UNIVERSITY OF Southampton

University of Southampton Research Repository
ePrints Soton

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g.

AUTHOR (year of submission) "Full thesis title", University of Southampton, name of the University School or Department, PhD Thesis, pagination

http://eprints.soton.ac.uk

UNIVERSITY OF SOUTHAMPTON
Faculty of Physical and Applied Sciences
Electronics and Computer Science

Learning Models for Data Fusion and Spatial Regression with Untrustworthy Crowdsourced Information

by Matteo Venanzi

Supervisors: Prof. Nicholas R. Jennings and Dr. Alex Rogers
Examiner: Prof. Mahesan Niranjan

A mini-thesis submitted for transfer from MPhil to PhD

January 7, 2013

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES
ELECTRONICS AND COMPUTER SCIENCE

A mini-thesis submitted for transfer from MPhil to PhD

by Matteo Venanzi

Information trustworthiness is a major issue in crowdsourcing application. In more detail, the practice of leveraging internet communities, commonly referred to as *crowds*, and social mobilisation for data collection tasks, taking advantage from the low cost of hiring people using conventional crowdsourcing platforms, such as Amazon Mechanical Turk or Crowdflower, and the global take up of mobile technologies through internet users, is threatened by the issue of untrustworthy users submitting low quality data. Then, one of the biggest challenges in this context is how to infer reliable knowledge from crowdsourced information, in particular how to fuse data reported by untrustworthy user, In particular, such users are typically only interested in the monetary reward paid for executing the task, thus exerting the minimum effort in taking observations so producing low quality data.

Against this background, the research presented in this report investigates reliable inference approaches to the fusion of unreliable data in crowdsourcing applications. In particular, we present two new models that formally model the concept of user trustworthiness in data fusion and spatial regression tasks with untrustworthy data. We then provide two algorithms that estimate the trustworthiness of each user and the fused output under such models. Furthermore, empirical results on synthetic and real-world datasets show the efficacy of our approach against the state-of-the-art algorithms. Finally, future work will focus on further extending our models to improve their applicability and on addressing the other requirements for a reliable crowdsourcing platform that are still open questions in our research.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

U	Set of users.
R	Set of reported observations.
$\mathbf{x}_0$	Ground truth target position.
$\mathbf{r}_i$	Target estimate reported by user i.
$\mathbf{x}_i$	Expected value of the target observation reported by user i.
θ_i	Precision of the target measurement by user i.
$\mathbf{x}$	Random variable for the target position.
$p(\mathbf{x}\|\mathbf{r}_i)$	Probability density function for $\mathbf{x}$ given $\mathbf{r}_i$
t_i	Trustworthiness of observer i.
$\mathbf{t}$	Set of trustworthiness parameters.
$p(\mathbf{x}\|\mathbf{r}_i)$	Probability density function for $\mathbf{x}$ given $\mathbf{r}_i$ and t_i
$f(\mathbf{x}\|R, \mathbf{t})$	Fusion function.
θ_f	Precision of the fused estimate.
μ_f	Mean of the fused estimate.
$L(t_i\|\mathbf{r}_i, f)$	Likelihood function for the single t_i parameter.
$L(t\|R)$	Likelihood function for the $\mathbf{t}$ parameters.
$\mathbf{t}_{ML}$	Maximum likelihood estimate of $\mathbf{t}$.
ϵ	Input noise rate.
O	Asymptotical complexity.
$\mathcal{N}$	Normal distribution.
U	Uniform distribution.
ρ	Percentage of untrustworthy observations.
k	Parameter of the LOF algorithm.
β	Parameter of the RM algorithm.
y	Reported measurement in a spatial report.
$\tilde{y}$	Reported measurement in a spatial report.
$K(\boldsymbol{x}, \boldsymbol{x}')$	Covariance function of the Gaussian process.
$m(\boldsymbol{x}, \boldsymbol{x}')$	Covariance function of the Gaussian process.
Σ_x	Diagonal marie of the noise rates.
$\boldsymbol{\Theta}$	Set of hyperparameters.
$\boldsymbol{\theta}$	Subset of covariance function hyperparameters.

Chapter 1

Introduction

Over the last decade, *crowdsourcing* has emerged as a new data-collection technique whose principle is to harvest the 2.17 billion internet users[1] (31.5% of the world's population) and spur them to get involved in micro-tasks, such as solving a problem, answering a question, or providing data, and this, when put together, can help solve complex, highly decentralised data-collection tasks. In particular, crowdsourcing is now widely used for solving information gathering tasks, where crowdsourcing is an important source of subjects for social projects and experimental research.

In more detail, in crowdsourced information gathering, a *task requestor*, or taskmaster (e.g., a company, an institutional organisation or a single user) posts for the internet community a task of gathering some environmental observations (e.g., a sensor measurement, or a picture of an observed target) and, in response to this, a set of users, or *task executors*, namely the physical users who accept the task, report their observations. In practice, such information reporting processes may occur either on a voluntary basis, where the user spontaneously submits reports of their own volition, or on a monetary base, where a contracted reward is paid by the task requestor to the user for each report. In addition, to support access of the task requestors to the crowdsourcing market, there are now a number of crowdsourcing web services, such as Amazon Mechanical Turk, Crowdflower and oDesk[2] that allow users to post tasks, collect the answers and pay task executors in this way.

The success of crowdsourcing in such applications has followed the growth of the internet population and the growing free time that people dedicate to internet activities.[3] Nowadays, many companies and corporations are increasingly crowdsourcing parts of

[1] Data source: Internet World Stats `internetworldstats.com`

[2] For reference, see `mturk.com`, `crowdflower.com`, `odesk.com`

[3] A recent survey stated that U.S people spent an average of 13 hours per week online in 2010, equal to 74% of their free time. Source: `eweek.com`)

FIGURE 1.1: The Haiti-Ushahidi project. The live Crowdmap of help requests, clustered in the red blobs, submitted through social media, after the 2011 Haiti earthquake.

their everyday business operations to profit from low cost labour and additional benefits of parallelising and decentralising the whole information gathering process.[4]

Furthermore, crowdsourcing approaches have been explored also in the non-profit sector which we particularly focus on in this work. Specifically, various projects in the citizen science and the disaster response domain showed that crowds are effective in providing valuable information for scientific and social purposes purely motivated by intrinsic motivations, such as personal interests and social incentives.[5] Technologically, this is particularly driven by the fact that today 1.08 billion people[6] (i.e. half of the internet's users) are equipped with smartphones and other mobile technologies, with 3G internet connection, that is used as an on-board computing platform equipped with video sensor, camera sensor and Global Positioning System (GPS) sensor. In particular, such a worldwide distribution of internet-connected users can be used to retrieve data from remote areas by asking people to submit the requested information content through their cell phones. In this context, the disaster management domain has been witness to the great potential of such crowdsourcing from the ubiquitous networks of local users where, concretely, a number of projects that followed the recent environmental disasters and humanitarian crisis leveraged local people to provide real-time information to first responders or to international monitoring organisations. For example, after the devastating earthquake in Haiti, 2011, a mapping platform called Ushahidi (`ushahidi.com`), was set up to allow people to fill a map by reporting geo-tagged facts of disaster events located in their area, such as trapped persons or damaged buildings. This created a crowdsourced live map of the disaster that was useful to the first responders to coordinate their rescue operations (Figure 1.1). Also, during the political crisis in Egypt,

[4]For examples, see `threadless.com`, `istockphoto.com` and `innocentive.com`)
[5]For examples, see `ispot.com` and `ushahidi.com`
[6]Data source: The Next Web `thenextweb.com`

2011, people sent reports to the Arabic Network of Human Right Information (ANHRI, `anhri.net`) through a mobile app to document the atrocities happening around them and help the international community to monitor the evolution of the crisis.

Beyond the disaster response domain, the same modalities of leveraging people ubiquitously located as problem solvers or data providers is also currently used for security purposes by the UK police, that asks the public to help identify images of suspects by sharing them with the users living in the neighbourhood where the suspect was sighted through a mobile phone app (Facewatch, `facewatch.co.uk`), or by the US police to crowdsource the patrol of the Texas-Mexico boarder by asking people to watch the images of internet cameras and report suspicious events (Watchboard, 2006).

In all these scenarios, the mobilisation of people to report information can be a key contribution to successful disaster management, potentially helping to save more human lives (Heinzelman and Waters, 2010). However, together with all its advantages, crowdsourcing also poses new challenges about how to manage the vast amount of information generated by such a mass social mobilisation. In particular, the primary challenge for task requestors is to convert to, or to *fuse*, the dataset collected through the crowd into a global prediction of the output. Specifically, such a data fusion problem typically involves to find the correct answer among the dataset of redundant observations. In some cases, crowdsourcing involve collecting observations of a non-stationary quantity, for example an radiation map or a weather map. In these cases, the fusion problem is to estimate the function that is likely to represents the observed environmental process.

Another challenge is to cope with the uncertainty about the *trustworthiness* of each report as there is no absolute guarantee as to what the user has reported is accurate and correct. For example, in the Hait-Ushahidi project there were cases of people misreporting the real urgency of their submitted help request to get more food or to receive help sooner, or in the Egypt-ANHRI project, some people could not report events genuinely as they could have an interest to hide the real happenings to the international community. More in general, the trustworthiness of crowd generated content is the issue that this work addresses in the context of fusing data and inferring the unobserved ground truth of crowdsourced problems. Thus, the intent of this work is to provides models and solutions that could help clarify the issue of how safe is it to trust and make inference with cheap information gathered from crowds.

In more detail, this issue of information reliability arises from the fact that people are not always accurate and have subjective biases as lay observers. In particular, user misreporting in crowdsourcing can be motivated by the fact that some users are only interested in receiving the reward paid for the task, therefore they do not commit to exert the required effort to execute the task, so producing low quality data, e.g. a wrong radioactivity measure or an inaccurate estimate of a location target. In addition, some users might even have an interest in misleading the decision of the task requestor to

their advantage, for instance to get more food or to receive a rescue intervention more promptly, or to delocalise the search for a subject by the police. All of these concerns feed the criticism that crowdsourced information is highly unreliable and that crowdsourced solutions may not always be a reliable alternative to the traditional approach of hiring a team of experts for the same purposes.

Against this background, we argue that there is a need for new intelligent systems to tackle the challenge of making reliable inference on crowdsourced information under the uncertainty of the individual trustworthiness of the users. Such systems should help task requestors to find the true value among a set of redundant, and perhaps contradictory, reported observations by identifying which reports are more reliable and which should be discarded. However, the challenge for such a system to address is a complex decision-making problem in a setting of many real-world problems where a report not only consists of a simple observed value but it could also include some values characterising the uncertainty of the user about its observation, such as the precision of a sensor or a reported confidence level of an observation. Furthermore, as discussed above, a crowdsourcing problem could be targeted at estimating a spatio-temporal function, such a radiation map, a temperature map or a weather map, for which inference is more challenging because of the requirement of taking such a spatial and temporal correlation of the data into account to make reliable predictions of the underlying function.

Against this background, this research is concerned with exploring new Artificial Intelligence (AI) approaches to the problem of making reliable inference on untrustworthy crowdsourced information and is targeted at providing new inference models and algorithm to be applicable to a wide range of real crowdsourcing problems.

1.1 Research Requirements

From the discussion of the problem in the previous section, this following are the key requirements for this research:

Req. 1: **Data Fusion for Crowdsourced Stationary Targets.**
The first basic requirement for the algorithm is to be able to estimate the value of a crowdsourced stationary target, i.e. a quantity that can be assumed to remain unchanged throughout the observation process. For instance, this requirement is relevant for the problem of crowdsourcing a location target, such as the position of a radio mast (see Chapter 3 for more details) or to localise a person trapped in a building. Specifically, the algorithm must be able to estimate the observed target feature (e.g., the location) as accurately as possible, while dealing with the individual subjectivity and the trustworthiness of the reports.

Req. 2: **Spatio-Temporal Inference for Crowdsourced Non-Stationary Targets.**

Extending the previous requirement, the system must also be able to estimate the ground truth for a crowdsourced non-stationary target. In more detail, this requirement refers to the problem of crowdsourcing a spatio-temporal function, such as a radiation field, a temperature map, or a weather map (see chapter 5 for more details), where the output of the algorithm must be an spatio-temporal estimate of the process that people are trying to observe.

Req. 3: **Predictive Uncertainty.**

When estimating the ground truth, a key requirement for the algorithm is to provide information about the uncertainty of its prediction, also called its *predictive uncertainty* (Quinonero-Candela et al., 2006). This is an essential requirement for many decision making tasks. For example, on awareness of the risk of inefficiently allocating the limited rescue forces, a rescue operator would prefer a prediction saying: "there is 40% probability that a person is trapped in this building" rather than the much less informative statement: "there is a person trapped in this building". Crucially, the algorithm must provide the value of the predictive uncertainty as part of its prediction in order to evaluate the informativeness of its estimate.

Req. 4: **Considering Reported Uncertainties.**

As discussed earlier, learning from human data must involve dealing with the uncertainty in the reports that relates to the subjectivity and accuracy of the human observers. In this respect, an important aspect to consider is the *reported uncertainty* that at times the user provides to describe the inaccuracies that it is aware of in its observations. Specifically, the user could report the precision of its observation obtained through self appraisal of its degree of confidence about the submitted answer, e.g. Likert-scale value used in a crowdsourced questionnaire, or empirically derived from the measuring tool. For instance, for reporting GPS locations, such a precision is provided by the GPS receiver itself on the basis of the number and geometry of the satellites being used to generate the fix. Thus, such crowdsourcing must permit solutions in which the report set includes not only pointwise observations, but also reported uncertainties.

Req. 5: **Learning User Trustworthiness.**

To be able to make reliable predictions of the ground truth, an essential requirement is to estimate the trustworthiness of each report and correlate it to the individual trustworthiness of each user. Specifically, we consider a report as untrustworthy, with respect to the observed target, when its value is significantly distant from the real target value and, in turn, an untrustworthy user is the one that consistently reports untrustworthy observations. Thus, learning

user trustworthiness is crucial for the algorithm to improve the accuracy of its inference, as opposed to the approach of considering all reports as being equally trustworthy (which is likely to lead to poor quality estimates).

Req. 6: **Verification.**
A way to reduce the uncertainty about the degree of trustworthiness of a report is verification, i.e. to ask a trusted (or more trusted) verifier to check and approve some reported information. Alternatively, when such a verifier user is not available, a way to verify a report is based on redundancy, i.e. repeatedly getting extra independent observations from the crowd. However, from the taskmaster's perspective, verification must be driven by a certain level of accuracy that it desires to achieve on its data with respect to a limited budget that it allocates to verification. Thus, the requirement for the algorithm is to efficiently use the available budget to reach the required data accuracy in verification.

Req. 7: **Incentivising Truthful Reporting.**
Finally, another way to improve information trustworthiness is to provide incentives to the user to report truthfully their observations. Specifically, this requirement refers to leveraging the rewards paid to the user within an appropriate incentive scheme that guarantees that it will always exert the maximum effort in taking its observation and will not deviate from reporting the truly observed value. In this way, such incentives should reduce the uncertainty about trustworthiness of an individual report and should leave potentially only the component of human subjectivity and limited accuracy as a source of noise in the data.

Against these requirements, this work mainly focusses on the first four of them i.e. inference models for crowdsourcing stationary (requirement 1) and non-stationary quantities (requirement 2), while considering user trustworthiness (requirement 3) and reported uncertainty (requirement 4), and providing estimates of predictive uncertainty (requirement 5). The other requirements of information verification (requirement 6) and incentive engineering for truthful reporting (requirement 7) are left as part of the future work.

1.2 Research Challenges

For this problem, there are a number of challenges that must be addressed for a solution to meet the requirements identified above. In particular, we discuss the three key challenges that we have addressed in this work to date:

1. When estimating a target value using crowdsourced data, a major challenge is to assume that such a problem's ground truth is unknown to the algorithm. For example, this is the case when first responders want to crowdsource the map of damaged buildings (see the example of the Haiti-Ushahidi project described earlier) or to monitor an dangerous nuclear cloud spreading across a disaster area, as it happened in the aftermath of the 2010 Fukushima earthquake in Japan. In such cases, it is difficult for the algorithm to evaluate the quality of its prediction or to estimate the trustworthiness of a report in when the problem's ground truth is not available.

2. The second challenge is to deal with the inaccuracies of human information. This is a key aspect to take into consideration when computing the aggregate of the reports into a single output. In more detail, crowdsourcing is a domain in which the information sources are humans, that act as observers and play the role of soft sensors in the emergent network of local responders. This moves beyond the use of traditional hard sensors to locate, characterise and describe physical and non physical targets. Clearly, however, data provided by humans is different from conventional sensor data in terms of subjectivity, malicious behaviour and the limited accuracy of human observers (Hall and Jordan, 2010). For example, the Ushahidi teams estimated that only the 1% of the 4000 Haiti-Ushahidi reports could be classified as reliable while the remaining set of these contained several false reports and misreported help requests due to people incentivised by the emergency situation to get more food aids for themselves. Thus, the challenge for the algorithm is to fuse crowdsourced data by seeking new fusion methods for dealing with human information that go beyond the data fusion methods for traditional hard sensors .

3. Finally, to use redundancy to verify a report, or to identify the most likely aggregated output, an important issue to address is how to get redundancy when the reports refer to a non-stationary function target (Requirement 2). Generally speaking, it is easy to create redundancy in the data for a static target case by just getting an extra report from the crowd. However, it is much harder to have redundant reports for a spatio-temporal target as, in fact, it is unlikely to get two reports from exactly the same location, or it might be impossible to ask for a report from a previous observation. Thus, the challenge is to characterise the concept of redundancy in the function space for the computing the aggregated output on untrustworthy spatio-temporal observations.

From these challenges, a number of research communities have been trying to find solutions to various assets of the overarching problem. These communities include the fields of human computation, citizen science, machine learning and multi-agent systems,. In particular, some work addressed the problem of inferring the ground truth from a set of noisy observations reported from multiple users (Bachrach et al., 2012; Kamar et al.,

2012; Welinder et al., 2010; Whitehill et al., 2009). However, none of the existing solutions focus on the aspect of considering reported user uncertainties as part of the input data outlined in the requirement 3, while they only consider the setting of users reporting pointwise observations. As such, inference in such models can be improved by considering the precisions reported by the users as an indication of the reliability of an observation.

In terms of addressing the problem of aggregating a set of reported estimates, a vast literature from the data fusion field provides a number of techniques to effectively fuse multiple noisy sensor estimates (see Section 2.3 form more details). However, while such multi-sensor fusion techniques could potentially be applied also in crowdsourcing, exploiting the paradigm of human usersas soft sensors (although this has not been done in previous work yet), the issue in doing this is that the notion of noise as modelled in sensor fusion does not capture the broader dimensions of uncertainty of human data. Thus, we seek new fusion techniques specialised on human data to be potentially more suitable for inference in the crowdsourcing setting.

Against this background, the research presented in this report investigates novel inference models and algorithms to deal with untrustworthy information in crowdsourcing. Specifically, the objective of this work is to provide new solutions to the problems of (i) fusing multiple untrustworthy estimates into a single prediction the ground truth and (ii) assessing the trustworthiness of a single user. By doing this, we expect to improve the reliability of crowdsourcing tools by making them more suitable and more robust for large-scale applications.

1.3 Research Contributions

Against the requirements and challenges described above, the contributions of this report to the state of the art are stated as follows:

- We provide a novel trust-based data fusion model which addresses the problem of crowdsourcing stationary quantities, with a dataset comprising reported observations and precisions. The salient feature of such models is to formally represent the concept of user trustworthiness to capture the uncertainty about the reliability the user's reports. Specifically, such a model represent trustworthiness as a latent scaling parameter of the uncertainty of an estimates. In this way, the effect is gradually de-emphasize noise on an untrustworthy estimate in the fused output. In more detail, the contribution of this work is presented in the following paper:

  > M. Venanzi, A. Rogers, N.R. Jennings. Considering trustworthiness for fusing crowdsourced data. *To be submitted to the 12th International*

Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2013.

- We provide the first algorithm, called MaxTrust, that, applies our user trust model to the problem crowdsourced target localisation problem. This algorithm infers the location of the target through the fusion the reports taking into account the level of trustworthiness of each report, and provides the estimates of the trustworthiness of each user.

- We also introduce a new trust-based inference approach for crowdsourcing non-stationary quantities. Specifically, we provide the first trust-based heteroscedastic Gaussian process regression model, called TrustHGP, to address the problem of inferring a spatial function from untrustworthy location-based observations. Such a model has the qualities of making spatial predictions of the observed function and estimating the trustworthiness of each user using the principled Bayesian inference framework of the Gaussian process.

- We show that both our algorithm are affective in solving real crowdsourcing problems through their empirical evaluation on real-world datasets. Specifically, for the data fusion model, we evaluate MaxTrust with an experiment on cell tower localisation using crowdsourced cell phone data provided provided by the OpenSignalMap project (`opensignalmap.com`). In particular, we show that our algorithm improves by 21% (on average, corresponding to an error lowered by 185 meters) the accuracy in localising the cell towers. Then, for our spatial inference model, we evaluate our TrustHGP on the key disaster response application of crowdsourced radiation monitoring using data from the 2011 Fukishima nuclear disaster provided by the network crowdsourced sensors connected to the Cosm platform (`cosm.com`). In particular, we show that the TrustHGP outperforms the standard GP approaches in making more accurate, by 13%, and more informative, by 89%, spatial predicions of radiation levels.

1.4 Report Outline

The remainder of this report is structured as follows.

In Chapter 2, we provide the background to this work by reviewing research related to reliable crowdsourcing, with emphasis on data fusion and spatial regression models.

In Chapter 3, addressing the problem of crowdsourcing static targets, we present our trust-based data fusion model for crowdsourced target localisation and detail the MaxTrust algorithm to estimate the target and the trustworthiness of each user jointly. This includes an empirical evaluation of the algorithm on location data with an experiment of cell tower localisation.

In Chapter 4, addressing the problem of crowdsourcing non-stationary functions, we present our trust-based, heteroscedastic Gaussian process regression model and describe the TrustHGP algorithm for learning the latent trustworthiness parameters under such a model. This includes the evaluate of the algorithm on the Cosm dataset of crowdsourced nuclear radiation sensor data.

Chapter 5 concludes and outlines the future work to follow this research, including a detailed plan of the activities until the completion of the PhD.

Chapter 2

Background

In this chapter, we review the key background research relating to the problem of untrustworthy information in crowdsourcing applications that will provide the theories for the models and the algorithms presented in Chapters 3 and 4. Specifically, the chapter begins with an overview of computational approaches to trust assessment in crowdsourcing (Section 4.1.1). Subsequently, Section 3.1.1 reviews the approaches to reliable crowdsourcing, in particular discussing the problem of aggregating multiple reports and introducing algorithms from the data fusion (Section 3.1.2) and spatial regression (Section 2.4) literatures. Finally, Section 4.4 summarises the contribution of the existing work against our problem's requirements that we described in the previous chapter (Section 1.2) and the limitations that need to be addressed by our research.

2.1 Approaches for Trusting Users

In the previous chapter, we argued that making crowdsourced information more reliable is primarily concerned with evaluating the trustworthiness of the reports and, on such a basis, assessing the trustworthiness of the reporters (see Section 1.2). In this respect, this section introduces some approaches to computational trust that are useful to address this requirement.

