Accuracy. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 826–837, 2010.

Thomas R Fanshawe, Andrew G Lynch, Ian O Ellis, Andrew R Green, and Rudolf Hanka. Assessing Agreement between Multiple Raters with Missing Rating Information, Applied to Breast Cancer Tumour Grading. *PLoS One*, 3(8), 2008.

Siamak Faridani, Björn Hartmann, and Panagiotis G Ipeirotis. What's the Right Price? Pricing Tasks for Finishing on Time. In *Proceedings of the Eleventh AAAI Conference on Human Computation*, pages 26–31, 2011.

Yoav Freund and Robert E. Schapire. Experiments with a New Boosting Algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, pages 148—-156, 1996.

Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective Sampling Using the Query by Committee Algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

Yifan Fu, Xingquan Zhu, and Bin Li. A Survey on Instance Selection for Active Learning. *Knowledge and Information Systems*, 35(2):249–283, 2013.

Chao Gao, Yu Lu, and Dengyong Zhou. Exact Exponent in Optimal Rates for Crowdsourcing. In *Proceedings of the Thirty-Third International Conference on Machine Learning*, pages 603–611, 2016.

Amirata Ghorbani and James Zou. Data Shapley: Equitable Valuation of Data for Machine Learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2242–2251, 2019.

Arpita Ghosh, Satyen Kale, and Preston McAfee. Who moderates the moderators? crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, pages 167—-176, 2011.

Daniel Golovin. Max-Min Fair Allocation of Indivisible Goods. Technical report, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2005.

Ronald L. Graham. Bounds for Certain Multiprocessing Anomalies. *Bell System Technical Journal*, 45(9):1563–1581, 1966.

Chien-Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. Adaptive Task Assignment for Crowdsourced Classification. In *Proceedings of the Thirtieth International Conference on Machine Learning*, pages 534–542, 2013.

Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

John J. Horton and Lydia B. Chilton. The Labor Economics of Paid Crowdsourcing. In *Proceedings of the Eleventh ACM Conference on Electronic Commerce*, pages 209–218, 2010.

Jeff Howe. The Rise of Crowdsourcing. *Wired Magazine*, 14(06):1–5, 2006.

Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality Management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 64–67, 2010.

Hyun Joon Jung and Matthew Lease. Improving Quality of Crowdsourced Labels via Probabilistic Matrix Factorization. In *Human Computation Workshop at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

Ece Kamar, Severin Hacker, and Eric Horvitz. Combining Human and Machine Intelligence in Large-Scale Crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 467–474, 2012.

David R. Karger, Sewoong Oh, and Devavrat Shah. Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems. *Operations Research*, 62(1):1–24, 2014.

Narenda Karmarkar and Richard M. Karp. The Differencing Method of Set Partitioning. Technical report, University of California, Berkeley, 1983.

Ashish Khetan and Sewoong Oh. Achieving Budget-Optimality with Adaptive Schemes in Crowdsourcing. In *Proceedings of the Twenty-Ninth International Conference on Neural Information Processing Systems*, pages 4844–4852, 2016.

Ashiqur R KhudaBukhsh, Jaime G Carbonell, and Peter J Jansen. Detecting Non-Adversarial Collusion in Crowdsourcing. In *Proceeding of the Second AAAI Conference on Human Computation and Crowdsourcing*, pages 104–111, 2014.

Hyun-Chul Kim and Zoubin Ghahramani. Bayesian Classifier Combination. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, pages 619–627, 2012.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

Richard E. Korf. A Complete Anytime Algorithm for Number Partitioning. *Artificial Intelligence*, 106(2):181–203, 1998.

Solomon Kullback and Richard A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Walter S. Lasecki, Christopher D. Miller, and Jeffrey P. Bigham. Warping Time for More Effective Real-time Crowdsourcing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2033–2036, 2013.

Matthew Lease and Gabriella Kazai. Overview of the TREC 2011 crowdsourcing track. In *Proceedings of TREC 2011*, 2011.

David D Lewis and William A Gale. A Sequential Algorithm for Training Text Classifiers. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

Christopher H Lin, Mausam, and Daniel S Weld. To Re (label), or Not To Re (label). In *Proceeding of the Second AAAI Conference on Human Computation and Crowdsourcing*, pages 151–158, 2014.

Chris Lintott, Kevin Schawinski, Steven Bamford, Anže Slosar, Kate Land, Daniel Thomas, Edd Edmondson, Karen Masters, Robert C Nichol, M Jordan Raddick, Alex Szalay, Dan Andreescu, Phil Murray, and Jan Vandenberg. Galaxy Zoo 1: data release of morphological classifications for nearly 900 000 galaxies. *Monthly Notices of the Royal Astronomical Society*, 410(1):166–178, 2011.