The first general approach to determining the trustworthiness of a user is to rely on historical data, and possibly also reputation reports from third parties, to infer the user's reliability (Ramchurn and Jennings, 2005). In this way, trustworthiness evaluation is supported by some empirical evidence of how the user behaved in the past with the system. However, one difficulty of applying such an approach in crowdsourcing is given by the openness of the crowd, i.e participants can join and leave at any time, and it is easy for a crowd member to anonymise its identity. This openness can make it unlikely to have multiple encounters with the same users which, in turns, makes it hard to assess

their reputation. In addition, building trust relying on historical data is unreliable when there are users that strategise with their reports to build a deceptive image of their reputation in the eyes of the taskmaster (Archak and Sundararajan, 2009). Since strategic behaviour of this kind is highly likely in a human reporter, we will not consider reputation-based trust in our work.

Alternatively, another approach to trust formation is based on *consensus*. That is, the technique of computing user trustworthiness according to the number of other independent observations within the crowd matching the one reported by the user (Kamar and Horvitz, 2012). Generally speaking, the consensus approach is typically applied when the *majority assumption* holds, i.e. the majority of the opinions will eventually agree on the ground truth, thus the consensus opinion is likely to reveal such ground truth. Also, as the majority assumption relates to datasets with high redundancy, then consensus in crowdsourcing is supported by the fact that it is relatively easy to create redundancy in the data by gathering extra reports at low cost. For this reason, many existing models advocate the consensus approach as a possible solution to the problem of trusting unknown users in crowdsourcing (Kamar et al., 2012; Raykar et al., 2010; Whitehill et al., 2009).

In practice, consensus is easy to compute for discrete classification problems where each report can be seen as a vote on a certain outcome and the consensus outcome is the most voted one, or for continuous classification problems by taking the average of the reported real values as the consensus value. However, it is less straightforward to compute consensus when the reports are estimates that includes also the precision of an observation, as is the case in the setting that we address in this work (requirement 4). Therefore, while adopting the consensus approach for trustworthiness evaluation, we will look at consensus techniques for crowdsourced estimates that we will review later in this chapter (see Section 3.1.2). In what follows, we review more general, non-trust approaches to reliable crowdsourcing.

2.2 Approaches to Reliable Crowdsourcing

In the previous chapter, we described crowdsourcing systems as human-powered tools that are useful to solve highly decentralised information gathering tasks (Section 1.2). We also mentioned that, for such systems, the natural core problem is to cope with the reliability of the human sources and the trustworthiness of their reports in order to achieve good quality results. In this section, we review the main approaches to reliable crowdsourcing proposed in the literature that are relevant to address this requirement (requirement 5).

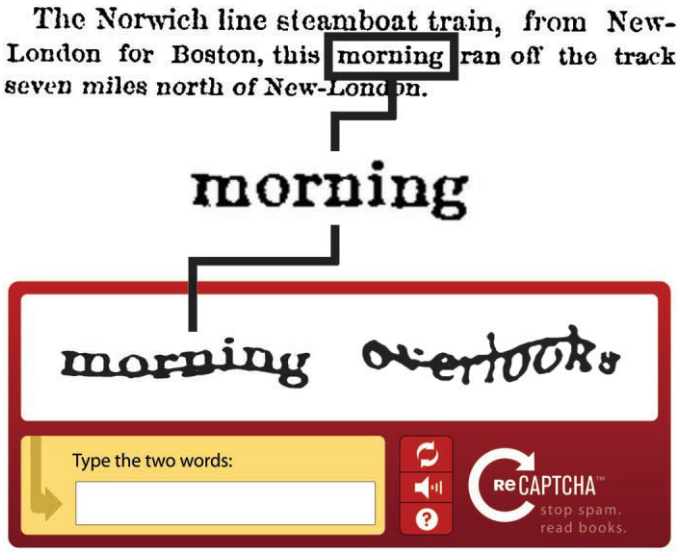


FIGURE 2.1: Example of gold-based quality assessment performed by ReCAPTCHA. The correctness of the user's answer to the unknown challenge word (morning) is evaluated on the basis of its answer to the known control word (overlooks).

2.2.1 Gold-Based Approach

The gold-based approach aims to identify unreliable users with a set of predefined ground truths, or *gold standards* (Oleson et al., 2011). Specifically, each user is trained on the gold set, i.e. the set of tasks with gold standards, and its trustworthiness is evaluated on the basis of the discrepancy between its answers and the correct one. Currently, many crowdsourcing platforms adopt gold-based mechanisms to provide assurance of data reliability to task requestors, including Amazon Mechamical Turk(`mturk.com`) , Crowflower (`crowdflower.com`) and ReCAPTCHA (`recaptcha.net`). For example, both Amazon Mechanicical Turk and Crowdflower offer the feature of specifying gold standards when creating Human Intelligent Tasks (HITs). Also, ReCAPTCHA, a tool that uses human answers to digitalise text, performs the *control word* test to decide whether an input answer to an unknown word is trustworthy or not (Figure 2.1). Specifically, such a test consists of presenting two words to the user in random sequence, where one of the two words is the actual challenge word that is unknown, and the other one is the control word that is known. In such a way, since the user does not know which of the two is the control word, this test is likely to increase the chances that the challenge word will by typed correctly.

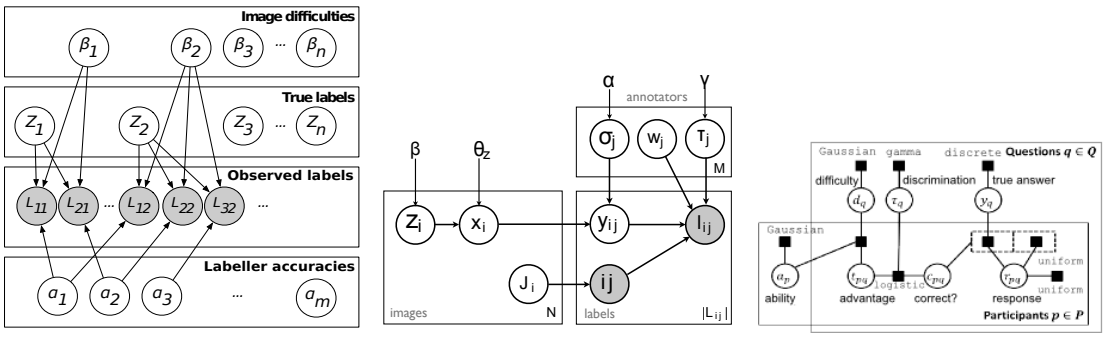
However, one disadvantage of the gold-based approach is that the cost of training users with gold standards is not always supported by an absolute guarantee of substantial gain in the quality of the data (Ipeirotis, 2010). Also, since we base our work on the unsupervised setting where gold standard are not available (requirement 2), then gold-based solutions will not be considered in this report.

2.2.2 Machine Learning Approach

Another approach to deal with the uncertainty of user reliability is to apply machine learning techniques to the data and try to estimate the correct answer and also learn some additional information about the task and the users. In general, the machine leaning approach in crowdsourcing consists of designing a statistical model of a crowd reporting process and then applying standard inference algorithms to estimate unobserved parameters based on the data gathered from the crowd (Dawid and Skene, 1979; Whitehill et al., 2009; Bachrach et al., 2012). In particular, there are a number of existing inference models that address the requirement of learning the ground truth from unreliable reports that is part of our problem. We review these models in the remainder of this sub-section.

2.2.2.1 EM Approach

The first algorithm to infer the answer for a classification task from multiple reports was proposed by Dawid and Skene in 1979 (well before the advent of crowdsourcing) to study the advantage of using low-cost noisy labellers for an image annotation task in the context of supervised learning. Specifically, the base of their algorithm is expectation-maximisation (EM): a well-known iterative method to find the maximum likelihood estimates of the parameters in a statistical model (Dempster et al., 1977). In more detail, the EM approach for a crowdsourcing problem consists of having an expectation step in which the correct answer is estimated from the data based on the current model parameters, where such parameters typically define the dependencies of the global output from the single observation, and then a maximisation step that updates such parameters by maximising the likelihood of the model. Ultimately, EM has been applied to more complicated problems such as image labelling, galaxy classification and IQ testing (Whitehill et al., 2009; Kamar et al., 2012; Bachrach et al., 2012). Furthermore, other work has used the Bayesian version of EM that considers some prior probabilities in the model and obtains posterior estimates of the parameters using the same iterative algorithm (DeGroot, 2004). This Bayesian approach is particularly useful when some prior knowledge about the problem is available. For example, we might know that some observations are more reliable than others and so chose the appropriate priors. This approach can lead to more accurate inference; as was shown by Raykar et al. (2010) in a crowdsourcing application of classifying cancer diagnoses in the medical domain. More generally, the EM approach is associated with the graphical modelling technique that has inspired the majority of work presented in this area, as it enables a clear and explicit design of a crowdsourcing model. As these models address the requirement of estimating the global output (requirement 3) and also the trustworthiness of each reporter (requirement 5), then we review them in the next section.



(a) Whitehill et al. (b) Welinder et al. (c) Bachrach et al.

FIGURE 2.2: Graphical models proposed in previous work for crowdsourced image labelling (a, b) IQ testing (c)

2.2.2.2 Graphical Models

In machine learning, graphical models are tools for representing an arbitrary probability distribution as a graph and so highlight its factorisation properties (Bishop, 2006). Specifically, a graphical model represents a set of random variable as nodes, distinguishing between observed nodes (shaded) and unobserved or latent nodes (unshaded), and directed links describe probabilistic dependencies between pairs of nodes. In these models, inference flows through the graph from the sources (nodes with no incoming links) to the leafs (nodes with no outgoing links) based on the conditional dependencies outlined by the links. In such a way, it is possible to estimate the probability densities of the latent nodes based on the data feeding the observed nodes.

A number of graphical models for crowdsourcing problems were recently presented, some of these are showed in Figure 2.2. In detail, Whitehill et al. (2009) model an image labelling task considering a set of n images, each of which belongs to one of two possible categories (e.g. face/non-face, male/female), and assuming that the observed label $L_{i,j}$ reported by labeler i for the image j depends upon the true binary label Z_j, the image difficulty β_j and the expertise of the labeler α_i (Figure 2.2 (a)). Then, putting Gaussian priors on α and β, the maximum-a-posteriori inference of the parameters is made through the EM algorithm. Furthermore, Welinder et al. (2010) extend this model to consider the accuracy of the labeler in a multidimensional space, with variables representing the competence, the expertise and the bias of the labeler, i.e. the α, w and τ parameters respectively (Figure 2.2 (b)). Finally, Bachrach et al. (2012) introduced a graphical model to analyse the responses from multiple participants to a set of questions and find the correct answer for each question, the difficulty level of the answer and the ability of the participants (Figure 2.2 (c)). In this context, all these algorithms were shown to empirically outperform the naïve majority voting approach i.e. find the correct answer as the one that received the largest number of votes amongst the users.

In general, inference algorithms based on graphical models are the first concrete solution for the learning problem in crowdsourcing. However, the main issue of such a graphical

modelling approach is that, while the model can be arbitrarily complicated by adding new nodes to the graph, inference can rapidly become analytically intractable. For this reason, a number of techniques to approximate posterior inference on graphical models are available, among these expectation-propagation, variational methods and sampling methods (see Bishop (2006) for a detailed overview). However the problem of such approximation techniques is that they are prone to find sub-optimal solutions in non-convex problems or to require many samples to achieve a good level of approximation.

Specifically for our problem, none of the discussed models considers observations with reported uncertainty as we require (Requirement 4), instead they focus only on pointwise observations, i.e. observations consisting of a single value with no numerical uncertainty. Therefore, using a similar probabilistic approach, in this work we will go a step forward in modelling also the precision reported by the user as part of the input data.

2.3 Data Fusion Methods for Untrustworthy Data

Addressing the requirement of fusing multiple observations into a global estimate, which is part of requirements 1 and 2, this section introduces data fusion methods. In particular, data fusion research studies how to integrate estimates from multiple sources to perform more efficient inference (Thrun et al., 2001). Typically, data fusion considers information sources as physical sensors that are employed in a target monitoring task, where each sensor provides observations of the target in its monitoring area, and fusion algorithms deal with how to aggregate the set of sensor observations into a single estimate to accurately predict the target position. In addition, as sensors are noisy, the requirement for such algorithms is to detect unreliable sensors and filter their noise in the fused estimate.

Thus, in the traditional data fusion approach, the human user is primarily viewed as an interpreter of the processing result that ultimately transforms the fused estimate into knowledge, and only rarely input data from human observers is considered. However, as discussed in Section 1.2, crowdsourcing introduces the new view of having humans acting as sensors and using their smart phones as an on-board computing platform to provide observations. For this reason, a new focus is emerging that studies the applicability of the current multi-sensor fusion algorithms to human information. In this vein, Hall and Jordan (2010) point out the key differences between information produced by humans compared to sensor information that needs to be taken into account in the fusion process. Particularly, they highlight the different types of noise from the two data sources, arguing that the inaccuracies of a sensor reading typically depends on the faults that temporally or permanently affect the functioning of the sensor, while it is unrealistic to think that the sensor would deliberately misreport its observation as might occur in crowdsourcing settings. Thus, while the problem of dealing with unreliable estimates in sensor fusion

is typically a problem modelling the sensor faults, now the changing role of humans in information fusion introduces new types of noise in the data related to subjectivity, expertise and bias of the human observers.

From this, we identify data fusion as a suitable methodology to meet requirements 1 and 2, although we require fusion methods that can effectively deal with the noise of human data. To this end, we now provide an overview of two standard fusion techniques for fusing multiple probabilistic estimates referring to the single-hypothesis and the multi-hypothesis fusion case. Then, we will review the outlier detection approach to address the problem of identifying untrustworthy estimates. Finally, we will discuss the class of algorithms specialised on learning sensor trustworthiness in sensor fusion.

2.3.1 Single-Hypothesis Case

In the single-hypothesis case, the set of observations refers to only one hypothesis of a correct answer. For example, a single-hypothesis crowdsoucing problem is typically the situation of collecting multiple observations of a static location target, such as cell tower locations, in which the output of the fusion of the set of independent location reports is single estimate of the actual tower location. In this case, the standard approach is to fuse the estimates together to reduce their noise in the merged estimate. Specifically, the covariance intersection (CI) method (Julier and Uhlmann, 2001) that is the standard method to fuse two Gaussian estimates is described below.

Covariance Intersection: Given two normally distributed estimates $e_1 = (\mu_1, \Sigma_1)$ and $e_2 = (\mu_2, \Sigma_2)$, where μ is the mean and Σ is the covariance matrix, then the covariance intersection estimate $e_{CI} = (\hat{\mu}_{CI}, \hat{\Sigma}_{CI})$ is computed as the linear sum the means weighted by the inverse of the covariance matrix (or precision matrix). That is:

$$\hat{\Sigma}_{CI}^{-1} = \Sigma_1^{-1} + \Sigma_2^{-1} \tag{2.1}$$

$$\hat{\mu}_{CI} = \hat{\Sigma}_{CI}(\mu_1\Sigma_1^{-1} + \mu_2\Sigma_2^{-1}) \tag{2.2}$$

Specifically, this CI method performs the fusion of multiple estimates considering both the mean and the precision of each Gaussian estimate. Now, this method is suitable to address the requirement 4 of considering reported uncertainties (precisions) because we can assume that each report is a subjective estimate of the observed target that can be fused together with the set of multiple observations through such a method. In particular, CI fuses the estimates as weighted by their individual precisions, i.e. the estimates with high precision (i.e. small variance in the univariate case) have higher weight in the fusion. In addition, as stated by Equation 2.1, the fused precision $\hat{\Sigma}_{CI}^{-1}$ is

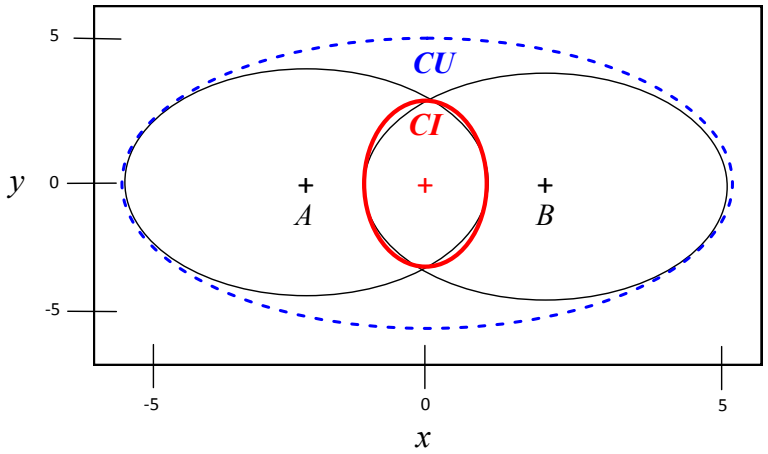


FIGURE 2.3: Merging two Gaussian estimates, represented by the two circles, using CI (solid line) and CU (dashed line).

computed as the sum of the precisions of each observation, therefore the uncertainty of the fused estimate decreases as more observations are added to the set. However, the potential issue of doing this is that the fusion can lead to wrong predictive outputs when the dataset includes untrustworthy reports. Thus, to rectify this, and so address our requirement 5 of performing trust-based inference, we envisage an extension of CI to take the trustworthiness levels of each user into account. This extension will be elaborated in our trust-based fusion model that we will present Chapter 3.

2.3.2 Multi-Hypothesis Case

In the multi-hypothesis case, there is more than one hypothesis which could be the correct answer due to the considerable variance in the set of reported estimates. For example, the multi-hypothesis setting occurs when the crowd observes a moving target where different observations describe the target position in different time instants. In this case, if the goal is to localise the target, then the conservative approach to aggregating the reports is not to discard any hypothesis and take their union as the most general output. Specifically the covariance union method (Reece and Roberts, 2010) to unify two Gaussian estimate is described as follows :

Covariance Union: Given two normally distributed estimates $e_1 = (\mu_1, \Sigma_1)$ and $e_2 = (\mu_2, \Sigma_2)$, where μ is the mean and Σ is the covariance matrix, then the covariance union estimate $e_{CU} = (\hat{\mu}_{CU}, \hat{\Sigma}_{CU})$ is any Gaussian estimate defined by the following constraint:

$$\begin{cases} \hat{\Sigma}_{CU} & \geq \Sigma_1 + (\hat{\mu}_{CU} - \mu_1)(\hat{\mu}_{CU} - \mu_1)^T \\ \hat{\Sigma}_{CU} & \geq \Sigma_2 + (\hat{\mu}_{CU} - \mu_2)(\hat{\mu}_{CU} - \mu_2)^T \end{cases} \tag{2.3}$$

Algorithm 1 LOF

Inputs :

R **:** report set.
kNN(r): k nearest neighbours for a report r.

Algorithm *LOF(R, k, t)*

1: Define $k\_distance(o)$ as the minimal distance of o from kNNs(o):
2: Compute reachability distances:
for each $r \in R$ **do**
for each $o \in kNN(r)$ **do**
$reach\_dist_k(r, o) = \max\{k\_distance(o), dist(r, o)\}$
end for
end for
3: Compute local reachability distances (lrd):
for each $r \in R$ **do**
$lrd(r) = \Big(\frac{\sum_{o \in kNN(r)} reach\_dist_k(r,o)}{|kNN(r)|}\Big)^{-1}$
4: **end for**
5: Compute local outlier factors (LOF):
for each $r \in R$ **do**
$LOF(r) = \Big(\frac{\sum_{o \in kNN(r)} \frac{lrd(o)}{lrd(r)}}{|kNN(r)|}\Big)^{-1}$
end for
6: Compute $\langle \hat{\mu}_{LOF}, \hat{\Sigma}_{LOF} \rangle$ fusing the inliers with $LOF(r) < t$.
7: **return** $(\hat{\mu}_{LOF}, \hat{\Sigma}_{LOF}, LOF(r))$

Then, among the family of the Gaussians defined by the constraint of Equation 2.3, the one that minimises some measurement of the size of $\hat{\Sigma}_{CU}$, e.g., $det|\hat{\Sigma}|$, or the ratio $det|\hat{\Sigma}|/\hat{\mu}$, is usually chosen. Formally:

$$\Sigma^*_{CU} = \arg\min_{\hat{\Sigma}_{CU}} \{det(\hat{\Sigma}_{CU})\} \tag{2.4}$$

Specifically, the CU method performs hypothesis merging through inflating the variance (in the univariate case) of the fusing estimate to include all the possible hypotheses. By doing this, the CU estimator has the property of always being consistent with all the possible hypothesis, as opposed to the CI estimator that at times is inconsistent with some estimates. For example, Figure 2.3 shows two Gaussian estimates, A and B, fused through CI and CU and it can be seen that CI is not consistent with A and B. Therefore, CU does not explicitly require us to know which observations are trustworthy and which are not, since it always takes the most general Gaussian estimate as the aggregated output. However, the drawback of doing this is that such a CU estimate is not very informative for making predictions due to its high level of uncertainty. Thus, as we seek aggregators with a good trade off between prediction accuracy and low uncertainty (requirement 3), the CU method will only be referred to as a conservative fusion benchmark in our approach.

2.3.3 Local Outlier Factor

A third way to deal with untrustworthy observations is to treat them as outliers. In detail, recalling the definition given by Hawkins (1980): "an outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism". This definition fits to the notion of an untrustworthy user given in Section 1.2, i.e. a user reporting observations that are significantly distant from the others. However, this only captures certain kinds of outliers, namely those points that outlay relative to the global dataset. For this reason, they are referred to as "global" outliers. In this context, a more general density-based definition of outliers in datasets with more complex structures is given by Breunig et al. (2000); they derive them as a point that outlies with regards to their local neighbourhood. Thus, these outliers are regarded as "local" or density-based outliers.

Given this background, one idea is to apply outlier-detection methods to a crowdsourced dataset to identify and remove untrustworthy observations. Subsequently, as part of our requirement 1, we can compute the aggregated estimate using the fusion methods described earlier applied to the inlier reports. To describe this methodology, we refer to the density-based outlier detection method of the *local outlier factor* (LOF) (Breunig et al., 2000). Specifically, such a method assigns a LOF score to each point as an indicator of its outlier level, measuring the relative density of the point compared to its neighbours. In doing this, it assumes that the density around an outlier is considerably different to the density around its neighbours. In more detail, the procedure for computing LOF scores is detailed in Algorithm 1. First, for each report r, the reachability distance of r from its neighbour is computed (step 2), where k is the input parameter that defines the locality region of r as the set of its k nearest neighbours. Next, Step 4 computes the local reachability distance $ldr(r)$ as the inverse of the mean reachability distance between r and its neighbours. Finally, in step 5, LOF(r) is computed as the ratio of its local reachability of r and the one of its neighbours. Once the scores are computed, the algorithm returns the fusion of the inliers identified as the reports with LOF lower than the threshold t. Furthermore, to measure the distance between two probabilistic estimates required in Step 2, the *Kullback-Leibler divergence* (KL) is typically used (Kullback and Leibler, 1951). This is a standard measure of the distance between two probability densities that, for the case of two normal densities of dimension d, the KL distance is expressed as follows:

$$\begin{aligned} KL(\mathcal{N}(\mu_1,\Sigma_1)||\mathcal{N}(\mu_2,\Sigma_2)) =& \frac{1}{2}\Big(tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1) \\ & - ln\left(\frac{det(\Sigma_1)}{det(\Sigma_2)}\right) - d\Big) \end{aligned} \tag{2.5}$$

In summary, the solution to the problem of identifying unreliable reports using outlier detection techniques is to compute the trustworthiness of each report as the LOF score and then compute the aggregate output on the inliers set. However, in so doing, a crucial step is to reflect the density of the report setin the choice of the outlier threshold t in order to enable this method to detect outliers correctly. Therefore, we will include this algorithm in our benchmarks and compare it to our approach in terms of performance and flexibility.