Nick Littlestone and Manfred K. Warmuth. The Weighted Majority Algorithm. *Information and Computation*, 108(2):212–261, 1994.

Qiang Liu, Jian Peng, and Alexander Ihler. Variational Inference for Crowdsourcing. In *Proceedings of the Twenty-Fifth International Conference on Neural Information Processing Systems*, pages 692–700, 2012.

Daniel J. Lizotte, Omid Madani, and Russell Greiner. Budgeted Learning of Naive-Bayes Classifiers. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, pages 378–385, 2003.

Nuno Luz, Nuno Silva, and Paulo Novais. A Survey of Task-Oriented Crowdsourcing. *Artificial Intelligence Review*, 44(2):187–213, 2015.

David J. C. MacKay. Information-Based Objective Functions for Active Data Selection. *Neural Computation*, 4(4):590–604, 1992.

Edoardo Manino. Source Code of `binary_sims.exe` and Related Datasets. *https://doi.org/10.5258/SOTON/D1505*, 2020.

Edoardo Manino, Long Tran-Thanh, and Nicholas R. Jennings. On the Efficiency of Data Collection for Crowdsourced Classification. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 1568–1575, 2018.

Edoardo Manino, Long Tran-Thanh, and Nicholas R. Jennings. On the Efficiency of Data Collection for Multiple Naïve Bayes Classifiers. *Artificial Intelligence*, 275(1): 356–378, 2019a.

Edoardo Manino, Long Tran-Thanh, and Nicholas R. Jennings. Streaming Bayesian Inference for Crowdsourced Classification. In *Proceedings of the Thirty-Third International Conference on Neural Information Processing Systems*, pages 12762–12772, 2019b.

Andre Manoel, Florent Krzakala, Eric W. Tramel, and Lenka Zdeborova. Streaming Bayesian Inference: Theoretical Limits and Mini-Batch Approximate Message-Passing. In *Proceedings of the Fifty-Fifth Annual Allerton Conference on Communication, Control, and Computing*, pages 1048–1055, 2017.

Andrew Mao, Ece Kamar, and Eric Horvitz. Why Stop Now? Predicting Worker Engagement in Online Crowdsourcing. In *Proceeding of the First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

Marquis de Condorcet. *Essai sur l'Application de l'Analyse à la Probabilité des Décisions Rendus à la Pluralité des Voix*. Imprimerie Royale, Paris, 1785.

Winter Mason and Duncan J. Watts. Financial Incentives and the "Performance of Crowds". In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 77–85, 2009.

David McAllester and Takintayo Akinbiyi. PAC-Bayesian Theory. In *"Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pages 95–103. Springer Berlin Heidelberg, 2013.

Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

Victor Naroditskiy, Nicholas R. Jennings, Pascal Van Hentenryck, and Manuel Cebrian. Crowdsourcing Contest Dilemma. *Journal of the Royal Society Interface*, 11(99):1–8, 2014.

An T Nguyen, Byron C Wallace, and Matthew Lease. Combining Crowd and Expert Labels using Decision Theoretic Active Learning. In *Proceedings of the Third AAAI Conference on Human Computation*, pages 120–129, 2015.

Shmuel Nitzan and Jacob Paroush. Optimal Decision Rules in Uncertain Dichotomous Choice Situations. *International Economic Review*, 23(2):289–297, 1982.

Besmira Nushi, Adish Singla, Andreas Krause, and Donald Kossmann. Learning and Feature Selection under Budget Constraints in Crowdsourcing. In *Proceedings of the Fourth AAAI Conference on Human Computation and Crowdsourcing*, pages 159–168, 2016.

David Oleson, Alexander Sorokin, Greg Laughlin, Vaughn Hester, John Le, and Lukas Biewald. Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing. In *Proceedings of the 11th AAAI Conference on Human Computation*, pages 43–48, 2011.

Aditya Parameswaran, Hector Garcia-Molina, Hyunjung Park, Neoklis Polyzotis, Aditya Ramesh, and Jennifer Widom. CrowdScreen: Algorithms for Filtering Data with Humans. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 361–372, 2012.

Joel Ross, Lilly Irani, M Six Silberman, Andrew Zaldivar, and Bill Tomlinson. Who Are the Crowdworkers?: Shifting Demographics in Mechanical Turk. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, pages 2863–2872, 2010.

Burr Settles. Active Learning Literature Survey. *Machine Learning*, 15(2):201–221, 2010.