2.3.4 Sensor Fusion Methods

In the sphere of multi-sensor data fusion, there is a class of methods that deal with uncertainty sensor fusion using trust modelling techniques (Reece et al., 2009; Momani et al., 2010; Guan et al., 2009). In more detail, such models adopt a probabilistic representation of the uncertainty related to the reliability of a sensor and then incorporate such an element in the data fusion model. In this class, we examine the model presented Reece et al. (2009) that explicitly focus on a method for sensor noise recovery in presence of unknown fault types using the concept of sensor trustworthiness. Specifically, they provided an algorithm for a target localisation problem to estimate the target position from a dataset of multiple sensor observations dealing with the inaccuracies deriving from possible sensor faults. In particular, such an algorithm removes the noise from the sensor estimates in two stages. In the first stage of noise recovery, each sensor attempts to remove the faults from its observation based on a pre-defined set of fault models.[1] Then, the second stage assigns an individual level of trustworthiness to each sensor to characterise the noise of untrustworthy sensors. Specifically, their method evaluates sensor trustworthiness based on redundancy and identifies untrustworthy sensors as the outliers i.e. the sensors that report an estimate that is significantly distant from the fused estimate. Thus, Reece et al.'s work contributes to our research as it provides (i) a consensus rule for fusing observations gathered from untrustworthy sensors and (ii) a method for computing the trustworthiness of each sensors based on a distance measure, in this case the Mahalanobis distance, between the reported observation and the fusion. In more detail, these two parts of the algorithm are described in what follows.

Consensus Rule for Fusing Untrustworthy Gaussian Estimates: Given two univariate Gaussian estimates, $e_1 =< \mu_1, \theta_1 >$ and $e_2 =< \mu_2, \theta_2 >$, where μ and θ are the mean and precision of the Gaussian distribution respectively, and given $t_1, t_2 \in [0, 1]$ as their trustworthiness levels, then the consensus estimate between e_1 and e_2 is the

[1] Specifically, they consider drift, spike, shock and echo faults.

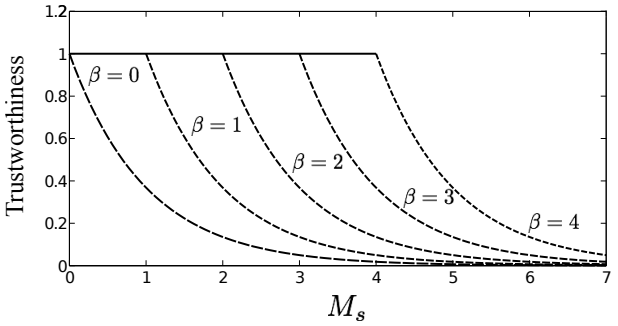


FIGURE 2.4: Plot of the trustworthiness function used in the Reece method varying the β parameter.

Gaussian $e_{cons} =< \mu_{cons}\theta_{cons} >$, with trustworthiness t_{cons} that is computed as follows:

$$\tilde{\mu} = (\theta_1\mu_1 + \theta_2\mu_2)/(\theta_1 + \theta_2) \tag{2.6}$$

$$\mu_{cons} = t_1 t_1 \tilde{\mu} + t_1(1 - t_2)\mu_i + t_2(1 - t_1)\tilde{\mu} \tag{2.7}$$

$$\tilde{\theta} = \theta_1 + \theta_2 \tag{2.8}$$

$$\theta_{cons} = t_1 t_2(\tilde{\theta} - (\tilde{\mu} - \mu_{cons})^2) + t_1(1 - t_2)(\theta_1 - (\mu_1 + \mu_{cons}^2)) + t_2(1 - t_1)(\theta_2 - (\mu_2 + \mu_{cons}^2)) \tag{2.9}$$

$$t_{cons} = t_1 + t_2 - t_1 t_2 \tag{2.10}$$

Mahalanobis Distance for Estimating Trustworthiness: Given the consensus estimate $e_{cons} =< \mu_{cons}, \theta_{cons} >$, then the trustworthiness of an univariate Gaussian estimate $e_i =< \mu_i, \theta_i >$ is defined as follows. Let M_s be the Mahalanobis distance between e_{cons} and e_i given by:

$$M_s = \frac{(\mu_{cons} - \mu_i)^2}{(\sigma_i + \sigma_{cons})} \tag{2.11}$$

Then, the trustworthiness t_i for the estimate e_i is:

$$t_i = \begin{cases} 1 & \text{if } M_s < \beta \\ \exp(-(M_s - \beta)) & \text{otherwise} \end{cases} \tag{2.12}$$

where β is the breakdown point in which the trustworthiness value starts to decrease exponentially, as Figure 2.4 shows by plotting the trustworthiness function for different β values. Then, combining these two rules of the Reece model in the EM approach described in Section 2.2.2.1, it is possible to obtain a new algorithm that iterates between the consensus fusion rule and the trustworthiness update rule, while appropriately tuning the β parameter. In this way, the algorithm estimates both the predictive answer and the

Algorithm 2 Reece Method

Inputs :

R : report set.
acc : accuracy bound.
$epochs$: number of training epochs.

Algorithm *ReeceMethod(R)*

1: Start with uniform max trust values on all the reports:
$$t^{(0)} =< 1, \ldots, 1 >$$
2: **while** ($|t^{(k-1)} - t^{(k)}| < acc$ **or** $k > epochs$) **do**
3: **(E step)** Fuse the observations using the consensus rule based on $t^{(k-1)}$:
$$f^{(k)} = consensus(R, t^{(k-1)})$$
4: **(M step)** Update trustworthiness parameters based on $f^{(k)}$:
for $i = 1 : n$ **do**
$$t_i^k = M_s(R, f^{(k)})$$
end for
5: **end while**
6: **return** $(t^{(k)}, f^{(k)})$

user's trustworthiness. In more detail, Algorithm 2 describes the EM implementation of Reece's method.

Thus, Reece et al.'s work on untrustworthy sensors provides the basis for a solution to solve our problem. However, this algorithm is natively designed for a multi-sensor domain that, as we discussed earlier (Section 3.1.2), is significantly different from the crowdsourcing one. Therefore, while using the Reece method as the main benchmark for our approach, we will empirically investigate whether this method is actually suitable for our problem. In so doing, we also contribute to the state of the art by providing its numerical evaluation on a crowdsourcing problem.

2.4 Crowdsourcing Spatial Functions

So far, we have discussed fusion techniques for aggregating multiple observations referring to an observed stationary value, i.e. one that does not change during the crowdsourcing process. This corresponds to the requirement 1 and examples of problems in such settings include crowdsourcing of the position of a radio mast (see Chapter 3 for more details) or the location of a person trapped in a damaged building. Now, as part of our requirement 2, we also need to extend our discussion to crowdsourcing problems in which the observed value changes dynamically as a function of the input. In particular, we focus crowdsourcing spatial functions that is relevant for a number of applications in disaster response including, for example, mapping the spreading of a contageous disease, estimating a temperature map, a weather map or a radioactivity map. In such settings,

untrustworthy observations are also problematic as they can lead to wrong predictions in the same way that we discussed for the stationary value case in Section 2.3.1. However, the spatial correlation in the data changes the inference significantly. To deal with this case, we now look at the class of learning techniques for spatial regression, particularly seeking flexible models of practical applicability to large-scale problems. While a number of regression techniques are available, ranging from least square regression and polynomial regression, to neural networks and kernel methods, there is typically a trade-off between the expressiveness of such models and their analytical tractability (Bishop, 2006). In this space, we identify Gaussian process regression as a rare exception of a model that is analytically tractable and at the same time a very flexible. This model is particularly suitable for our problem. Thus, we describe it in the next section.

2.4.1 Gaussian Process Spatial Regression

The Gaussian process (GP) is a Bayesian non-parametric model widely used for spatial and temporal regression in many real-world applications (Rasmussen, 2004). Specifically, in a spatial regression task, we are given a dataset of n geo-located observations of an unknown spatial function $f(\boldsymbol{x})$, where an observation normally consists of a pair of geographical locations (latitude and longitude) and output values, i.e. $\mathcal{D} = \{\boldsymbol{x}_i \in \mathcal{R}^2, y_i \in \mathcal{R} : i = 1, \ldots, n\}$, and the objective is to determine $f(\boldsymbol{x})$ from the data. Furthermore, Gaussian process regression assumes that the distribution of any subset of such observations is jointly Gaussian and that y_i is a noisy measurement, with zero-mean Gaussian noise, of the actual value of the function $\tilde{y}_i$ at the location $\boldsymbol{x}_i$. Formally:

$$y_i = \tilde{y}_i + \epsilon, \qquad \tilde{y}_i = f(\boldsymbol{x}_i), \qquad \epsilon \sim \mathcal{N}(0, \sigma_n) \tag{2.13}$$

In more detail, to make Bayesian inference in the function space, the GP model introduces a prior over the function f defined by a mean function $m(\boldsymbol{x}) = E[f(\boldsymbol{x})]$ and a covariance function $K(\boldsymbol{x}, \boldsymbol{x}') = cov(\boldsymbol{x}, \boldsymbol{x}')$. Thus, the GP is completely specified as:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x}')) \tag{2.14}$$

Specifically, the mean function represents the default value of f in the regions where no correlated training points are available (and it is often conventionally taken to be zero for notational simplicity). Instead, the covariance function is a crucial element in a GP predictor and needs to be appropriately chosen for a specific dataset. In particular, the covariance function is typically a stationary function that only depends on the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$, and it has some free parameters called the *hyperparameters*. For example, the squared-exponential is a stationary function generally used for a GP, with

two hyperparameters that are the signal variance σ_f and the length scale l. That is:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f \exp\Big(-\frac{1}{2l^2}(\boldsymbol{x} - \boldsymbol{x}')^2 \Big) \tag{2.15}$$

Then, if we wish to predict the value of f at a new test location $\boldsymbol{x}_*$ from the data, and let such a value be y_*, then, assuming that y and y_* are Gaussian random vectors, we can write the joint distribution at the test location as:

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N}\left(m(\boldsymbol{x}), \begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n & K(\boldsymbol{x}, \boldsymbol{x}_*) \\ K(\boldsymbol{x}_*, \boldsymbol{x}) & K(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{bmatrix} \right) \tag{2.16}$$

Now, conditioning y_* on y and using the marginalisation properties of the Gaussian distribution, we can derive the key equations of the predictive distribution for Gaussian process regression as follows:

$$p(y_*|\boldsymbol{x}, y, \boldsymbol{x}_*) = \mathcal{N}(E[y_*], \sigma^2(y_*)) \tag{2.17}$$

where

$$E[y_*] = m(\boldsymbol{x}) + K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2]^{-1} y \tag{2.18}$$

$$\sigma^2(y_*) = K(\boldsymbol{x}_*, \boldsymbol{x}_*) - K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2]^{-1} K(\boldsymbol{x}, \boldsymbol{x}_*) \tag{2.19}$$

Specifically, Equations 2.18 and 2.19 denote the predictive distribution of $f(\boldsymbol{x}_*)$ given by the predictive mean value and the predictive variance that is particularly suitable for the estimating the predictive uncertainty as outlined in requirement 2. Also, by integrating the likelihood, $p(y|\boldsymbol{x}) = \mathcal{N}(f, \sigma_n)$, over the GP prior (Equation 2.13), then we can obtain the analytical equation of *marginal likelihood* (typically expressed as the log-likelihood that is a monotonically increasing function and simplifies the notation) which is useful to train the hyperparameters (Rasmussen, 2004):

$$\begin{aligned} \log\, p(f|\boldsymbol{x}, \theta) = &-\frac{1}{2} y^T [K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n]^{-1} y \\ &-\frac{1}{2} \log|K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n| - \frac{n}{2} \log(2\pi) \end{aligned} \tag{2.20}$$

Specifically, such a GPR predictive distribution is derived with a process noise ϵ having a constant variance σ_n. That is, in practice, all the observations reported by the users have the same level of noise, which in statistics is referred to as *homoscedastic* regression (Silverman, 1985). However, as part of requirement 4, we require that each observation has an independent noise level and, to address this, we now introduce the *heteroscedastic* variant of Gaussian process regression where the inputs have independent noise terms.

2.4.2 Input-Dependent Noise

In heteroscedastic regression, the noise of the process varies across the inputs and the model is formally described as follows:

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i \tag{2.21}$$

Then, if we assume that such noise terms are independent and normally distributed, we get the same model studied by Goldberg et al. (1997), that is:

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2(x_i)) \tag{2.22}$$

and defining $\Sigma_x = diag\{\sigma^2(x_i)\}$, then the predictive distribution for such a model is:

$$E[y_*] = m(\boldsymbol{x}) + K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x]^{-1} y \tag{2.23}$$

$$\sigma^2(y_*) = K(\boldsymbol{x}_*, \boldsymbol{x}_*) - K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x]^{-1} K(\boldsymbol{x}, \boldsymbol{x}_*) \tag{2.24}$$

In general, in heteroscedastic Gaussian process (HGP) regression, inference is more challenging because the function of the noise rates, σ^2, is typically unknown. Thus, a common approach is to use a second GP prior over the noise function σ^2, with the drawback that this makes the model analytically intractable. For this reason, a significant amount of research has focussed on approximating inference in HPG regression, particularly using Markov Chain Monte Carlo (MCMC) (Goldberg et al., 1997), EM-like procedures (Kersting et al., 2007) and variational Bayes approximation (Lázaro-Gredilla and Titsias, 2011). Still, for our problem, there are two key observations that greatly simplify our analysis. Firstly, the noise rates of the individual observations are known since they are part of the user's report (Requirement 5). Secondly, it is reasonable to assume that such noise rates are reported independently by the users, thus there is no correlation between the noise terms that needs to be smoothed with a second GP. Given these assumptions, the HGP model can be applied to our problem and its predictive distribution described by Equations 2.22 and 2.23 can be used for making predictions of the unknown spatial function. However, the limitation of the current model is that it does not consider the individual trustworthiness of the reports presenting the same issue discussed earlier for the CI method (Section 3.1.2). Therefore, in Chapter 5, we will detail our novel trust-based HGP the considers also the trustworthiness dimension of each input.

2.5 Summary

In this chapter we introduced the key notions within the literature for dealing with untrustworthy information in crowdsourcing situations. Specifically, we began by discussing the computational approaches to trust evaluation and we identified the consensus approach as a suitable basis for our problem. In particular, we highlighted its property of effectively dealing with redundant datasets that are typical of a crowdsourcing problem. However, the standard consensus methods need to be extended to be applicable to datasets of observations with reported uncertainties and, to this end, in Chapter 3 we will introduce a novel consensus technique that addresses this limitation.

Then, reviewing approaches to reliable crowdsourcing, we discarded the gold-based approach as it is based on gold standards (available ground truths) which are not part of our requirements. However, the machine learning approach was identified as more suitable and various learning models for crowdsourcing problems based on graphical models were discussed in Section 2.2.2. Unfortunately, none of these models fully meet our requirements as, in particular, they only deal with datasets of pointwise observations. To address this limitation, we sought alternative learning techniques for aggregating estimates within the sub-field of data fusion.

Reviewing data fusion models, we first highlighted the key differences between the traditional fusion approach for sensor data and the required fusion approach for human data, pointing out the fact that the former is not appropriately designed for modelling human-like sensor behaviour. Then, we discussed two standard techniques for the single-hypothesis and the multi-hypothesis fusion case, namely covariance intersection (CI) and covariance union (CU). In this context, the CI method is good basis for dealing with noise reduction in fused estimates, however it needs to be extended to consider the individual trustworthiness of a report to meet our requirement 3. Therefore, for our new trust-based fusion method in Chapter 3, we will design a variant of the CI method where the estimates are fused according to their level of trustworthiness. Furthermore, the CU method was identified as the conservative solution for a fusion problem for its property of being an aggregator with guaranteed consistency but also with low informativeness due to its inflated variance. Therefore, CU will only be referred to as a benchmark of conservative fusion in our approach.

Furthermore, the outlier detection approach was discussed as a possible methodology to identify untrustworthy reports and the LOF method of density-based outlier detection was introduced. This method is sentitive to the choice of the the outlier threshold t and the parameter k that defines the locality region of outlier search. Therefore, we will also use this method as a benchmark and compare it to our trustworthiness estimation approach in terms of performance and flexibility.

Subsequently, we described an algorithm derived by the model presented by Reece et al. (2009), for untrustworthy sensors in multi-sensor fusion. This algorithm was judged as a solution that could potentially meet our requirements, although its underlying model is natively defined for a sensor fusion problem rather than for a crowdsourcing one. Therefore, we intend to evaluate this algorithm in a crowdsourcing context by using it as the main benchmark for our approach.

Finally, we discussed the problem of dealing with crowdsourced spatial functions that is part of our requirement. To this end, we reviewed techniques for heteroscedastic spatial regression. In particular, we introduced Gaussian process spatial regression with input-dependent noise which was identified as a valid technique for our problem's requirements. However, its limitation is that it does not provide any support against untrustworthy observations. Thus, we will address this shortcoming by detailing a new trust-based HGP model for untrustworthy inputs in Chapter 4.

Chapter 3

A Trust-Based Fusion Model for Crowdsourced Location Reports

In this chapter, we present and evaluate our trust-based fusion model for crowdsourcing location reports which directly addresses the requirement of crowdsourcing a stationary value with untrustworthy information (requirement 1). Specifically, in this space, we look at the problem of crowdsourcing a location target which is relevant for a number of industry and disaster response applications including cell tower localisation and finding missing or trapped person. Specifically, in this problem, the objective is to localise a target placed at undisclosed location from a set of possibly untrustworthy reports.

To address this problem, Section 4.1 will define a crowd reporting model for GPS location estimates reported by multiple untrustworthy users. In particular, the salient feature of such a model is to deal with the unknown reliability of the reports through formally modelling the trustworthiness of the user, so addressing the requirement of considering trustworthiness in the crowdsourcing inference model (requirement 5). Subsequently, a user's trustworthiness can be incorporated into the data fusion method to estimate the predictive distribution of the target location.

In more detail, we will define a novel trust-based fusion rule which builds upon the covariance intersection rule discussed Section 2.3.1 extended to incorporate the knowledge of user trustworthiness in the fused estimate. Based on such a rule, we will then derive a maximum likelihood estimator for the trustworthiness parameters which also provides, in turns, an estimate of the fused output. Subsequently, as a key contribution of this work, Section 3.2 will detail an algorithm, called MaxTrust, that implements the numerical optimisation of the likelihood to jointly estimate the trustworthiness parameters, so fulfilling the requirement 5, and the target location, so fulfilling the requirement 1.

Then, Section 4.3 will empirically evaluate the performance of MaxTrust against a number of trust-based and non-trust methods that were discussed in the previous chapter. In

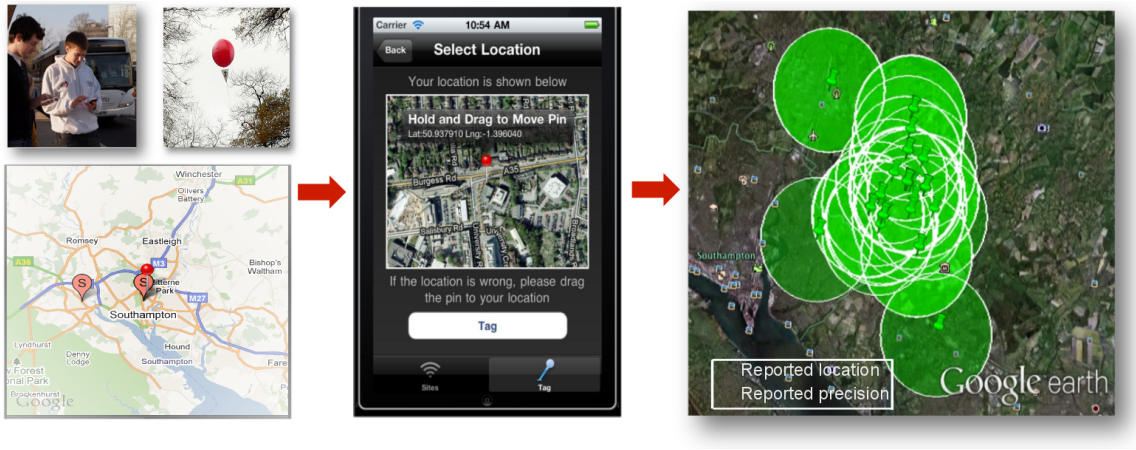


FIGURE 3.1: Illustration of a typical crowd reporting scenario with a user observing a target and reporting its observed location through its GPS-enabled smart phone.

particular, we will evaluate MaxTrust on synthetic data analysing the properties of the algorithm in various simulated crowd reporting settings. Then, we will show that MaxTrust can work also in real settings with an experiment on real-world data. Specifically, we identify the application of crowdsourced mast localisation as a suitable benchmark for a stationary value crowdsourcing problem, where the observed stationary value is the position of the mast that needs to be determined from multiple crowdsourced cell detections. This experiment involves a real crowdsourced dataset of cell detections provided by the OpenSignalMaps project. Finally, Section 4.4 concludes the chapter with a summary of the results.

3.1 Model Description

In this section, we formally describe our crowd reporting model (Section 3.1.1). Then, we detail the procedure for computing the fused estimate (Section 3.1.2) and estimating the trustworthiness parameters (Section 3.1.3).

3.1.1 A Trust Model for Crowdsourced Location Reports

In this model, a crowd of n users $U = \{1, \dots n\}$ observe a target placed at an undisclosed location $\boldsymbol{x_0}, \in \mathcal{R}^2$, where a location is specified by the pair of the geographical latitude and longitude value (although an analogous model description could be provided for an observed d-dimensional target). Each user i reports *one* location estimate $\boldsymbol{r}_i$ of the target obtained through the GPS sensor of its phone, with such an estimate comprising the following two values: (i) the GPS position $\boldsymbol{x_i} \in \mathcal{R}^2$ and (ii) the *precision* of the GPS fix: $\theta_i \in [LB, UB]$ where $0 < LB < UB < +\infty$, with θ_i that is automatically provided by the GPS receiver itself (estimated on the basis of the number and geometry

of satellites being used to generate the fix, see Android (`developer.android.com`) and Apple (`developer.apple.com`) API for more details). In this case, each report $\boldsymbol{r}_i = \langle \boldsymbol{x_i}, \theta_i \rangle$ means that user i estimates $\boldsymbol{x_0}$ as $\boldsymbol{x_i}$ with precision θ_i. and the objective is to estimate $\boldsymbol{x}_0$ from the set of reports. For example, Figure 3.1 illustrates a typical scenario described by our model. Specifically, there is a crowd that observes a specific target, that is this case is represented by a "red ballon" inspired the red ballon DARPA challenge.[1] Each user reports GPS estimates of the balloon location using a mobile app designed for submitting such reports. Then, the collected reports can be represented on map as estimates (circles) of where the balloon could be located based on the reported GPS location and the GPS precision.

Formally, we represent the uncertainty of a user's reported location as a probability density function (PDF) over the two-dimensional search space. Specifically, given $\boldsymbol{r_i}$, we assume that the probability density at a generic point $\boldsymbol{x} \in \mathcal{R}^2$ is normally distributed:

$$p(\boldsymbol{x}|\boldsymbol{r_i}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{x_i}, \theta_i \mathrm{I}) = \sqrt{\frac{\theta_i}{2\pi}} \exp\left(-\frac{\theta_i ||\boldsymbol{x} - \boldsymbol{x_i}||^2}{2}\right) \tag{3.1}$$

where the precision matrix $\theta_i \mathrm{I}$, with $\mathrm{I} = 2 \times 2$, denotes an uncorrelated and equally distributed variance along the two latitude and longitude dimensions. In statistics, this setting is also referred to as *heteroscedatic* inference where a collection of random variables has different variabilities quantified by the individual noise terms (see Section 2.5 for more details).

Then, we consider each user i as having an individual level of trustworthiness determined by the quality of its estimates. More formally, we assume that a report $\boldsymbol{r_i}$ is trustworthy with respect to $\boldsymbol{x_0}$ if the following condition holds:

$$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{x}|\boldsymbol{x_0}, \theta_i \mathrm{I}) \tag{3.2}$$

That is, $\boldsymbol{x_i}$ is drawn from a normal distribution centred on the actual target location with noise proportional to the user's precision. Conversely, $\boldsymbol{r_i}$ is untrustworthy with respect to $\boldsymbol{x_0}$ if $\boldsymbol{x_i}$ is drawn from any other statistics that differs from the one above. For example, a typical case of an untrustworthy report is when $\boldsymbol{x_i}$ is drawn from a biased distribution $\mathcal{N}(\boldsymbol{x}|\boldsymbol{x_0} \pm b, \theta_i \mathrm{I})$ where the mean is shifted from $\boldsymbol{x_0}$ with a random bias b.

To capture such user trustworthiness, we introduce a set of *trustworthiness parameters* $\boldsymbol{t} = \langle t_i \dots t_n \rangle^T$ where $t_i \in [0, 1]$ denotes the trustworthiness of user i (1 if i is fully trustworthy, 0 if i is untrustworthy). Accordingly, we express the new probability density

[1] This challenge aimed to find 10 red balloons in the US using crowdsourcing and social media and it was won by a MIT team in less than nine hours (`archive.darpa.mil`).

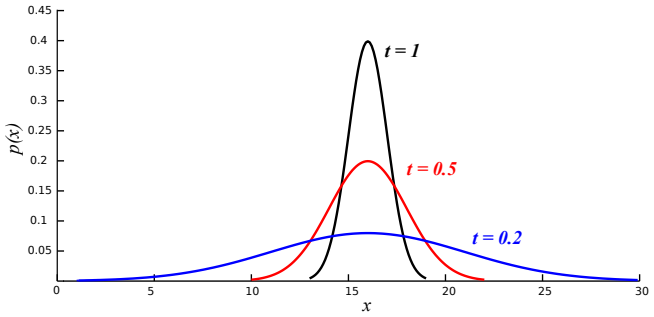


FIGURE 3.2: Effect of the trustworthiness parameter on a Gaussian estimate.

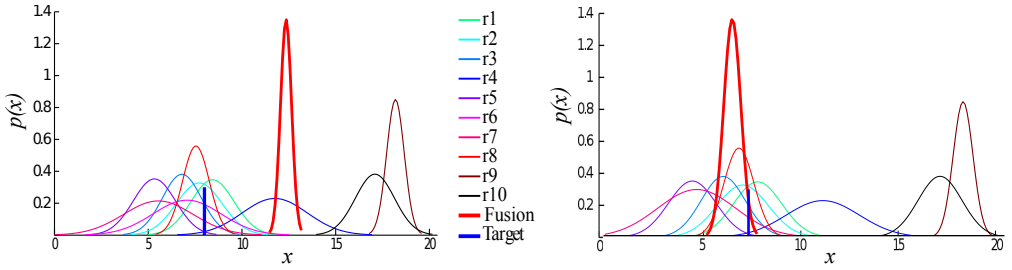
of an untrustworthy estimate $\boldsymbol{r_i}$ as a Gaussian PDF defined as follows:

$$p(\boldsymbol{x}|\boldsymbol{r_i}, t_i) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{x_i}, t_i\theta_i \mathrm{I}) = \sqrt{\frac{t_i\theta_i}{2\pi}} \exp\left(-\frac{t_i\theta_i||\boldsymbol{x} - \boldsymbol{x_i}||^2}{2}\right) \tag{3.3}$$

That is, similarly to Snelson and Ghahramani (2006), we use t_i as a *scaling parameter* of the precision of the Gaussian estimate. In such a way, the uncertainty of an estimate is regulated by the trustworthiness parameter, i.e. if a report is fully trustworthy ($t_i = 1$) then the uncertainty of its estimate is equal to the reported precision. Otherwise, if a report is untrustworthy ($t_i \ll 1$) then the uncertainty will increase to the extent of having an approximately uniform density across any $\boldsymbol{x} \in \mathcal{R}^2$ as t_i tends to 0. In more detail, such a scaling effect of uncertainty is shown in Figure 3.2 where a one-dimensional Gaussian estimate, $\boldsymbol{r} = \langle 16, 3 \rangle$ is plotted for different values of trustworthiness, $t_i = \{1, 0.5, 0.2\}$. From this, we can see that the PDF flattens on the x-axis as a consequence of inflating its variance proportionally to t_i thus producing the effect of de-emphasizing an untrustworthy estimate that we require.

3.1.2 Trust-based Fusion Model

As our ultimate objective is to convert the crowd's reports into a global estimate of the observed target location, we now require a method to fuse the reports that takes into account the individual level of trustworthiness of each participant (requirement 5). To this end, drawing from the realm of the data fusion methods discussed in Section 2.5, we consider this problem as a single-hypothesis fusion problem that we detailed in Section 2.5. This assumption is motivated by the fact that since the observed target is stationary then all the reports are assumed to refer to only one hypothesis of its correct value. As such, we use the covariance intersection (CI) rule, that is the standard method for single-hypothesis fusion problems, as the baseline technique for computing the aggregated output in our model. Specifically, given a set of untrustworthy reports $R = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\}$, with associated trustworthiness $\boldsymbol{t}$, then their CI fusion is a new PDF



(a) Standard CI fusion (b) Trust-based fusion

FIGURE 3.3: Example of 10 Gaussians fused through the standard fusion (a) and trust-based fusion (b).

$f(\boldsymbol{x}|R, \boldsymbol{t})$, formally defined as follows:

$$f(\boldsymbol{x}|R, \boldsymbol{t}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{x_f}, \theta_f \mathrm{I}) \tag{3.4}$$

$$\theta_f = \boldsymbol{t}^T \boldsymbol{\theta} \tag{3.5}$$

$$\boldsymbol{x_f} = \theta_f^{-1}(\boldsymbol{t} X^T \boldsymbol{\theta}) \tag{3.6}$$

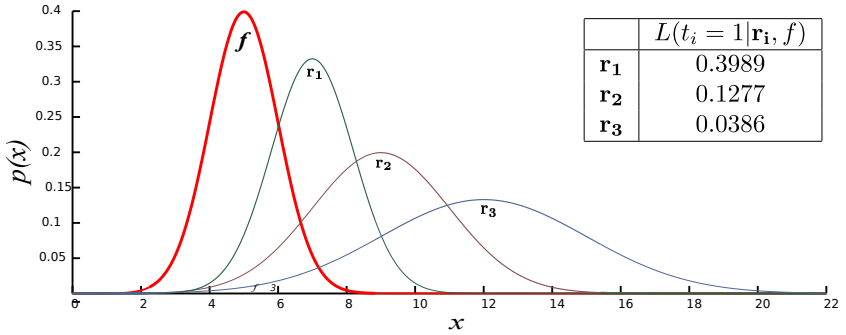
where X is the matrix with $\boldsymbol{x_i}$ as row vectors and $\boldsymbol{\theta} = \langle \theta_1, \dots \theta_n \rangle^T$ is the precision vector. For the two-dimensional case, the equations above can also be rewritten as follows:

$$\theta_f = t_1\theta_1 + \cdots + t_n\theta_n \tag{3.7}$$

$$x_{f,1} = \theta_f^{-1}(x_{1,1}t_1\theta_1 + \cdots + x_{n,1}t_n\theta_n) \tag{3.8}$$

$$x_{f,2} = \theta_f^{-1}(x_{1,2}t_1\theta_1 + \cdots + x_{n,2}t_n\theta_n) \tag{3.9}$$

where $\boldsymbol{x_i} = \langle x_{i,1}, x_{i,2} \rangle$ and $\boldsymbol{x_f} = \langle x_{f,1}, x_{f,2} \rangle$. Specifically, CI applied to our trust model defines a new trust-based fusion model in which each estimate is fused as jointly weighted by θ_i and by t_i. This determines that more trustworthy reports are considered with a higher degree, while the untrustworthy ones are gradually downgraded in the fusion. In so doing, such trust-based fusion incorporates the knowledge of report trustworthiness in the fused output and deviates from the traditional CI fusion which considers all the estimates as equally trustworthy. Furthermore, comparing these two fusion approaches, Figure 3.3 shows an example of a set of 10 one-dimensional Gaussians, referring to a target placed in position $x_0 = 8$ (blue bar), that are fused through CI (Figure 3.3, (a)) and through the trust-based fusion described above (Figure 3.3 (b)). Specifically, in this example the trustworthiness values are set to $t_{1:7} = 1$ and $t_{8:10} = 0$ to denote that the last two estimates are believed to be untrustworthy. As an effect, it can be noticed that the trust-based estimate is much closer to $\boldsymbol{x_0}$ compared to the estimate of the non-trust method as the former assigns lower weights to the reports $\boldsymbol{r}_9$ and $\boldsymbol{r}_{10}$ that are indeed inconsistent with $\boldsymbol{x_0}$. More generally, our trust-based fusion model is designed in such a way that its estimation accuracy is related to the correct assignment of the trustworthiness parameters.



	$L(t_i = 1\|\mathbf{r_i}, f)$
$\mathbf{r_1}$	0.3989
$\mathbf{r_2}$	0.1277
$\mathbf{r_3}$	0.0386

FIGURE 3.4: Likelihood of three reports r_1, r_2, r_3 over the fused estimate f.

3.1.3 A Maximum Likelihood Trustworthiness Estimator

We now describe a computational approach to estimate the trustworthiness parameters of each user using the maximum likelihood estimator. In particular, maximum likelihood is the widely used frequentist estimation approach which sets the parameters in statistical model to the values that maximise the probability of the observed dataset (likelihood) (Bishop, 2006). Therefore, we start by defining the likelihood of the trustworthiness of a single report as follows. For each report $\boldsymbol{r_i}$, considering for notational simplicity that the two-dimensional vectors can also be written as $\boldsymbol{x} = \langle x_1, x_2 \rangle$, $\boldsymbol{x_i} = \langle x_{i,1}, x_{i,2} \rangle$ and $\boldsymbol{x_f} = \langle x_{f,1}, x_{f,2} \rangle$ respectively, then the likelihood of t_i given $\boldsymbol{r_i}$ and $f(\boldsymbol{x}|R, \boldsymbol{t})$ is the joint product of the two PDFs described in Equation 3.3 and 3.4, integrated over the two-dimensional space. Formally:

$$\begin{aligned} L(t_i|\boldsymbol{r_i}, f) &= \int_{R^2} p(\boldsymbol{x}|\boldsymbol{r_i}, t_i) f(\boldsymbol{x}|R, \boldsymbol{t}) dx \\ &= \int_{x_1} \int_{x_2} \frac{t_i \theta_i \theta_f}{4\pi^2} \exp\Bigg(-\frac{1}{2}(t_i\theta_i(x_1 - x_{i,2})^2 \\ &\quad + t_i\theta_i(x_2 - x_{i,2})^2 + \theta_f(x_1 - x_{f,1})^2 \\ &\quad + \theta_f(x_2 - x_{f,2})^2 \Bigg) dx_1 dx_2 \end{aligned} \tag{3.10}$$

Then, applying basic rules of Gaussian integration, Equation 3.4 can be solved in closed form as follows:

$$\begin{aligned} L(t_i|\boldsymbol{r_i}, f) = &\frac{1}{2\pi(\frac{1}{t_i\theta_i} + \frac{1}{\theta_f})} \exp\Bigg(-\frac{t_i\theta_i}{2}(x_{i,1} + x_{i,2})^2 \\ &+ \frac{(t_i\theta_i x_{i,1} + \theta_f x_{f,1})^2 + (t_i\theta_i x_{i,2} + \theta_f x_{f,2})^2}{2(t_i\theta_i + \theta_f)} \\ &- \frac{\theta_f}{2}(x_{f,1} + y_{f,2})^2 \Bigg) \end{aligned} \tag{3.11}$$

That is, such a likelihood of t_i is taken as the product of the probabilities assigned by $\boldsymbol{r_i}$ and f to the area of $\Delta\boldsymbol{x}$, then taking the limit $\Delta\boldsymbol{x} \to 0$, and summing up for each

possible $\Delta\boldsymbol{x}$, this gives the integral over $\boldsymbol{x}$. Such an integral is equal to the exponential of the pairwise distance between $\boldsymbol{x}$ and $\boldsymbol{x_i}$ and between $\boldsymbol{x_i}$ and $\boldsymbol{x_f}$, scaled by $t_i\theta_i$ and θ_f respectively. In more detail, a numerical example of computing the likelihood of a report is provided in Figure 3.4, where the likelihood of $t_i = 1$ is computed for three reports, $\boldsymbol{r}_1 = \langle 7, 0.7\rangle, \boldsymbol{r}_2 = \langle 9, 0.25\rangle, \boldsymbol{r}_3 = \langle 12, 0.11\rangle$ and $f = \langle 5, 1\rangle$. In particular, it can be seen that the likelihood value is proportional to the area shared between $\boldsymbol{r_i}$ and f i.e. the further is, $\boldsymbol{r_i}$ from f, the lower is its likelihood and, in this example $\boldsymbol{r_1}$ and $\boldsymbol{r_3}$ are the most and least likely estimate respectively, given f.

Next, assuming independence between t_i and t_j for any $i \neq j$, equivalent to saying that each user's trustworthiness is independent from any other user, the global likelihood of $\boldsymbol{t}$ given R is the product of the individual likelihood terms. Formally:

$$\begin{aligned} L(\boldsymbol{t}|R) &= \prod_{i=1}^{n} L(t_i|\boldsymbol{r_i}, f) \\ &= \prod_{i=1}^{n} \Big(\int_{R^2} p(\boldsymbol{x}|\boldsymbol{r_i}, t_i) f(\boldsymbol{x}|R, \boldsymbol{t}) dx \Big) \end{aligned} \tag{3.12}$$

Notice that the function above does not directly depend on f as this is implicitly derived from R and $\boldsymbol{t}$ which are already function parameters (see Equation 3.4). Then, by taking the log-likelihood of Equation 3.12 we obtain the following expression:

$$\begin{aligned} \ln\big(L(\boldsymbol{t}|R)\big) &= \sum_{i=1}^{n} \ln\big(L(t_i|\boldsymbol{r_i}, f)\big) \\ &= -\, n\ln(2\pi) + \sum_{i=1}^{k} \Bigg(\ln(t_i\theta_i + \theta_f) + \ln(t_i\theta_i\theta_f) \\ &\quad + \frac{(x_{i,1}t_i\theta_i + x_{f,1}\theta_f)^2 + (x_{i,2}t_i\theta_i + x_{f,2}\theta_f)^2}{2(t_i\theta_i + \theta_f)} \\ &\quad - \frac{t_i\theta_i}{2}(x_{i,1} + x_{i,2})^2 - \frac{\theta_f}{2}(x_{f,1} + x_{f,2})^2 \Bigg) \end{aligned} \tag{3.13}$$

Finally, factoring in the expressions of θ_f and $\boldsymbol{x}_f$ (Equations 3.7, 3.8 and 3.9), then we get the final likelihood function (omitted here for simplicity) that we maximise to find the maximum likelihood values of the trustworthiness parameters, formally:

$$\boldsymbol{t_{ML}} = \arg\max_{\boldsymbol{t}} \sum_{i=1}^{n} \ln\big(L(t_i|\boldsymbol{r_i}, f)\big) \tag{3.14}$$

In so doing, we notice that there are two singularities in the function for $t_i = -\theta_f/\theta_i$ and $t_i = 0$ (see Equation 3.13). We discuss these two cases individually. Specifically, the case of $t_i = -\theta_f/\theta_i$ is excluded by our initial assumptions of having θ_i and t_i positively defined (see, Section 3.1.1). The case of $t_i = 0$ implies that the trustworthiness value of any report cannot be set to zero otherwise this would give an infinite variance which

is not tractable numerically. To avoid this, we set the range of t_i to be open in 0, i.e. $t_i \in (0, 1]$, thus approximating the value of an untrustworthy report with a small value ϵ, i.e. $t_i \in [\epsilon, 1]$. Having now described our model formally, an algorithm for computationally optimising the likelihood to estimate the parameters is provided in the next section.

3.2 The MaxTrust Algorithm

This section describes the algorithm, called MaxTrust, for computing the trustworthiness parameters and the fused estimate in our model implementing the maximal likelihood estimator formally described in the previous section. Such an algorithm is designed in a way to trade-off a good quality in the numerical approximation with polynomial complexity.

Before going into further detail, however, we discuss the following two computational issues concerning the analysis of our model. First, the non-linear expression of the likelihood described in Equation 3.13 is not tractable analytically. Thus, we need to use numerical optimisation to carry out such a function maximisation. Second, there is a mutual dependency between the trustworthiness parameters, i.e. when we update t_i also we do need to update the remaining $t_{(-i)}$ parameters. Thus, a natural way to solve this computationally is to iteratively set values of each of the t_i parameters until these converge. Specifically, to do this, we use the *Jacobi* iteration (Hageman and Young, 2004), which is a standard numerical technique for solving non-linear systems that sequentially update only one system parameter at a time using the values of the previous iteration[2]. Drawing these two points together, our MaxTrust algorithm can now be described (see Algorithm 3.1).

In more detail, in step 1, the algorithm initially sets t_i uniformly to 1, i.e. it is conservative in considering all reports as trustworthy in the first iteration. Then, steps 3-6 implement the Jacobi iteration where, at the h-th iteration, $t_i^{(h)}$ is updated through the pointwise maximisation of f with only t_i left as a free parameter using the values of $t_{-i}^{(h-1)}$ from the previous iteration (step 5). After convergence is achieved, and such a convergence was empirically found to be reached in approximately 5 - 20 iterations, then the algorithm returns the trustworthiness values $\boldsymbol{t}^{(h)}$ and the fusion parameters $\langle \boldsymbol{x_f}, \theta_f \rangle$ from the last iteration (step 7-8). Thus, MaxTrust computes the output in $O(|S||T|)$ polynomial time, where $|T|$ is the number of trustworthiness parameters and $|S|$ is the number of samples used to perform the pointwise function maximisation in step 5. In contrast, an exhaustive numerical function maximisation would require $O(|S|^{|T|})$ that is exponential time to try all the possible combinations of the $|S|$ samples for each of the

[2]The dual method, the *Gauss-Seidel* iteration (Black and Moore, 2006), is also suitable although this was found to be less numerically stable in our experiments.

Algorithm 3 MaxTrust

Variables :

R : Report set.
$\boldsymbol{t}^{(h)}$: Trustworthiness vector at the h-th learning epoch.
f : Fused estimate.
err : Error bound.
$epochs$: Maximum number of learning epochs.

Algorithm *MaxTrust*(R)

1: Start with uniform prior trustworthiness:
$\boldsymbol{t}^{(0)} := \langle 1, \ldots, 1 \rangle$
2: $h := 0$
3: **while** ($|t^{(h-1)} - t^{(h)}| < err$ **and** $h < epochs$) **do**
4: $h := h + 1$
5: **for** $i := 1 : k$ **do**
$t_i^{(h)} := \arg\max_{\boldsymbol{t}} L(\langle t, t_{-i}^{(h-1)} \rangle | R)$ (Eq. 3.13)
end for
6: **end while**
7: $\theta_f := (\boldsymbol{t}^{(h)})^T \boldsymbol{\theta}$,
$\boldsymbol{x_f} := \theta_f^{-1} (\boldsymbol{t}^{(h)} X^T \boldsymbol{\theta})$ (Eq. 3.4 - 3.6)
8: **return** $(\boldsymbol{t}^{(h)}, \boldsymbol{x_f}, \theta_f)$

t_i parameters. From this, the saving in complexity of MaxTrust in computing such a function optimisation is readily apparent.

Having now described our algorithm, its empirical evaluation is detailed in the next section.

3.3 Experimental Evaluation

In this section, we present the results of the evaluation of MaxTrust on a target localisation problem with both synthetic and real-world data. Specifically, the first experiment on synthetic data aims to analyse the properties of the algorithm in various simulated crowd reporting settings (Section 3.3.1). Then, the second experiment shows the effectiveness of MaxTrust on the real-world crowdsourcing application of cell tower localisation (Section 3.3.2).

3.3.1 Experiment on Synthetic Data

In this first experiment we evaluate MaxTrust on a target localisation problem using synthetic, one-dimensional data. Specifically, the experimental setting is as follows. A target is placed in position $x_0 = 8$ and 50 univariate Gaussians are randomly generated by first sampling $\theta_i \sim U[0.2, 10]$ and then sampling $x_i \sim \mathcal{N}(x_0, \theta_i)$, particularly

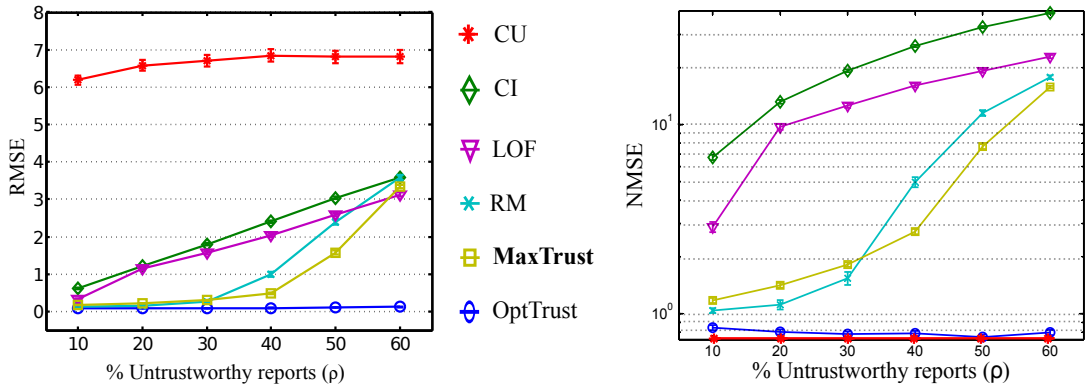
following our model assumptions made in Section 4.1. Furthermore, a percentage ρ of such estimates is untrustworthy. Specifically, an untrustworthy estimate is generated by adding a random noise w to x_i as follows:

$$\hat{x_i} = x_i + w \quad \text{where} \quad w = \begin{cases} \sim U[2, 10] & \text{with probability 0.5 for each run} \\ \sim U[-10, -2] & \text{otherwise} \end{cases} \tag{3.15}$$

In particular, as Equation 3.15 states, for a single run of the simulation, w is randomly sampled from either a positive or a negative range. We do so as to get an unbiased setting for testing the algorithm. In fact, by sampling w uniformly from $[2, 10]$, we avoid the situation in which two untrustworthy estimates could be symmetric, with $+w$ and $-w$ respectively, and so balance their noise in the linear fusion. However, in so doing, our results would be susceptible to any algorithm that biases the result in one particular direction. To rectify this, we alternate sampling with probability 0.5 between the intervals $[2, 10]$ and $[-10, -2]$ for each run. In this way, we avoid the symmetric noise case but we also consider positive and negative noises.