Edwin Simpson and Stephen Roberts. Bayesian Methods for Intelligent Task Assignment in Crowdsourcing Systems. In *Scalable Decision Making: Uncertainty, Imperfection, Deliberation*, pages 1–32. Springer, 2014.

Edwin Simpson, Stephen Roberts, Ioannis Psorakis, and Arfon Smith. Dynamic Bayesian Combination of Multiple Imperfect Classifiers. In *Decision Making and Imperfection*, pages 1–35. Springer, 2013.

Edwin Simpson, Stephen J Roberts, Arfon Smith, and Chris Lintott. Bayesian Combination of Multiple, Imperfect Classifiers. In *Proceedings of the Twenty-Fourth International Conference on Neural Information Processing Systems*, pages 1–8, 2011.

Adish Singla and Andreas Krause. Truthful Incentives in Crowdsourcing Tasks Using Regret Minimization Mechanisms. In *Proceedings of the Twenty-Second International Conference on World Wide Web*, pages 1167–1178, 2013.

Aleksandrs Slivkins and Jennifer Wortman Vaughan. Online Decision Making in Crowdsourcing Markets: Theoretical Challenges. *SIGecom Exchanges*, 12(2):4–23, 2014.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and Fast – but is It Good?: Evaluating Non-expert Annotations for Natural Language Tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263, 2008.

Alexander Sorokin and David Forsyth. Utility Data Annotation with Amazon Mechanical Turk. In *Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Workshops*, pages 1–8, 2008.

Kai Ming Ting and Ian H. Witten. Stacking Bagged and Dagged Models. In *Proceedings of the Fourteenth international Conference on Machine Learning*, pages 367–375, 1997.

Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R. Jennings. Efficient Crowdsourcing of Unknown Experts Using Bounded Multi-Armed Bandits. *Artificial Intelligence*, 214(1):89–111, 2014.

Luis von Ahn and Laura Dabbish. Labeling Images with a Computer Game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 319–326, 2004.

Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently Scaling Up Crowdsourced Video Annotation. *International Journal of Computer Vision*, 101(1):184–204, 2013.

Abraham Wald. On Cumulative Sums of Random Variables. *The Annals of Mathematical Statistics*, 15(3):283–296, 1944.

Dongxia Wang, Tim Muller, Jie Zhang, and Yang Liu. Quantifying Robustness of Trust Systems Against Collusive Unfair Rating Attacks Using Information Theory. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 111–117, 2015.

Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The Multidimensional Wisdom of Crowds. In *Proceedings of the Twenty-Third International Conference on Neural Information Processing Systems*, pages 1–9, 2010.

Peter Welinder and Pietro Perona. Online Crowdsourcing: Rating Annotators and Obtaining Cost-Effective Labels. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Workshops*, pages 25–32, 2010.

Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *Proceedings of the Twenty-Second International Conference on Neural Information Processing Systems*, pages 2035–2043, 2009.

Kyle W. Willett, Chris J. Lintott, Steven P. Bamford, Karen L. Masters, Brooke D. Simmons, Kevin R. V. Casteels, Edward M. Edmondson, Lucy F. Fortson, Sugata Kaviraj, William C. Keel, Thomas Melvin, Robert C. Nichol, M. Jordan Raddick, Kevin Schawinski, Robert J. Simpson, Ramin A. Skibba, Arfon M. Smith, and Daniel Thomas. Galaxy Zoo 2: Detailed Morphological Classifications for 304 122 Galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, 435(4):2835–2860, 2013.

David H Wolpert. Stacked Generalization. *Neural Networks*, 5(2):241–259, 1992.

Lei Xu, Adam Krzyżak, and Ching Y Suen. Methods of Combining Multiple Classifiers and their Applications to Handwriting Recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):418–435, 1992.

Yan Yan, Romer Rosales, Glenn Fung, and Jennifer G. Dy. Active Learning from Crowds. In *Proceedings of the Twenty-Eighth International Conference on Machine Learning*, pages 1161–1168, 2011.

Hao Zhang, Yao Ma, and Masashi Sugiyama. Bandit-based Task Assignment for Heterogeneous Crowdsourcing. *Neural Computation*, 27(11):2447–2475, 2015.

Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I. Jordan. Spectral Methods Meet EM: A Provably Optimal Algorithm for Crowdsourcing. *Journal of Machine Learning Research*, 17(1):3537–3580, 2016.

Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5):541—-552, 2017.

Dengyong Zhou, John C. Platt, Sumit Basu, and Yi Mao. Learning from the Wisdom of Crowds by Minimax Entropy. In *Proceedings of the Twenty-Fifth International Conference on Neural Information Processing Systems*, pages 2195–2203, 2012.