Given this setting, we benchmark MaxTrust against two classes of fusion algorithms that are described as follows:

- **Non-Trust Fusion Algorithms**: These are the fusion algorithms that do not explicitly consider report trustworthiness. In this class, we consider the following three algorithms that are representative of the fusion approaches discussed in Chapter 2:
  - **CI**: The standard CI fusion rule as described in Section 2.2.
  - **CU**: The covariance union rule (CU) that merges the estimates by taking the union of their covariances as described in Section 2.2.
  - **Local Outlier Factor Fusion (LOF)**: This algorithm is based on the density-based outlier detection method that was described in Section 2.2. Specifically, the algorithm first removes outliers identified as the estimates with LOF greater than 1 (using $k = 5$ as the number of nearest neighbours), and then applies CI to fuse the remaining inliers.
- **Trust-Based Fusion Algorithms**: These are algorithms that consider a report's trustworthiness in their fusion process similarly to our approach. Specifically, in this class, we consider the following two algorithms:
  - **Optimal CI (OptTrust)**: This is a *hypothetical* optimal algorithm with full knowledge of the trustworthiness value of the single report. That is, the subset of the untrustworthy reports within R is known and is assigned with zero trustworthiness. As such, this algorithm represents the optimal performance for a fusion algorithm in our model.



(a) (b)

FIGURE 3.5: Plot of the RMSE and the NMSE for the six algorithms against increasing untrustworthiness.

– **Reece Method (RM)**: This is the algorithm presented by Reece et al. (2009) for fusing sensor data using sensor trustworthiness described in Algorithm 2.1. In particular, we run the algorithm setting $\beta = 3$ as the authors suggest in the paper.

In summary, a set of six algorithms {CI, CU, LOF, OptTrust, RM, MaxTrust} were tested, representing the non-trust versus the trust approach. The accuracy of a predictor over the N simulations was measured as the root mean square error (RSME) with respect to x_0. That is:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_f^i - x_0)^2} \tag{3.16}$$

where $\langle x_f^i, \theta_f^i \rangle$ are the parameters of the predictive estimate produced at the i-th run. We also consider the normalised mean square error (NMSE) as a more comprehensive accuracy measurement that considers also the uncertainty of a prediction. That is:

$$NMSE = \frac{1}{N}\sum_{i=1}^{N}\theta_f^i(x_f^i - x_0)^2 \tag{3.17}$$

Finally, for RM and MaxTrust, we are also interested in measuring the error of their trustworthiness estimates to evaluate the quality of their learning. To this end, we consider the RMSE of the trustworthiness parameters defined given by:

$$t\_error = \sqrt{\frac{1}{|R|}\sum_{i=1}^{|R|}(\hat{t}_i - t_i)^2} \tag{3.18}$$

where $\hat{t}_i$ is the true value of trustworthiness assigned by the simulator to $\boldsymbol{r}_i$. The results of this experiment are presented in the next section.

Results

Figure 3.5 shows the results obtained by each algorithm for $N = 600$ runs with a percentage of untrustworthy reports increasingly set to $\rho = \{10, 20, 30, 40, 50, 60\}$ (the error bars are invisible due to their very small values). Specifically, Figure 3.5 (a)) shows that, as expected, the estimation error of all the algorithms increases as ρ increases. However, OptTrust constantly keeps its error to approximately zero due to the prior knowledge of the untrustworthy estimates. Amongst the other methods, CU constantly gets the highest error because of its property of unify the estimates under the most general output which typically estimates the target inaccurately with respect to x_0. LOF's error marginally improves over CI as an effect of removing outliers from the report set, while CI always considers all the reports as inliers. Then, for $\rho = 60\%$ every algorithm achieves a comparable error due to the majority of the reports no longer converging to the ground truth.

Crucially, we notice that the trust-based fusion algorithms (OptTrust, MaxTrust and RM) do significantly better than the non-trust algorithms (CI, CU). Moreover, among these, MaxTrust gets the lowest error and it gains up to 51% compared to RM when $\rho = 40\%$ (a paired-sample t-test revealed that this result is statistically significant at the 0.01 level, $t(289.71) = 0.183$, $p = 1.6 \cdot 10^{-3}$). The performance of these two algorithms is explained by their trustworthiness estimation error that is reported in Table 3.1. From this, we can see that MaxTrust's t_error is 59% lower than RM for $\hat{t}_i = 1$ (0.31 vs 0.67) and is comparable for $\hat{t}_i = 0$, (this result was also found to be statistically significant by a paired t-test at the 0.01 level, $t(71.48) = 5.12, p = 2.3 \cdot 10^{-6}$).

Another interesting result is the NMSE showed in Figure 3.5 (b) that highlights the consistency of the estimates of each algorithm. Here, we can see that the normalised error of MaxTrust, as well as the one of RM, is the closest to the optimum (OptTrust) with a value of approximately 1 for $\rho < 30\%$ that means that the actual target is only one standard deviation away from the estimated point. Furthermore, the only exception to the results of these two method is CU whose error is always zero due its high uncertainty. Thus, this shows that our algorithm is not only accurate, but it is also very informative with a low predictive uncertainty.

From these results, we contend that the trust-based approach significantly improves the accuracy of localising the target based on synthetic data. In particular, MaxTrust is the algorithm with the best trade-off of good accuracy and low uncertainty amongst the tested methods, and with the additional property of yielding the most accurate learning of trustworthiness. As the next step, we explore the effectiveness of MaxTrust on a real-world dataset.

	RM	**MaxTrust**
t_error for trustworthy estimates ($\hat{t}_i = 1$)	0.67 ± 0.15	0.31 ± 0.10
t_error for untrustworthy estimates ($\hat{t}_i = 0$)	0.26 ± 0.20	0.24 ± 0.11
Average	0.46 ± 0.27	0.28 ± 0.11

TABLE 3.1: The trustworthiness estimation error (*t_error*) for RM and MaxTrust.

3.3.2 Experiment on Real-World Data

In this second experiment, we focus on the problem of localising cellular masts from crowdsourced cell detections. This is a key application for the mobile phone industry (Ahern et al., 2006). In fact, all the major phone manufacturers, including Apple, Google and Nokia, are engaged in the effort of mapping cellular masts to improve the positioning system of their mobile phones. Specifically, by having a map of the masts located in the phone's local area, then triangulation can rapidly give an accurate phone position with minimal battery drain. In this way, phones would no longer be constrained to use the GPS, which is slow (up to 3 minutes to get the signal) and has a high battery depletion. Moreover, cell towers positioning would allow them to localise themselves also in indoor environments.

However, the task of mapping the masts cannot be easily achieved manually since the topology of cellular networks may change frequently and the network operators do not always make available the map of their installed masts in every country. For this reason, a number of projects have recently explored a crowdsourcing approach to this problem consisting of leveraging the multitude of mobile phones disseminated across the cell to collect data about mast locations.[3] Specifically, GPS-equipped phones can provide the list of the masts scanned in their surrounding area together with the phone's current GPS position. Then, the mast location can be determined through merging multiple cell detections reported by the phones from different positions.

However, in so doing, an important issue to consider is the untrustworthiness of some of the reported cell detections. Specifically, GPS readings are often inaccurate, mainly because of the limited update frequency of the device that often returns out-of-date locations. Also, the signal strength read by the phone does not always accurately indicate the current phone-mast distance as the signal may change dynamically across the cell due to obstacles and reflections. As such, inaccuracies are an issue to reliably localise the masts. We now show how the MaxTrust algorithm can be applied to this problem to improve localisation accuracy through estimating the trustworthiness of a reported detection. Specifically, we focus on the case of an omni-directional cell tower network illustrated in Figure 3.6, namely where a mast is placed at the centre of each hexagonal cell. We do so as in such network topologies the probability of cell detections is spherically uniformly distributed since, the mast radiates the signal spherically across the cell.

[3]For examples, see `cellmapper.net`, `epitiro.com` and `skyhookwireless.com`

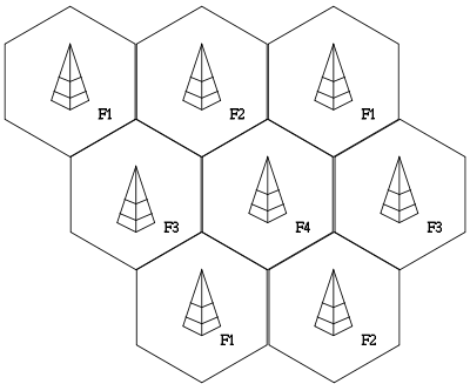


FIGURE 3.6: Topology of a cellular network with omni-directional radio masts.

Thus, this case better represents the assumption of normally distributed probability of the mast location given an observation made by our model in Section 3.1.1.

Dataset

In this experiment, we used the dataset provided by the OpenSignalMaps project (`opensignalmaps.com`) that includes 1563 records of anonymised mobile phones that reported detections for a set of 129 omni-directional cellular masts (max=46, min=6, avg=12 reports per masts) in the area of Southampton, UK (bounding box: 50.97 N, 1.525 W and 50.85 N, 1.25 W). Specifically, each report comprises (i) the Cell ID (CID) and Location Area Code (LAC) of the phone's cell, (ii) the GPS location of the phone (latitude and longitude degrees), (iii) the precision of the GPS fix (in meters). However, for privacy issues, the dataset did not provide any user identifier that could link between the single user and its multiple reports. Therefore, in this experiment, we can only consider the *single-reporting case* in which each user is assumed to report only one cell detection, A complete description of the dataset is provided in Appendix A. Furthermore, a second official dataset of mast locations is made available by the Authority of UK Communication (OFCOM, `ofcom.org.uk`) which we consider as a more reliable dataset for this experiment. Therefore, we will refer to the OFFCOM dataset as the ground truth for evaluating our algorithm. Then, to define a setting for such a dataset suitable for applying our model, each geographical position (latitude-longitude value) is converted into planar coordinates (in meters) applying the following projection:

$$R_{\text{Lat-Lon}} = \begin{pmatrix} lat \\ lon \end{pmatrix} \begin{matrix} (degrees) \\ (degrees) \end{matrix} \mapsto R_{\boldsymbol{x}} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \begin{matrix} (meters) \\ (meters) \end{matrix}$$

$$x_1 = 111,229 \cdot \cos(Lat) \cdot (lon - lon_0) \tag{3.19}$$

$$x_2 = 111,229 \cdot (lat - lat_0) \tag{3.20}$$

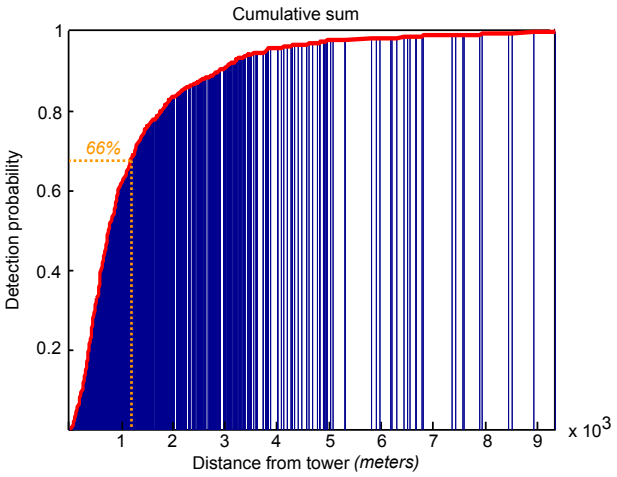


FIGURE 3.7: Cumulative distribution of cell detections according to the phone-tower distances.

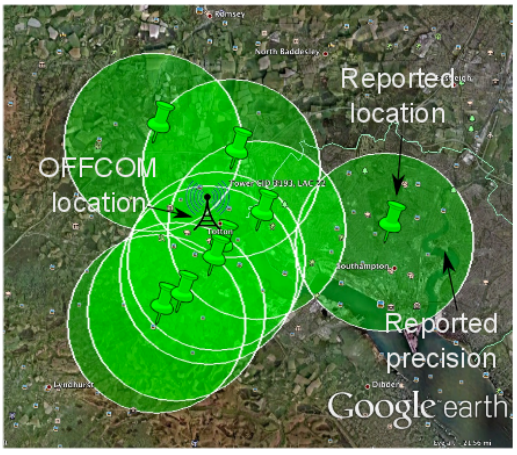


FIGURE 3.8: Screenshot of the reports for the cell tower (CID 3139, LAC 22) from the OpenSignalMap dataset.

where lat_0 and lon_0 are the coordinates of the point taken as the origin in the new system (conventionally set to 50.84 N, 1.52 E). In more detail, such a transformation approximates the more complex Haversine formula that provides the appropriate trigonometric treatment for spherical distances between two location points but is constrained for numerical computation (Kells et al., 1951). Instead, given that at 50N one degree of latitude corresponds to 111,229 meters, the projection described in Equations 3.19 and 3.20 is more efficient to compute the new position for each point with an approximation error that is small enough for the geographical area that we consider.

Then, to calculate θ_i, i.e. the precision of a reported detection, we notice that 66% of the readings were within 1200 meters from the ground truth tower location, as it is apparent from the cumulative distribution of the phone-mast distance showed in Figure 3.7. Hence, setting $\sigma_0 = 1200$, then θ_i is determined as follows:

$$\theta_i = (\sigma\_\text{GPS}_i^2 + \sigma_0^2)^{-1} \tag{3.21}$$

where $\sigma\_\text{GPS}_i^2$ is the inverse of the GPS precision reported by user i. Summing up, the estimate associated with each phone's report is represented as $\boldsymbol{r}_i = \langle x_{i,1}, x_{i,2}, \theta_i \rangle$, where $\langle x_{i,1}, x_{i,2} \rangle$ is the position of the user and θ_i is the precision of such a position, respectively. As an example, Figure 3.8 shows the reports collected for the cell (CID 3139, LAC 22) represented as a two-dimensional normal distribution according to the representation given above. Specifically, the centre of the circle corresponds to the expected location and the radius its the 99% confidence interval (i.e. $3/\theta_i$).

Similarly to our previous experiment, we measure the accuracy and the consistency of the algorithms as the RMSE and the NMSE respectively, formally:

$$\text{RMSE} = \sqrt{\frac{1}{|masts|} \sum_{i=1}^{masts} |\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i|^2} \quad \text{(meters)} \tag{3.22}$$

$$\text{NMSE} = \frac{1}{|masts|} \sum_{i=1}^{masts} \theta_i |\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i|^2 \tag{3.23}$$

Finally, to demonstrate that our algorithm outperforms other fusion methods, we compare it to the same set of benchmarks described in Section 3.3.1 (excluding OptTrust which is unfeasible in this case since there is no prior knowledge of report trustworthiness available), that are now adapted to process two-dimensional location data as it is the case of the OpenSignalMap dataset. The results of this experiment are described in the next section.

Results

The performance of the five algorithms, CI, CU, LOF, RM, MaxTrust, are reported in Table 3.2 in terms of the geographical distance (in meters) of the estimated mean from the ground truth. For brevity, the table reports only the results of 15 out of 129 randomly selected masts but the results for the other masts are similar as is apparent also from Figure 4.9 (a). In particular, we can see from the table that the estimates of the masts made by CI and LOF have an error, on average, of approximately 1250 meters from the actual tower location. In contrast, the trust-based methods, RM and MaxTrust, lower this error by at least 30% that corresponds to an accuracy increase by approximately 384 meters. In particular, MaxTrust is able to reduce such an error even further, 21% lower than RM, which is equivalent to an improvement of 200 meters compared to RM that is globally 600 meters more accurate than CI and LOF.

Furthermore, Figure 4.9 (a) shows the RMSE for the algorithms taken over the whole set of the 129 masts. From this, we can see that this error is in line with the results on the smaller set, with MaxtTrust outperforming the other methods by 42% (equivalent to 467 meters) compared to CI, and by 22% (equivalent to 185 meters) compared to RM. Interestingly, plotting the error for MaxTrust and CI over the number of reports available in each cell (Figure 3.10), we notice that MaxTrust minimises its error when the size of the report set is small (i.e. < 20 reports), while its error is comparable to CI for larger report sets. This is explained by the fact that, when there are sufficiently many reports in the cell, then there is likely to be a majority of trustworthy reports that mitigate the error of the untrustworthy ones. However, in cells where not many reports are available, than our algorithm can be more accurate in localising the mast. Finally,

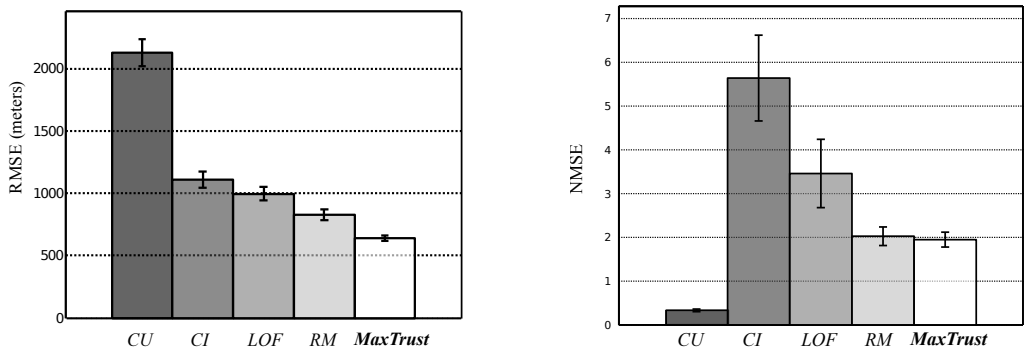


FIGURE 3.9: Bar plots of the RMSE (a) and NMSE (b) for the five algorithm in estimating the positions of the 129 masts.

as for the previous results, CU gets the highest error that means that the union estimate is typically distant from the real mast location.

A final interesting result is the consistency of algorithms, as measured by the NMSE in Figure 4.9 (b). As before, CU's error is minimal because of its very high predictive uncertainty. In particular, the trust-based algorithms, MaxTrust and RM, outperform the other two non-trust algorithms, CI and LOF, by up 50% compared to LOF. Specifically, likewise the results from the previous experiment (Section 3.3.1), their mean normalised error is approximately 2 that means that the location estimated by these two methods is only two standard deviations away from the actual tower location in the two-dimensional space. Thus, this confirms that our algorithm is not only accurate, but that it also has a low predictive uncertainty. From this, we conclude that the trust-based fusion performed by MaxTrust improves the accuracy in solving the mast localisation problem and additionally learns the trustworthiness of each detection. In particular, MaxTrust is able to localise the mast with an error that is an average of 181 meters lower than any other method.

3.4 Summary

In this chapter, we presented our trust-based fusion model for crowdsourced location reports as part of addressing requirement 1 for crowdsourcing a stationary location target. To address this problem, we introduced a trust model of a user reporting location estimates of an observed fixed target where the trustworthiness of a user is modelled as a scaling parameter of the reported uncertainty. Based on this, we described a fusion method for computing global estimates taking into account the individual levels of trustworthiness of the users.

Then, as part of the contributions of this work detailed in Section 1.3, we presented our MaxTrust algorithm that, given the report set, estimates the target location through

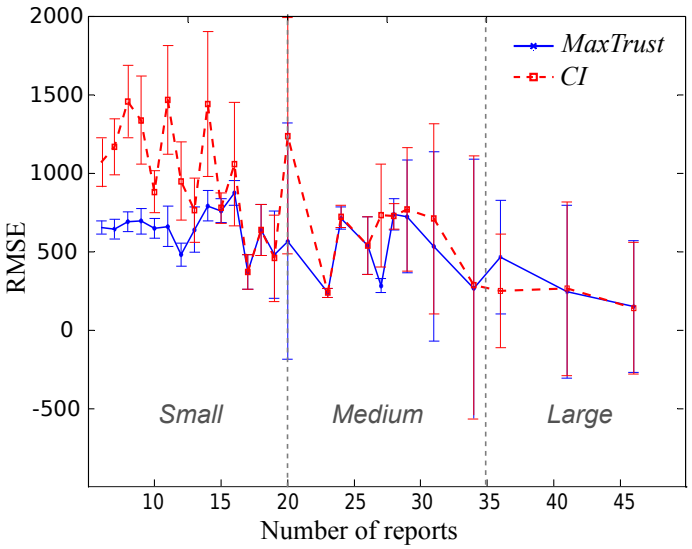


FIGURE 3.10: Error of CI and MaxTrust over the number of reports for each mast.

learning the trustworthiness of each user using the maximum likelihood estimator. In particular, we showed that MaxTrust outperforms five of the state-of-the-art methods in accuracy and predictive uncertainty in estimating the target. In more detail, empirical results on synthetic data showed that our algorithm predicts the target more accurately, by up to 51% and with lower uncertainty by up to 80%. Furthermore, we showed that our algorithm can also work in real settings with an experiment on crowdsourced cell tower localisation using real-world data. In this experiment, MaxTrust improved the precision of predicting the cell tower location by up 21% compared to the benchmark, that, on average, corresponds to 185 meters lower localisation error.

However, the current model is limited by the condition that the observed target is a stationary value which is typically the case for a location target. Furthermore, in the application of such a model to the OpenSignalMap cell tower dataset, we could only consider single-reporting by the users since we were not provided with the user identifier that could link between the user and its multiple reports. To remove these limitations, and so be able to apply our trust-based approach to a broader range of crowdsourcing problems, the next chapter will introduce a more general trust-based crowdsourcing model setting for a non-stationary function target, particularly considering crowdsourcing spatial function targets in a multi-reporting. Such a model will also be referred to the stationary target problem investigated in this chapter as a special case of a constant value spatial function. We will then apply this model to a more comprehensive real-world dataset provided with user identifiers which will allows us to consider multiple-reporting.

Tower ID [CID, LAC]	CU	CI	LOF	RM	**MaxTrust**
[1687, 608] *(50.908 N 1.358 W)*	1440m	957m	700m	582m	528m
[11259544, 109] *(50.907 N, 1.408 W)*	1461m	1061m	955m	1020m	924m
[209873204, 3202] *(50.923 N, 1.434 W)*	919m	487m	539m	420m	465m
[24155, 122] *(50.909 N, 1.408 W)*	1740m	1055m	1177m	959m	985m
[45995383, 217] *(50.911 N, 1.447 W)*	1309m	1042m	935m	914m	901m
[62172, 608] *(50.915 N, 1.459 W)*	1350m	1368m	301m	1390m	850m
[46005029, 217] *(50.917 N, 1.287 W)*	1929m	644m	768m	783m	744m
[4664508, 43582] *(50.904 N, 1.417 W)*	1246m	257m	424m	243m	192m
[46195850, 21] *(50.876 N, 1.265 W)*	2947m	2767m	3574m	295m	400m
[45995383, 217] *(50.911 N, 1.447 W)*	1309m	1042m	935m	914m	901m
[4684349, 43582] *(50.939 N, 1.350 W)*	495m	1208m	1071m	1131m	689m
[46195491, 21] *(50.887 N, 1.291 W)*	3125m	1593m	1638m	1074m	853m
[11694, 122] *(50.908 N, 1.400 W)*	1050m	1159m	938m	1040m	889m
[45988753, 217] *(50.900 N, 1.311 W)*	1332m	1468m	259m	812m	268m
[4671127, 43582] *(50.951 N, 1.382 W)*	1256m	368m	589m	493m	282m
Average	1673.60m	1243.70m	1253.90m	866.17m	684.43m

TABLE 3.2: Distance (in meters) from the predictive mean produced by the algorithms from the ground truth location (reported in brackets) for 15 cell towers.

Chapter 4

A Trust-Based Heteroscedastic Gaussian Process Model for Crowdsourcing Spatial Functions

In this chapter, we present our second model that addresses part of the shortcomings of MaxTrust and extends the generality of our trust approach. Specifically, as discussed in Section 3.4, MaxTrust suffers from the limitation of assuming that the target observed by the crowd is a stationary value and, as such, observations can be assumed to be all referring to one single correct value. In particular, this setting is suitable for applications of crowdsourcing location targets as in the example of crowdsourced cell tower localisation presented in Section 3.3.2. However, it is of practical interest for many other crowdsourcing applications to consider the more general setting of crowd reporting about a non-stationary quantity. In this context, the typical example is a crowd observing an environmental process, such as a temperature map, a weather map or a radiation map. In this case, the spatio-temporal correlation on the collected dataset makes inference more challenging especially when also the individual trustworthiness of the reports needs to be considered.

Therefore, we now address the requirement of making inference over a non-stationary quantity from reported observations (requirement 5) while dealing with the uncertainty of data trustworthiness which characterises crowdsourced information (requirement 1). Specifically, as a first step to address this problem, we focus on spatial inference with crowdsourced data where the objective is to learn an unknown spatial function from a set of observations reported by untrustworthy users. To tackle this problem, we propose a new trust-based heteroscedastic Gaussian process (HGP) model that combines the trust approach underpinning MaxTrust (see Section 3.1.1) in spatial regression. Specifically, the qualities of this model are to (i) make predictions of an underlying spatial process from a set of untrustworthy observations (requirement 2), (ii) to consider the reported

uncertainty on the single observation in the inference (requirement 4) and (iii) to provide estimates of the trustworthiness of the single user (requirement 5).

The remainder of this chapter is structured as follows. Section 4.1 formally describes our trust-based HGP model for untrustworthy location-based observations. Then, Section 4.2 describes the algorithm, called TrustHGP, to compute the predictive distribution of the observed spatial function under such a model, including user trustworthiness estimation. Section 4.3 empirically evaluates TrustHGP using both synthetic data and real-world data. In particular, the experiment with synthetic data aims to show the effectiveness of TrustHGP in various simulated settings of untrustworthy crowds. Then, the second experiment presents a real application of TrustHGP on the key disaster response application of crowdsourced radiation monitoring. In particular, this experiment will involve crowdsourced radiation data from the 2011 Japan's earthquake used to estimate the radiation map of Japan after the Fukushima nuclear disaster. Finally, Section 4.1.2 provides a discussion of the results and concludes.

4.1 Model Description

This section formally describes our trust-based HGP model for spatial regression with untrustworthy data. First, Section 4.1.1 introduces the trust model of a user in reporting spatial estimates. Then, 4.1.2 details the HGP model to incorporate such a trust model in spatial regression.

4.1.1 A Trust Model for Spatial Crowd Reporting

In this model, there is a crowd of n users $U = \{i, \ldots, n\}$ observing an environmental process represented by the function $f : \mathcal{R}^n \to \mathcal{R}$. Specifically, recalling the examples in disaster management given in Section 1.3, f may represent the spreading of a nuclear cloud generated by a nuclear accident, as it happened in the aftermath of the 2011 Japan earthquake, or the waterborne disease that spread across the Haiti population due to ground water contamination provoked by of the 2010 earthquake (Walsh, 2010). Generally speaking, f represents an output varying spatially across a set of input locations, where the domain of f is the continuous set of locations describing land area covered by the process and the codomain is the range of real values that such a process can assume. Therefore, we will conveniently describe the model for the case of $n = 2$ to conform to with the case of spatial functions.

Thus, in a multi-reporting setting, each user i reports a set of p_i observations, where each observation $\boldsymbol{r}_{i,j}$ provides (i) a location $\boldsymbol{x}_{i,j} \in \mathcal{R}^2$, namely the position of the observer, (ii) the output $y_{i,j} \in \mathcal{R}$, namely the value measured at $\boldsymbol{x}_{i,j}$ and (iii) the precision $\theta_{i,j} \in \mathcal{R}$ with $0 < LB < \theta_{i,j} < UB$, namely the precision of $y_{i,j}$. In particular, as also detailed in

Section 1.2, $\theta_{i,j}$ may be referring to the precision of a sensor, or user's confidence level, or the variance of some repeated measurements. Summing up, $R = \{\langle \boldsymbol{x}_{i,j}, y_{i,j}\rangle | j = 1 \dots p_i\}$ is the dataset of dimension $p = \sum_{i=1}^{n} p_i$; $\boldsymbol{x} = \{\boldsymbol{x}_{i,j} | i = 1, \dots n \quad j = 1, \dots p_i\}$ is the set of the reported locations and $\boldsymbol{y} = \{y_{i,j} | i = 1, \dots n \quad j = 1, \dots p_i\}$ is the vector of the observations.

Then, as in the MaxTrust model (Section 3.1.1), we assume that each user has an individual level of trustworthiness denoted by the parameter $t_i \in (0, 1]$ (1 for a fully trustworthy user and 0 for an untrustworthy use) and $\boldsymbol{t} = (t_1, \dots, t_n)$ is the set of such trustworthiness parameters. Specifically, we use the concept of *consistency* to characterise the trustworthiness of a user. In more detail, trustworthiness is shaped on the level of how the user reports observations that are representative samples of f. That is, a trustworthy user is expected to consistently report observations sampling from f with a random noise. In contrast, untrustworthy users typically reports observations uncorrelated with f. (see Section 3.1 for further details). Given this, we assume that reported observation are normally distributed with respect to the function value. Formally, let $\tilde{y}_{i,j}$ be actual value of f at $\boldsymbol{x}_{i,j}$, i.e. $\tilde{y}_{i,j} = f(\boldsymbol{x}_{i,j})$ then we consider that $y_{i,j}$ is a noisy measurement of $\tilde{y}_{i,j}$ with an additive zero-mean, Gaussian noise $\epsilon_{i,j}$ parametrised by t_i. That is:

$$p(\tilde{y}_{i,j} | y_{i,j}, t_i) = \sqrt{\frac{t_i \theta_{i,j}}{2\pi}} \exp\Big(- \frac{t_i \theta_{i,j} (\tilde{y}_{i,j} - y_{i,j})^2}{2} \Big) \tag{4.1}$$

or conventionally written as:

$$y_{i,j} = \tilde{y}_{i,j} + \epsilon_{i,j}, \qquad \tilde{y}_{i,j} = f(\boldsymbol{x}_{i,j}), \qquad \epsilon_{i,j} \sim \mathcal{N}(0, 1/(t_i \theta_{i,j})) \tag{4.2}$$

In such a way, we obtain the same noise scaling effect of an untrustworthy estimate described in MaxTrust (Section 3.1.2), where an untrustworthy estimate is downgraded by increasing its uncertainty proportionally to t_i

In more detail, Figure 4.1 shows an example of six users with different kinds of trustworthy behaviours in observing a one-dimensional function f represented by beta distribution with parameters $\alpha = 6, \beta = 18$ (blue-dotted line). Specifically, in this example, each user repors 5 estimates along x, and each estimate is plotted as its mean value $y_{i,j}$ (starred point) and the bars denote the 95% confidence interval given by $\pm 2/t_i \theta_{i,j}$. From this, we can see that user 1 and user 3 are highly trustworthy since all their estimates are consistent with the actual value $f(x_{i,j})$. Furthermore, user 2 has only one (the left-most) estimate that is inconsistent with $f(x_{i,j})$, so it's behaviour is mostly trustworthy. In contrast, users 4 and 6 are highly untrustworthy reporters since all of their estimate are significantly distant (more than 2 standard deviations) from $f(x_{i,j})$. Finally, the behaviour of user 6 is mostly untrustworthy since only one of its five estimates is consistent with $f(x_{i,j})$. Therefore, while in principle user trustworthiness is assumed to be binary, i.e. its strategy is either or not to submit trustworthy estimates, then we capture such

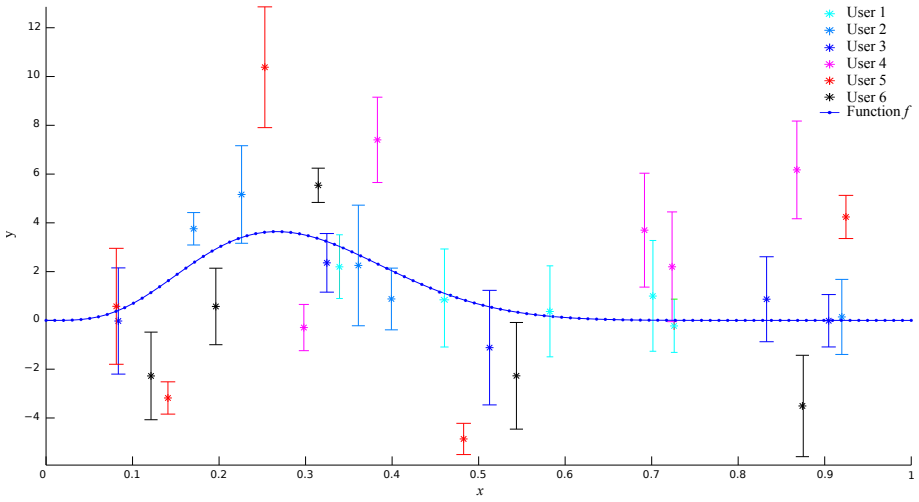


FIGURE 4.1: Example of trustworthy (user $1, 2, 3$) and untrustworthy (user $4, 5, 6$) reporting behaviour.

a behavioral feature as a (0,1] value, represented by the t_i parameter, based on the level of consistency of the user's estimates with the ground truth's function. However, the challenge here is how to find the values of t_i when f is not available, thus we detail how we address this problem through a Gaussian process regression approach in the following section.

4.1.2 Trust-Based Heteroscedastic Gaussian Process Regression

As discussed in Section 2.4, the Gaussian process (GP) is a machine learning tool widely used to solve non-linear regression problems in a principled Bayesian framework performing analytically tractable inference (Rasmussen, 2004). Specifically, we use the GP to infer f from the dataset described above taking into account the trustworthiness parameters of each users. In particular, given the noise model stated in Equation 4.2, where the noise is individually set on the observations as a function of the reported precision and the trustworthiness parameter, then we look at the family of heteroscedastic Gaussian process models (HGP) that deals with data with individual noise terms. In particular, we use the HGP model studied by Goldberg et al. (1997) discussed in Section 2.4 that assumes that the noise terms are jointly independent. In our setting, this corresponds to the assumption that (i) a user reports independent precisions and (ii) a user is individually trustworthy. These, as discussed in Section 2.5, are reasonable assumptions in spatial crowd reporting settings.

Specifically, we define a new trust-based HGP model obtained from combining the Goldberg et al.'s HGP to the model of untrustworthy reports defined by Equation 4.1. Formally, a GP prior is placed over f with a mean function $m(x)$, (hereafter we assume to

be zero for simplicity), and a covariance function or kernel $K(x, x')$, that is:

$$f(x) \sim \mathcal{GP}(0, K(x, x')) \tag{4.3}$$

From this, in analogy with Equations 4.1 and 4.2, we assume that the likelihood of a vector of observations $\boldsymbol{y}$ given f is a normal distribution expressed as follows:

$$p(\boldsymbol{y}|f) = \mathcal{N}(\boldsymbol{y}|f, \epsilon_{i,j}) \qquad \epsilon_{i,j} \sim \mathcal{N}(0, 1/(t_i\theta_{i,j}) \tag{4.4}$$

Now, let $\boldsymbol{x}_*$ be a new test location in the domain of f, and y_* be the corresponding unobserved output, then the joint distribution of y_* and $\boldsymbol{y}$ under the current model is a Gaussian PDF written as follows:

$$\begin{bmatrix} \boldsymbol{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x & K(\boldsymbol{x}, \boldsymbol{x}_*) \\ K(\boldsymbol{x}_*, \boldsymbol{x}) & K(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{bmatrix}\right) \tag{4.5}$$

where

$$\Sigma_x = diag((t_i\theta_{i,j})^{-1}) \tag{4.6}$$

Specifically, Σ_x is the diagonal matrix of the noise terms $\epsilon_{i,j}$ denoting that the noise of each input e covariance matrix is jointly regulated by $\theta_{i,j}$ and t_i. Notice that, if such a noise is constantly set to σ_n, then Equation 4.5 is the same equation of the standard GP with $\Sigma_x = \sigma_n \mathrm{I}$.

Under such a model, predictions can be made by conditioning $\boldsymbol{x}_*$ to the set of reports collected from the crowd R, given the trustworthiness of individual crowd members be defined by $\boldsymbol{t}$. Then, using the marginalisation properties of the Gaussian distributions, the predictive distribution of $f(\boldsymbol{x}_*)$, i.e. the density over the function at the test location, is derived as follows.

$$p(\boldsymbol{y_*}|\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}_*, \boldsymbol{t}) = \mathcal{N}(E[\boldsymbol{y}_*], \sigma^2(\boldsymbol{y}_*)) \tag{4.7}$$

where

$$E[\boldsymbol{y}_*] = K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x]^{-1}\boldsymbol{y} \tag{4.8}$$

$$\sigma^2(\boldsymbol{y}_*) = K(\boldsymbol{x}_*, \boldsymbol{x}_*) - K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x]^{-1}K(\boldsymbol{x}, \boldsymbol{x}_*) \tag{4.9}$$

Furthermore, integrating out f from Equation 4.4 using the GP prior of Equation 4.3, then we derive the marginal log-likelihood as follows:

$$\begin{aligned} \log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{\theta}) = &-\frac{1}{2}\boldsymbol{y}^T[K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x]^{-1}\boldsymbol{y} \\ &-\frac{1}{2}\log|K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x| - \frac{n}{2}\log(2\pi) \end{aligned} \tag{4.10}$$

As we notice, the model depends on a set of hyperparameters $\boldsymbol{\Theta}$ that includes the k hyperparameters $\boldsymbol{\theta}$ of the covariance function K, controlling its smoothness properties, and the n trustworthiness parameters. Such hyperparameters define the properties of the model for a particular set of input observations.

In summary, we derived the key equations of the TrustHGP to predict the mean (Equations 4.8) and the variance (Equations 4.9) of f from the data at any input location. In particular, the marginal likelihood is useful to train the GP for model selection. Specifically, this expression allows us to derive the maximum likelihood estimates of the hyperparameters through maximising the function as follows:

$$\boldsymbol{\Theta}_{\mathrm{ML}} = \{\boldsymbol{\theta}_{\mathrm{ML}}, \boldsymbol{t}_{\mathrm{ML}}\} = \arg\max_{\boldsymbol{\theta}', \boldsymbol{t}} (\log\, p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{\theta}) \tag{4.11}$$

Therefore, in a similar vein to Kersting et al. (2007), we adopt the maximum likelihood estimation approach to learn the hyperparameters and, as the next step, we provide an algorithm to perform such a likelihood maximisation computationally.

Before this, as introduced in Section 2.5, we observe that there is another model presented by Groot et al. (2011) which deals with regression with multiple reports using the standard GP with a rational quadratic covariance function, that is

$$K(\boldsymbol{x}_*, \boldsymbol{x}) = \sigma_f \Big(-\frac{1}{2}(x - x')^T A^{-1}(x' - x) \Big) \tag{4.12}$$

This is a kernel that has an individual length scales l_i for each input dimension denoted as $A = diag(l_1^2, \ldots, l_p^2)$, and σ_f is a signal noise parameter. Then, likewise our approach, they use the maximum marginal likelihood estimator to learn the hyperparameters referring to each length scale l_i as the accuracy (trustworthiness) of the input i. However, given that their model is not designed to take the precisions $\theta_{i,j}$ into account, it also suffers from scalability issues when the set of p hyperparameters (one for each report) is large. In contrast, we only use the n trustworthiness hyperparameters (one for each user) that is more feasible computationally. Given this, the algorithm to learn the hyperparameters and ultimately predict the function and the trustworthiness levels is described in the next section.

4.2 The TrustHGP Algorithm

In this section, we describe the algorithm, called TrustHGP, for computing the trustworthiness parameters and the predictive distribution of f maximising the marginal log-likelihood of our trust-based HGP (see Algorithm 4). However, since such a likelihood is non-linear function, then its maximisation is not tractable analytically and it must be carried out numerically. Specifically, we use the non-linear conjugate gradient method (Saad and Saad, 1996). This is a iterative method for minimising quadratic

Algorithm 4 TrustHGP (Non-linear conjugate gradient)

Variables :

R : Report set.
$\boldsymbol{\Theta}^{(h)}$: Hyperparameters at the h-th iteration.
$\boldsymbol{\theta}^{(0)}$: Initial guess of the covariance function's hyperparameters.
$\Delta\boldsymbol{\Theta}^{(h)}$: Negative derivatives of the marginal log-likelihood with respect to the hyperparameters of the h-th iteration.
$\boldsymbol{x}_*$: Test inputs.
error : Estimation error bound.
$h^{\max}$: Maximum number of iterations.

Algorithm *TrustGP*$(R, \boldsymbol{x}_*)$

1: $\boldsymbol{t}^{(0)} := \langle 1, \ldots, 1 \rangle$
2: $\boldsymbol{\Theta}^{(0)} := \langle \boldsymbol{\theta}^{(0)}, \boldsymbol{t}^{(0)} \rangle$
3: $s^{(0)} = -\frac{\partial}{\partial \Theta}\big(\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Theta}^{(0)})\big)$ (Equation 4.10)
4: $h := 0$
5: **while** ($|\boldsymbol{\Theta}^{(h-1)} - \boldsymbol{\Theta}^{(h)}| < err$ **and** $h < h^{\max}$) **do**
6: $\quad h := h + 1$
7: $\quad \Delta\boldsymbol{\Theta}^{(h)} := -\frac{\partial}{\partial \Theta}\big(\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Theta}^{(h-1)}\big)$
8: $\quad \beta^{(h)} := \frac{(\Delta\boldsymbol{\Theta}^{(h)})^T(\Delta\boldsymbol{\Theta}^{(h)} - \Delta\boldsymbol{\Theta}^{(h-1)})}{(\Delta\boldsymbol{\Theta}^{(h-1)})^T\Delta\boldsymbol{\Theta}^{(h-1)}}$ (Polak-Ribière method)
9: $\quad s^{(h)} := \Delta\boldsymbol{\Theta}^{(h-1)} + \beta^{(h)} s^{(h-1)}$ (Wolfe line search)
10: $\quad \alpha^{(h)} := \arg\max_\alpha p(\boldsymbol{y}|\boldsymbol{x}, (\boldsymbol{\Theta}^{(h-1)} + \alpha s^{(h-1)}))$
11: $\quad \boldsymbol{\Theta}^{(h)} := \boldsymbol{\Theta}^{(h-1)} + \alpha^{(h)} s^{(h)}$
12: **end while**
13: $\boldsymbol{\Theta}^{(h)} := \langle \boldsymbol{\theta'}^{(h)}, \boldsymbol{t}^{(h)} \rangle$
14: Compute $E[\boldsymbol{y}_*|\boldsymbol{x}_*]$ as by Equation 4.8.
15: Compute $\sigma^2(\boldsymbol{y}_*|\boldsymbol{x}_*)$ as by Equation 4.9.
16: **return** $(\boldsymbol{t}^{(h)}, E[\boldsymbol{y}_*], \sigma^2(\boldsymbol{y}_*))$

functions (that in our case is equal to minimise the negative log-likelihood) following the steepest conjugate gradient direction given by the analytical gradient of the function. In particular, as is apparent from the illustration of Figure 4.2, such a method typically converges to a (local) minimum faster than gradient descent that follows perpendicular (zig-zag) directions.

In more detail, the algorithm is described as follows. Step 1 and 2 initialises the hyperparameters with $\boldsymbol{t}$ uniformly set to 1, that corresponds to start by assuming that all the users as trustworthy, and making a random guess of $\boldsymbol{\theta}$. Then, the conjugate gradient loop (step 5-12) computes the gradient with respect to the hyperparameters of the previous iteration $\boldsymbol{\theta}^{(h-1)}$ and the search directions given by the β and α parameters. In particular, there are a number of methods for computing β based on different versions of the conjugate gradient algorithm (Saad and Saad, 1996). Among these, we use Polak-Ribiere method (step 8) that was found to be computationally more stable in our setting. Then, step 10 computes the search directions s and the step length along each directions α through Wolfe line search condition. In particular, by using such a condition

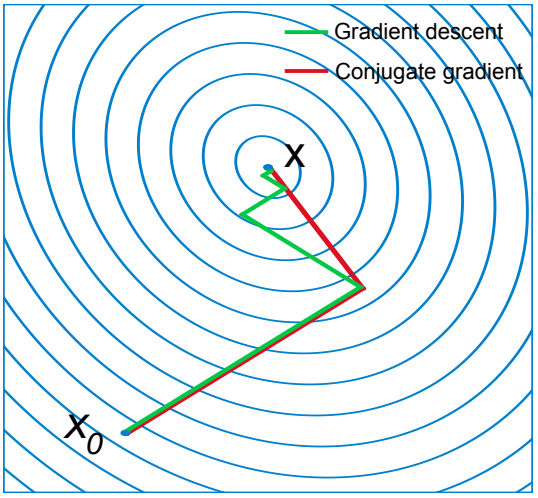


FIGURE 4.2: A comparison of the convergence of the gradient descent and the conjugate gradient methods in minimising a quadratic function.

to update of the step length, the method is proved to ensure stability and convergence (Wolfe, 1969). Finally, the hyperparameters are updated according to the new α and s in step 11. After convergence is achieved, and such a convergence is empirically reached in 20-100 iterations, then the algorithm returns the trustworthiness values from the last iteration and the mean and variance predictions of the function at the test inputs $\boldsymbol{x}_*$. Analysing its complexity, the algorithm requires $O(p^3)$ time to compute the output due to the inversion of the covariance matrix that is a lower-bound complexity for any GP. However, after the inversion of the the covariance matrix, then prediction only takes $O(p)$ time for the predictive mean and $O(p^2)$ for the predictive variance.

Having now described our TrustHGP algorithm, the following section provides its empirical evaluation against other non-trust GP regression approaches.

4.3 Experimental Evaluation

In order to empirically evaluate our TrustHGP, we conduct experiments on both synthetic and real-world data. In the first experiment, we test the algorithm on synthetic data simulating spatial crowd reporting with different levels of untrustworthy users within the crowd (Section 4.3.1). Then, the second experiments, we look at the key disaster response application of crowdsourced radiation monitoring evaluating TrustHGP in making spatial predictions on a dataset of crowdsourced radiation data from the 2011 Japan's earthquake (Section 4.3.2).

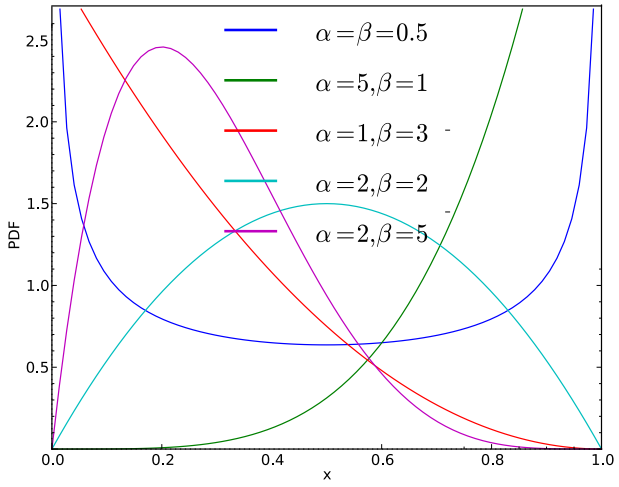


FIGURE 4.3: Beta probability distribution for different values of shape parameters.

4.3.1 Experiment on Synthetic Data

In this experiment, we evaluate TrustHGP in estimating a one-dimensional function from the reports provided by a crowd of untrustworthy observers. Specifically, the experiment is set up as follows. The spatial function f is a beta distribution, $Beta(\alpha, \beta)$ with the two positive shape parameters α and β randomly sampled, i.e. $\{\alpha, \beta\} \sim U[1, 20]$. In particular, such a distribution can reproduce various shapes of a continuous function by tuning the two positive parameters α and β as it is showed by Figure 4.3. Then, a number of observations are provided by a crowd of 20 users. Specifically, each user i reports p_i estimates $\langle x, y, \theta_i \rangle$, with $p_i \sim [3, 20]$, where x is a point randomly selected in [0,1] (i.e. the domain of f), y is the observed output and θ_i is the reported precision, respectively. Specifically, each report contains noisy observations of f generated as follows:

$$\theta \sim U[0.5, 20] \qquad x \sim U[0, 1] \tag{4.13}$$

$$y = f(x) + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \theta^{-1}) \tag{4.14}$$

Then, to simulate a percentage ρ of untrustworthy users within the crowd, the estimates of untrustworthy users are added with an extra noise w drawn from $\pm[1, 5]$, formally:

$$y = f(x) + \epsilon + w \qquad \epsilon \sim \mathcal{N}(0, \theta^{-1}) \qquad w \sim \pm U[1, 5] \tag{4.15}$$

In particular, we adopt such setting for generating noise in order to avoid to introduce biases in our results as detailed in Section 3.3.1. Finally, to compare our TrustHGP against other non-trust GPs, we consider the following benchmark methods:

- **Standard GP**: This algorithm refers to the standard homoscedastic GP discussed in Section 2.5.

- **HGP**: This algorithm is the Goldberg et al.'s HGP (Section) 2.5.1 that uses θ_i as input noise terms and without including the trustworthiness parameters.
- **OptimalHGP**: This is the *hypothetical* optimal HGP provided with the knowledge of the correct values of trustworthiness for each user. That is, trustworthy users are set with $t_i = 1$ and untrustworthy users are set with $t_i = 0$.

Thus, the four algorithms: GP, HGP, TrustHGP, OptimalHGP were tested in N repeated trials. For the core of the GPs, we used the squared-exponential covariance function described as follows:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f \exp\Big(-\frac{1}{2l^2}(\boldsymbol{x} - \boldsymbol{x}')^2 \Big) \tag{4.16}$$

where the two hyperparameters are the signal variance σ_f and the length scale l respectively. Then, to measure the quality of the distribution predicted by each method, we use the accuracy metrics used described below.

Accuracy Metrics

We want to measure the accuracy of the algorithm in terms of the accuracy in predicting f as well as the level of uncertainty in the predictive distribution. In particular, we seek accuracy metrics that consider jointly the predictive mean and the predictive uncertainty in scoring predictors. To this end, we draw from the lessons learned by the "evaluating predictive uncertainty challenge" (EPUC) that was organised within the machine learning community in December 2004, with the aim of evaluating a number of submitted prediction methods competing in various classification and regression tasks (Quinonero-Candela et al., 2006). Such a challenged revealed that a good scoring method requires the property of *properness* in the sense that the true generative distribution must have the best expected score. However, Kohonen and Suomela (2006), a team that participated to the challenge, discuss that in addition to properness another requirement for such a scoring rule is the *non-locality* property. i.e the score must also be *distance sentitive* and dependent on how much predictive probability mass is placed near the true target. In contrast, a *local* score depends only on the predictive density exactly at the true target values. Therefore, following the suggestions of the authors, we use the non-local scoring rule of the*continuous ranked probability score* (CRPS) (Gneiting and Raftery, 2007). In particular, for scoring Gaussian predictive distributions, the CRPS averaged over K point predictions and N simulations is given by:

$$\begin{aligned} CRPS(\mathcal{N}(\boldsymbol{y}, \boldsymbol{\sigma}^2), \boldsymbol{y}^*) = & \frac{1}{NK} \sum_{i=1}^{N} \sum_{j=1}^{K} \sigma_{i,j} \left(\frac{1}{\sqrt{\pi}} - 2\varphi\left(\frac{y^*_{i,j} - y_{i,j}}{\sigma_{i,j}} \right) - \right. \\ & \left. \frac{y^*_{i,j} - y_{i,j}}{\sigma_{i,j}} \left(2\phi\left(\frac{y^*_{i,j} - y_{i.j}}{\sigma_i} \right) - 1 \right) \right) \end{aligned} \tag{4.17}$$

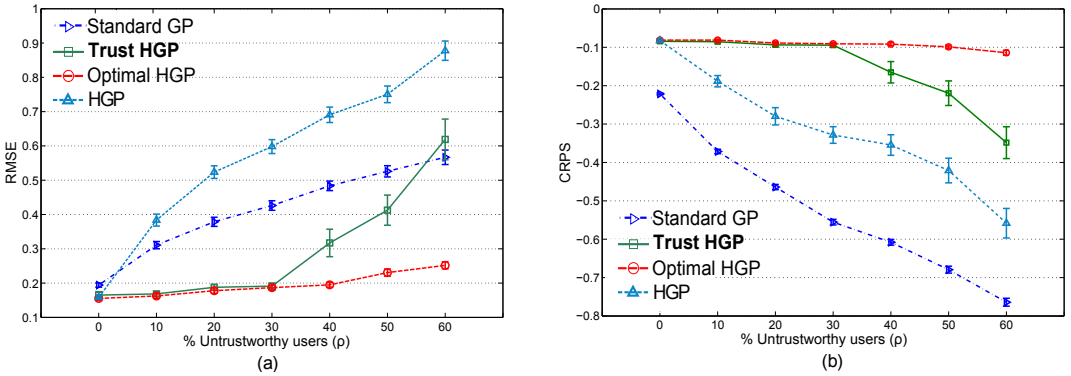


FIGURE 4.4: Performance of the four methods measured by the root mean square error error (a) and the continuous ranked probability score (b).

where φ and ϕ denote the probability density function and the cumulative distribution of a standard normal random variable, respectively. Specifically, $\langle y_{i,j}, \sigma_{i,j} \rangle$ is the predicted Gaussian estimate of the actual point $y^*_{i,j}$ at the i-th simulation.

Furthermore, we will also use the root mean square error (RMSE) averaged over K predictions and N simulations to measure of accuracy of the prediction only based on the predictive mean. That is:

$$RMSE(\boldsymbol{y}, \boldsymbol{y}^*) = \sqrt{\frac{1}{NK} \sum_{i=1}^{N} \sum_{j=1}^{K} (y^*_{i,j} - y_{i,j})^2} \tag{4.18}$$

Given these accuracy metrics, the results of this experiment are discussed in the following section.

Results

The results of run $N = 200$ simulation varying the vaue of ρ as follows: $\rho = \{$ 0, 10, 20, 30, 40, 50, 60 $\}$. are showed in Figure 4.4. From this, we can see that, as expected, the global RMSE (Figure 4.4 (a)) (the lower the better) of the algorithms grows progressively with ρ, meaning that a large number of untrustworthy users penalises the accuracy of the predictions. Notwithstanding, the TrustHGP outperforms the other methods by up to 34% when $\rho = 30$ (the statistical significance of this result was tested by a paired t-test at the 0.01 level: $p = 3.4 \cdot 10^{-33}, t(14.48) = 0.11$). In particular, its error is very close to the optimum until $\rho = 30$ and is generally the lowest for $\rho < 50$.

Another interesting result is the CRPSs of the four methods showed in Figure 4.4 (b) (the higher, the better). From this, we can see that, the scores of the TrustGP are significantly higher than the other methods (excluding the OptimalHGP) for any ρ value. In particular, it outperforms the standard GP by 80% when $\rho = 30$ (statistical significance tested by a paired t-test at the 0.01 level: $p = 3.38 \cdot 10^{-124}$, $t(56.37) = 0.06$). Thus, this shows that our algorithm is not only the most accurate among the tested

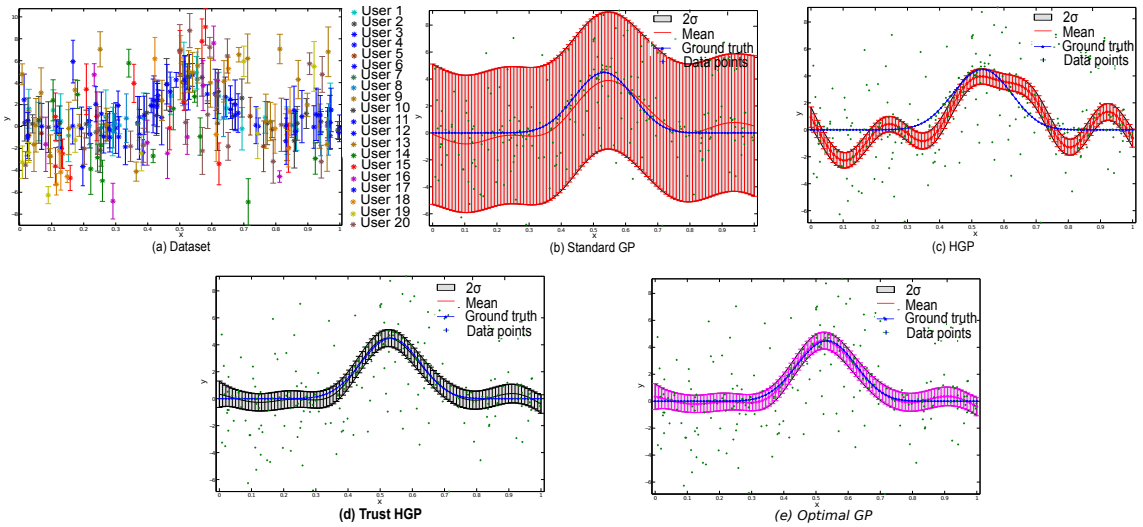


FIGURE 4.5: Example of regression of the four methods on a sample dataset of 20 users, 241 data points and 30% untrustworthy users.

methods but it is also very informative due to a low predictive uncertainty. Interestingly, the CRPS also reveals that the standard GP ranks below the the HGP in terms of predictive uncertainty, even though the RMSE of the former is typically lower.

In more detail, Figure 4.5 shows the typical prediction results produced by the four methods. Given the dataset illustrated in Figure 4.5 (a), with 20 users, with totally 241 reports, and $\rho = 30$, the Standard GP prediction is showed in Figure 4.5 (b). In particular, such a prediction is very noisy due to the training that increases the signal noise σ_n to include all the points, thus exhibiting a behaviour similar to the CU method (Section 3.3.1). Furthermore, the HGP prediction showed in Figure 4.5 (c)) has typically lower uncertainty but is inaccurate due to the effect of chasing every points considered by the process to a trustworthy report. In contrast, the TrustHGP prediction showed in Figure 4.5 (d) is the best trade-off between accuracy and low uncertainty due to a correct estimation of trustworthiness parameters that allows the process to exclude most of the untrustworthy points. In particular, consistently with the scores, its prediction is very close (almost identical) to the one of the OptimalGP showed Figure 4.5 (e), that makes our results more valuable given that the latter has the advantage of knowing the trustworthiness values in advance.

Thus, we showed that our method outperforms the benchmarks in both accuracy and informativeness in predicting the function from synthetic data. Now, to reinforce this empirical result, we provide an evaluation of the TrustHGP also on real-world data.

4.3.2 Experiment on Real-World Data

In this experiment, we consider the real-world application of crowdsourced radiation monitoring in disaster response introduced in Section 1.1. In particular, we refer to

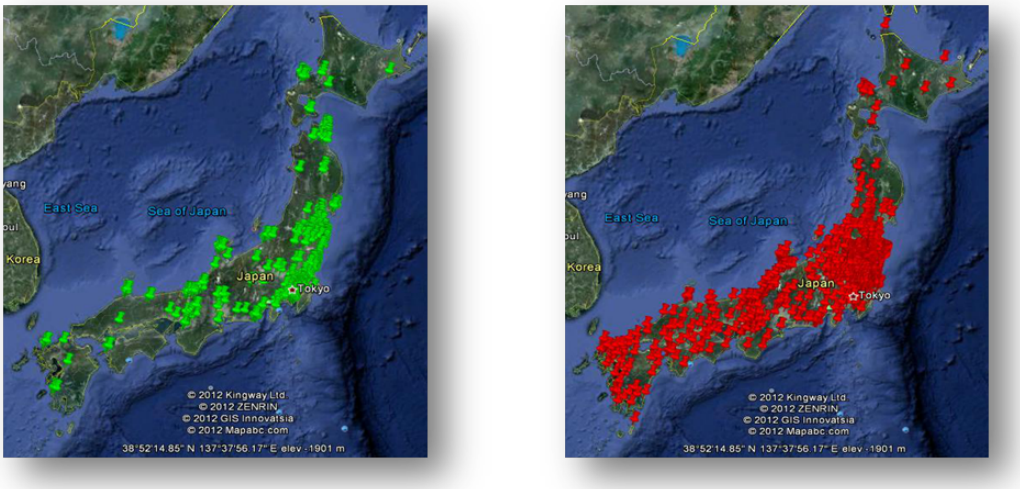
the scenario of the aftermath of the 2011 Japan earthquake where the effort of local communities contributed to install more than 500 radiation sensors across the country in less than two weeks following the disaster. Such a crowdsourced sensor network, created through the COSM platform (`cosm.com`), that we will refer to as the COSM network, provided real-time radioactivity measurements to first responders and was acknowledged to be of great help for monitoring the spreading of the nuclear cloud that was generated by a nuclear power plants damaged by the earthquake. However, a key challenge for the first responders was to manage such large amount of sensor data into a global spatial prediction of radiations, whilst taking into account the fact that some sensors were verifiably unreliable (Borden, 2011). Against this background, we now detail how to apply the TrustHGP in such a radiation monitoring application to help address this challenge.

Dataset

The COSM network consists of 557 sensors placed at known locations and each sensor provides readings at the average frequency is of 2 readings per hour. Specifically, the sensor periodically reports (i) the measured radiation value in the standard unit of microsieverts per hour ($\mu Sv/h$) and (ii) the timestamp of such a measurement. The complete description of the COSM dataset in provided in the Appendix B. Furthermore, the ministry of education, culture, sport, science and technology of Japan (MEXT) maintains a national radiation sensor network named SPEEDI: system for prediction of environmental emergencies and dose information (`www.bousai.ne.jp`). SPEEDI includes 2122 sensors that provide readings in the unit of $\mu Sv/h$ at the frequency of 6 measurement per hour.. Specifically, Figure 4.6(b) shows the map of the SPEEDI network. From these two networks, we downloaded the data for each sensor over one day, 1 April 2012, i.e. 13 months after the earthquake.[1] For each sensor, we compute the mean and the variance of the readings that we refer to as the expected measurement x_i and the variance θ_i^{-1} of the sensor i, respectively. In more detail, Figure 4.6 shows the map of (a) the COSM network and (b) the SPEEDI network.

In radiation monitoring, the SPEEDI network is typically considered more reliable than the COSM network, therefore we use the SPEEDI data to build a ground truth for our experiment as follows. We use the standard GP to predict the spatial field of radiations from the SPEEDI dataset and such a prediction is shown in Figure 4.7 (a) as a radiation heat map with a colormap in the scale of 0 - 400 $\mu Sv/h$. Then, we use the mean value of such prediction as a comparative ground truth to evaluate the predictions made on the COSM data.

[1]The SPEEDI network offers digitalised data only starting from April 2012 so this date was selected such that data were available by both the two networks.



(a) (b)

FIGURE 4.6: Picture of the 557 radiation sensors of the COSM network (a) and the 2122 radiation sensors of the SPEEDI sensors (b) located in Japan.

As in the previous experiment, we use the squared-exponential function as the kernel of our GPs. However, given that the squared-exponential function is a stationary function, i.e. its value depends only on the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$, then such a function can be applied to location data and compute the covariance matrix directly using the geographical line distance between the two location points as follows:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f \exp\Big(- \frac{d(\boldsymbol{x}, \boldsymbol{x}')^2}{2l^2} \Big) \tag{4.19}$$

where

$$d(\boldsymbol{x}, \boldsymbol{x}') = R_0 \sqrt{x^2 + y^2} \tag{4.20}$$

$$x = (lon - lon_0) cos(lat) \tag{4.21}$$

$$y = (lat - lat_0) \tag{4.22}$$

Specifically, R_0 is the mean radius of the Earth equal to 6,371 km, and $\langle lat_0 = 24°, lon_0 = 124° \rangle$ is the origin point for the projection. In particular, the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$, i.e. $d(\boldsymbol{x}, \boldsymbol{x}')$, is computed through the equilateral projection described above, that is computationally more efficient than computing the grand-circle, Haversine distance. Given this setting, we discuss the results of the experiment in the next section.

Results

To analyse the behaviour of the TrustHGP with different tests on the COSM dataset, we sample 90% of the COSM sensors in $N = 100$ rounds. In each rounds, we run the Standard GP and the TrustHGP and their predictions are compared against the

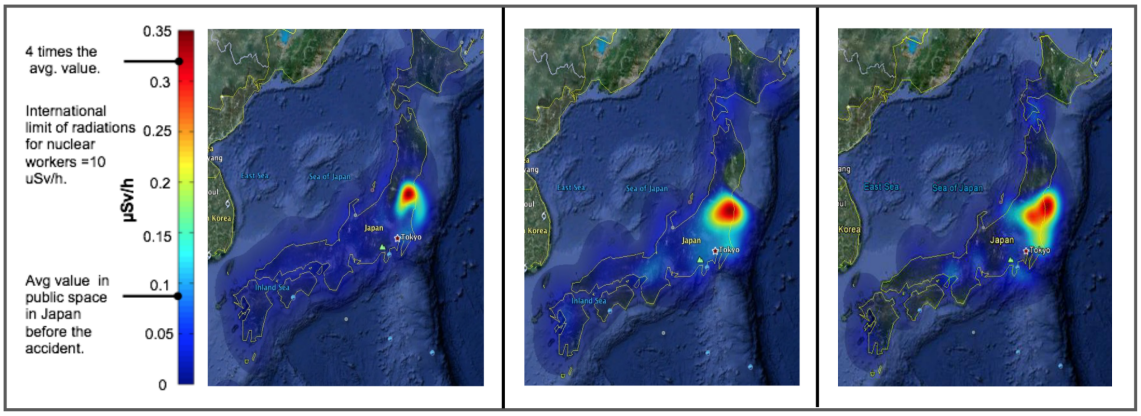


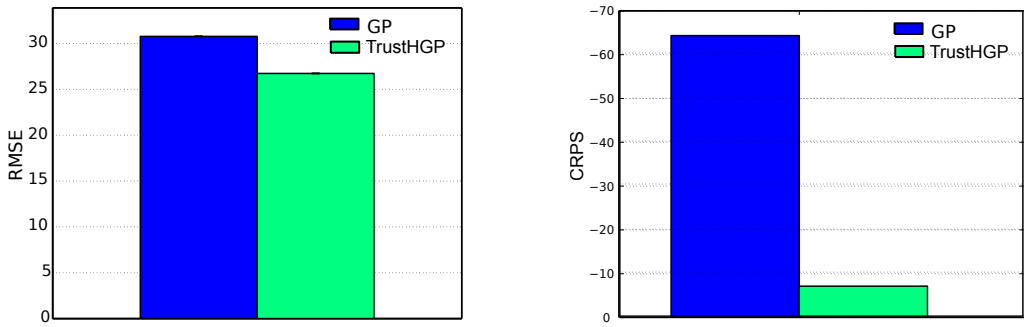
FIGURE 4.7: Radiation heat maps showing the following predictions: the standard GP on the SPEEDI dataset (a), the standard GP on the COSM dataset (b) and the TrustHGP on the COSM dataset (c).

GP-SPEEDI ground truth using the metrics described in Section 4.3.1. Figure 4.7 (b) and Figure 4.7(c) shows the predictive mean of the standard GP and the TrustHGP on the COSM data, respectively. In particular, we can see that all the methods are similar in predicting a field with a high peak of radioactivity near the area Fukushima that is indeed the area that was most significantly affected by the nuclear accident. In particular, the estimated radiations in such an area is around 0.35 $\mu Sv/h$, i.e. four times the average radiation level of 0.09 μSv measured in Japan before the earthquake.[2]

In addition, Figure 4.9 shows the global scores for the two predictors in terms of (a) RMSE and (b) CRPS. From this, we can see that our algorithm outperforms the standard GP by 13% with respect to the absolute error (RMSE) (statistical significance tested by a paired t-test at the 0.01 level: $p = 2.31 \cdot 10^{-167}$, $t(197.99) = 95.27$), and by 89% with respect to the predictive uncertainty (statistical significance tested by a paired t-test at the 0.01 level: $p = 0$, $t(141.07) = -5.86 \cdot 10^{-3}$). In particular, the result of the CRPS is significant as it shows that our prediction is considerably more informative than a normal GP prediction. This is even more evident by the 3D visualisation of the two predictions showed in Figure 4.8 where the red bars show the 2σ predictive standard deviation at each location. From this, we can see that the TrustHGP has very narrow bars compared to the high bars of the standard GP. Finally, our method estimated that 17% of the COSM sensors and only 1% of the SPEEDI sensors as untrustworthy which seems to be realistic given the nature of the crowdsourced COSM network opposed to the national SPEEDI network.

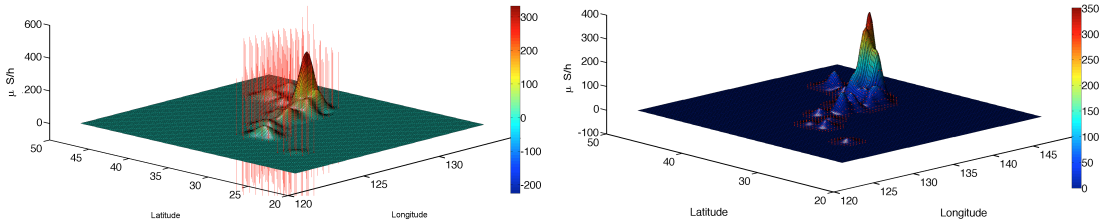
In summary, we find that the TrustHGP improves the quality of spatial prediction of nuclear radiations in this real-world application of crowdsourced radiation monitoring. In particular, it improves significantly in terms of lower predictive uncertainty which is

[2]Data source: Japan Radiation Open Data `sendung.de/japan-radiation-open-data`



(a) (b)

FIGURE 4.8: Bar plots of the RMSE (a) and the CRPS (b) of the two GPs.



(a) (b)

FIGURE 4.9: 3D visualisation of the GP prediction (a) and the TrustHGP prediction (b) on the COSM data.

a valuable property as our method generally provides highly informative estimates of the radiation levels at specific locations.

4.4 Summary

In this chapter, we showed that our trust-based approach effectively addresses the problem of estimating non-stationary quantities from untrustworthy data that complements the previous results of MaxTrust about stationary values in the same setting. In particular, we presented a new trust-based heteroscedastic Gaussian process for spatial regression to model a dataset of crowdsourced data reported by untrustworthy users. The salient feature of such a model is to integrate user trust learning in spatial regression through the principled heteroscedastic Gaussian process framework using a set of user trustworthiness parameters to scale the noise of the user's reported estimates.

Then, we presented the TrustHGP algorithm to estimate the hyperparameters under our model, including the learning of the trustworthiness of each user, using the maximum marginal likelihood estimator. Evaluating our method on synthetic data, we show that it outperform the standard, non-trust GPs being 34% more accurate and 80% more informative. Furthermore, a real application to the problem of crowdsourced radiation monitoring in Japan showed that our method estimates the radiation field with 13% lower error and 89% lower predictive uncertainty compared to the standard GP. However, we

envisage that the qualities of the TrustHGP could be advanced by considering also the temporal dimension into the spatial evaluation of trustworthiness and inference could be extended to a broader spectrum of information sources such as mobile sensors and single observations. Therefore, we outline the plan for future work in the next chapter.

Chapter 5

Conclusions

In this report, we developed a novel trust-based approach to make reliable inference on untrustworthy information in crowdsourcing applications. In particular, we focused on the inference tasks of information fusion and spatial regression with untrustworthy crowdsourced data. The motivation behind this research is that the data collected through the crowd is typically untrustworthy because it is provided by unreliable sources who may be inaccurate, uncommitted to the task and/or can also strategise which information to report. The key requirement in these applications is to combine the multiple reports gathered from the crowd into a predictive output of the underlying problem. In this context, the major challenge has been outlined in producing good quality aggregated estimates in the presence of untrustworthy reports. In more detail, this problem relates to the requirement of identifying trustworthy users within a crowd and evaluating a user 's trustworthiness according to the quality of its reports. Given this, we considered the approach of modelling user trustworthiness as a key informative concept for making reliable inference on crowdsourced data.

In more detail, in Chapter 2, the relevant literature in the field of machine learning for reliable crowdsourcing, including information fusion and spatial regression models, was reviewed. Emphasis was given to the class of graphical models that focus on learning the accuracy of the single user from crowdsourced datasets. However, the main limitation of such models lies in the fact that they do not consider users' reported uncertainties as part of the input data. This is a problem because crowd reported observations, generated through smartphones, do not only include point-based observations but also reported uncertainty values that numerically quantify the precision of the user with regard to its observation. As such, crowdsourcing must permit solutions in which inference takes into account such reported uncertainties in a crowdsourced dataset. In addition, in the field of information fusion, we discussed a number of methods that deal with the fusion of probabilistic estimates in the single-hypothesis and the multiple-hypothesis setting. These are the CI method and the CU method respectively. Whilst the latter was identified as the conservative fusion benchmark because of its property of unifying the estimates

under the most general output, the former was considered as a more valid basis for the design of our trust-based fusion method due to its property of reducing noise in fused estimates. Moreover, to address the requirement of identifying untrustworthy reports within a crowdsourced dataset, we discussed density-based outlier detection methods and sensor fusion algorithms for untrustworthy sensors. In particular, the former and the Reece et al.'s algorithm were considered as suitable representatives of these two classes respectively, and were included as benchmarks to evaluate our approach. Furthermore, we provided the key background of spatial inference with crowdsourced data when presenting the Gaussian process (GP) regression framework. In particular, we considered the class of heteroscedastic Gaussian processes (HGP) to model data with input-dependent noise that are suitable for dealing with reports with individual uncertainties as our problem requires. However, the standard HGP regression model does not provide any support against untrustworthy input data. To address this shortcoming, we developed a new trust-based HGP model for spatial regression that considers different levels of trustworthiness on each input.

The contribution of this work was detailed in the Chapters 3 and 4. Specifically, Chapter 3 presented our trust-based fusion model for crowdsourcing stationary quantities, taking as an application example a crowdsourced target localisation problem. The key feature of such a model is the introduction of a parametric representation of the uncertainty of a reported observation based the level of trustworthiness of the user. In so doing, the model can effectively reduce the noise of untrustworthy estimates in the aggregated output. In addition, we developed the MaxTrust algorithm to perform approximated maximum likelihood inference of the trustworthiness parameter and the fused estimate under such a model. The key qualities of MaxTrust are as follows: (i) it estimates user trustworthiness and the fused output without prior knowledge of the ground truth, (ii) it is free from any control parameter and (iii) its time complexity is polynomial in the size of the report set. From our experimental evaluation, empirical results show that MaxTrust outperforms the benchmarks on both synthetic and real-world data. Specifically, our algorithm is 52% more accurate and 80% more consistent (low predictive uncertainty) evaluated on target localisation tasks with simulated data. Furthermore, when employed in an application of crowdsourced cell tower localisation, MaxTrust improves the localisation accuracy by 21% compared to the other methods. In practice, corresponds to an average lowering of the error by 185 meters.

Furthermore, Chapter 4 detailed our model of trust-based heteroscedastic Gaussian process (HGP) regression for crowdsourcing spatial functions. This model integrates the trust approach underpinning MaxTrust within the principled Bayesian inference framework of heteroscedastic Gaussian process model. Then, the training of the hyperparameter through the maximum marginal likelihood estimator reveals the level of trustworthiness of the single user. Experiments on synthetic and real-world data show the

efficacy of our method. Specifically, the TrustHGP empirically outperforms the non-trust GP benchmarks by 34% in terms of accurate and by 80% in terms of predictive uncertainty in simulated experiments. In addition, the TrustHGP showed a significant impact when applied to the disaster response application of crowdsourced radiation monitoring. In particular, using radiation data from the 2011 Fukushima nuclear disaster provided by crowdsourced sensors, our method outperforms the benchmarks in making more accurate, by 13%, and with significantly lower uncertainty predictions, by 89%, of the radiation spatial field. Given these results, future extensions of the current model will allow the TrustHGP to integrate jointly the temporal and the spatial dimension in the evaluation of trustworthiness.

5.1 Future Work

The work presented in this report is an important step towards achieving the goal of delivering reliable inference models for crowdsourced information. The limitations of the models presented in this report and the requirements that were not achieved so far in this work pave the way for our future research. Specifically, the areas that we intend to investigate in the remaining 16 month of this PhD are as follows:

- **Spatio-Temporal Inference.** To fully meet requirement 2 of spatio-temporal inference of untrustworthy data, we need to include the time dimension in the trustworthiness evaluation in combination with the spatial inference performed by our TrustHGP model. This will allow us to characterise the dynamic trust behaviour of a user. For example, a user can be untrustworthy only in selected time windows or it can be particularly inaccurate only when observing the target from certain locations. In this context, the GP framework used by the TrustHGP is sufficiently flexible to integrate time analysis and spatial inference. However, to doing so, research must address the issue of representing trustworthiness as a temporal function in the GP model.

- **Active Learning (AL)-Driven Incentives.** Another requirement of this research lies at the intersection of the two areas of incentive engineering for crowd-based interactions and active learning from crowd reported information. In more detail, the use of incentives in crowdsourcing applications is typically targeted at motivating users to exert the required effort in executing their assigned micro-task and to deviate from the selfish behaviour of executing tasks with the minimum effort to maximise their reward. In this space, Endriss et al. (2011) claim that the principle of an effective incentive strategy for rational agents is to design rewards such that incentives can balance the cost for an agent to deviate from its best utility action to take another lower utility action that is required to achieve system-wide desirable outputs. On the other hand, incentives must be driven by

active learning on the data. Specifically, online predictions based on the incoming reports from the crowd must guide the decision of where to take the next ground observation based on the regions where there is higher uncertainty (Quinn et al., 2011). Drawing these two points together, we envisage that an important area for further research will focus on combining active learning and incentive engineering to motivate crowds to provide requested information.

- **Reinforcement Learning (RL) for Task Verification.** In Section 1.1, we briefly discussed the role of verification as a means for ensuring data accuracy and we also introduced the challenges implied by the verification of crowdsourced data. However, a concrete work on this problem has not yet been undertaken in this research. Therefore, in our future work, we will consider possible directions for researching the data verification problem. In particular, we will explore the idea of using RL to learn budget-limited policies for verifying data in a crowdsourcing process. In fact, a good strategy of verification must accommodate the cost of requiring extra information needed to verify a reported opinion and its tradeoff with the accuracy that is to be increased, given the budget constraint. In this direction, we will research efficient verification strategies for crowdsourced data to improve information quality in this setting.

To conclude, Figure 5.1 shows the timetable for the schedule of the reseach activities listed above until the completion of the PhD, including also the following background tasks:

- **Conference papers**: We aim to submit the results of our work to the research communities from international AI and multi-agent systems conferences. In more detail, we plan to submit a full paper describing our trust-based data fusion work (Chapter 3) to the AAMAS 2013 conference (`aamas-conference.org`) and to detail the contribution of our trust-based heteroscedastic Gaussian process (Chapter 4) in a full paper to be submitted to the IJCAI 2013 conference (`ijcai.org`). In addition, the future work on task verification and active learning will be targeted for publication in the AAAI 2013 conference (`aaai.org`).
- **Journal paper**: Upon achieving our goals concerning spatio-temporal inference models and active learning-driven incentives, we aim to publish our results in an international AI journal, AIJ (`journals.elsevier.com/artificial-intelligence/`)

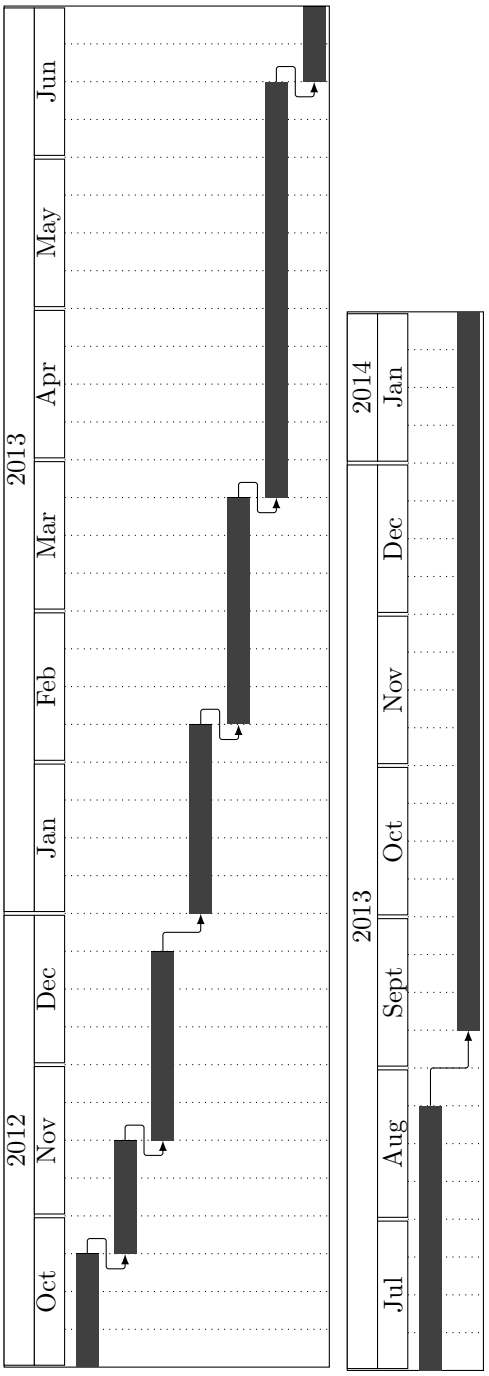


FIGURE 5.1: Gantt chart of the activities scheduled until the PhD thesis submission.

Appendix A

OpenSignalMap Cell Tower Dataset

This appendix describes the crowdsourced dataset of cell detections collected by the OpenSignalMap project that was used in the experiment presented in Chapter 3. The intent of this project is to map cell towers and signal coverage by collecting reports about cell detections submitted by Android devices. In particular, we received a set of 68,714 reports collected in Septemper 2011 which were located in the area of Southampton, UK, bounding box: 50.85 N, 1.25 W and 50.97 N, 1.525 W (see Figure A.1). Each report is described by the following fields:

- **entity_id:** Record identifier.
- **inserted_at:** Timestamp of the detection.
- **device_type:** Model of the device, e.g. HTC Desire, GT-I9000, Nexus S, etc.
- **network_type:** Type of cellular connection: EDGE, GPRS, HSPA, UMTS, Unknown.
- **network_name:** Name of the network operator: Three, O2, Orange, T-Mobile, Virgin, Vodafone, MCP Maritime Com, Unknown.
- **network_id:** A 5 digit identifier of the network operator combining the Mobile Country Code (MCC) (first 3 digits) and the Mobile Network Code (MNC) (second 2 digits): 23410 (O2-UK), 23415 (Vodafone-UK), 23420 (Three), 23430 (T-Mobile), 23433 (Orange-UK), 90112 (Telenoir Maritime Communications), Unknown.
- **roaming:** Flag indicating whether the device is connected via roaming: 1=roaming, -1=non-roaming.
- **my_lat:** Latitude (in degrees) of the device's current location.

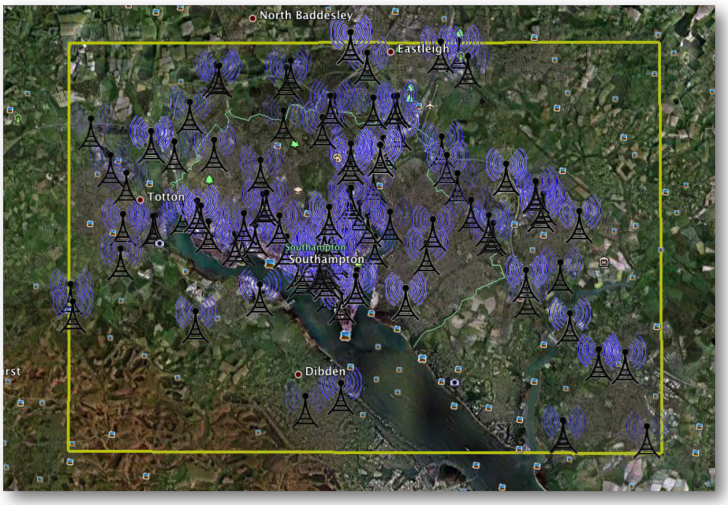
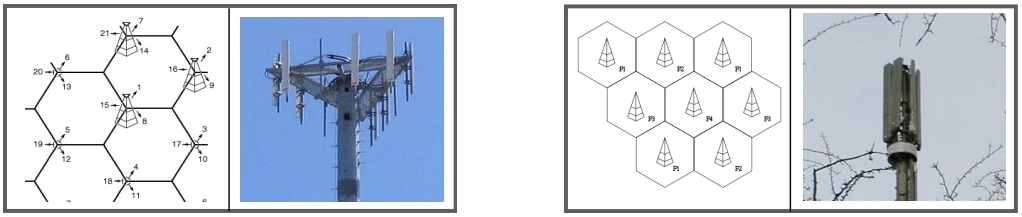


FIGURE A.1: Screenshot showing the bounding box of the Southampton, UK area and the location of the masts (based on the cell_lat and cell_lon fields) tagged within the OpenSignalMaps dataset.

- **my_lon:** Longitude (in degrees) of the device's current location.
- **my_altitude:** Altitude (in meters) at the device's location.
- **location_source:** Flag indicating the positioning system used to discover the device's location: 0=GPS, 1=WIFI.
- **location_inaccuracy:** Precision (in meters) of the location fix.
- **location_speed:** Speed (in meters/seconds) of the device over ground.
- **rssi:** Received signal strength in "Arbitrary Strength Unit" (ASU) (dBm = 2×ASU - 113).
- **CID:** Cell Identifier.
- **LAC:** Local Area Code.
- **cell_lat:** Latitude degrees of the mast location estimated by the OpenSignalMap system (if available).
- **cell_lon:** Longitude degrees of of the mast location estimated by the OpenSignalMap system (if available).
- **app_version:** Version of the OpenSignalMap-Android app used to generate the report.

Specifically, the location inaccuracy had values ranging between 2 and 4930 meters and the received signal strength indication (rssi) between 1 and 99 ASU. In addition, a



(a) (b)

FIGURE A.2: Illustration of the topology and picture of the mast for a directional (a) and an omni-directional (b) cellular network.

considerable number of reports were found to be duplicates and were removed. This duplication was probably generated by the software feature available on the app that enables the device to send reports periodically on behalf of the user, and is likely to generate duplicates when the device is statically in one place. Thus, the dataset was reduced by 66% after filtering, see Table A.1 which shows statistics for before and after removing duplicates. In particular, it shows that the device type was unknown in 60% of the reports and that 53% of the detections came from Vodafone cells. In addition, more than 96% of the reports were sent using 3G mobile connection (GPRS + HSPA + UMTS) and 67% of the devices used GPS for positioning.

Network operator	Num. of reports	(after filtering)	Device type	Num. of reports	(after filtering)
Vodafone	31838	10308	HTC	3728	1455
Orange	10644	3712	Samsung	2903	1056
T-Mobile	10919	3925	Motorola	2480	612
O2	8492	2715	Orange MT	100	26
Three	4609	1794	LG	42	29
Virgin	1122	359	Sony Ericsson	2	1
MCP Maritine Com	1	1	Unknown	59486	20157
Unknown	1116	318			
Network type	**Num. of reports**	**(after filtering)**	**Positioning**	**Num. of reports**	**(after filtering)**
EDGE	2160	711	GPS	66165	22337
GPRS	33252	11725	WIFI	2576	1006
HSPA	26312	8847			
UMTS	6691	1901			
Unknown	325	159			

TABLE A.1: The number of reports, before and after filtering, for network operator, device types, network types and location sources.

Furthermore, the reports tagged a total of 2291 base stations whose locations are shown in Figure A.1. Among these, we were able to reliably identify 157 masts as omni-directional base stations through an on-site, visual inspection.[1] In more detail, the two topologies of cellular networks that are typically adopted for mobile telecommunications based on directional and/or omni-directional radio masts are showed in Figure A.2. In an omni-directional cellular network, the land area is divided into regular hexagonal cell.

[1]In the experiment presented in Chapter 3, we considered only the omni-directional masts with more than 5 reports and this discarded 28 base stations from this group.

A cell tower is placed in the centre of each cell with a set of antennas transmitting and receiving at the assigned cell frequency range. Thus, the signal is radiated approximately spherically (360 degrees angle) across the cell. In a directional cellular network, a cell tower is placed at the corners of each cell and each tower has three sets of directional antennas pointing in different directions with an opening angle of 120 degrees. In this case, a mobile device receives the signal from three different masts within the same cell depending on the nearest corner where it is located. We discussed in Chapter 3 that directional networks are much more difficult to localise from this dataset because the reports do not provide the information about the direction in which the cell tower lies.[2]

[2]Sometimes, an approximate bearing of the cell tower position can be inferred by knowing that the carriers conventionally number the thee sectors of a cell in clockwise order and the sector number is usually indicated by one digit of the CID (e.g. CID=jxxx where j is either 0=omni-directional, 1=south, 2=north-west or 3=north-east). However, we were not able to reliably identify such a digit for each carrier in our data.

Appendix B

COSM Radiation Dataset

This appendix describes the radiation dataset provided by the Cosm sensors located in Japan. In total, the dataset comprises 446 feeds from sensors. The datapoints provided by each sensor are formatted according to the following XML template:

```
<feeds end= "end of period timestamp" start="start of period timestamp">

        <feed id= "Sensor Pachube Identifier" >
        <title> "Sensor name" </title>
        <lat> "Sensor latitude" </lat>
        <lon> "Sensor longitude" </lon>
        <unit>  "Unit of measurement" </unit>
        <elevation> "Sensor altitude" </elevation>

        <datapoints>
                <value at= "timestamp" > "Value" </value>
        </datapoints>
</feed>
```

Specifically, the feeds can classified as follows:

- **Bad Unit**: The unit of measurement is invalid.
- **Unreadable Format**: The feed is reported in an XML that not readable for COSM.
- **Empty Dataset**: The series of datapoints is empty.
- **Bad Values**: The datapoint value is invalid.
- **Single Datapoint**: The series of datapoints has only one value.
- **Multiple Datapoints**: The feeds that report more than one datapoint for their set of measurements. This category of feeds is the one that has been used for performing the experiment presented in Section 4.3.2.

The number of feeds for each of these categories found in this dataset is reported in Table B.1. This data is also showed graphically by the pie chart in Figure B.1.

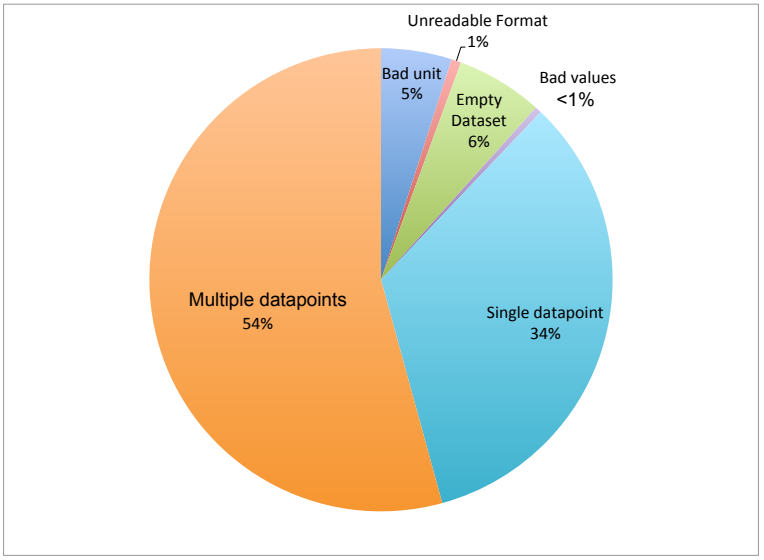


FIGURE B.1: Pie chart of the COSM dataset

	Feeds
Bad Unit	22
Unreadable Format	3
Empty Dataset	27
Bad Values	2
Single Datapoint	150
Multiple Datapoints	242
Total	446

TABLE B.1: Analysis of the Pachube dataset

An example sample of seven feeds taken from the COSM dataset, with the category of each feed indicated in the attached XML comments:

```
<feeds end="2011-08-24T17:38:43Z" start="2011-07-26T17:38:43Z">

        <feed id="29316"> <!-- Multiple Datapoints -->
                <title>Geiger Counter in Hachioji, Tokyo, JPN</title>
                <lat>35.6660131471511</lat>
                <lon>139.317798614502</lon>
                <unit>uSv/h</unit>
                <elevation>4m</elevation>
                <datapoints>
                <value at="2011-08-24T16:14:32.528963Z">0.06666667</value>
                <value at="2011-07-26T22:40:36.575907Z">0.083333336</value>
                <value at="2011-07-27T11:37:32.348107Z">0.13333334</value>
                <value at="2011-07-27T23:14:05.721960Z">0.116666675</value>
                <value at="2011-07-28T11:59:15.094900Z">0.09166667</value>
                <value at="2011-07-28T23:19:07.424571Z">0.083333336</value>
                <value at="2011-07-29T11:41:06.914655Z">0.09166667</value>
```

```
        <value at="2011-07-29T23:12:59.255784Z">0.09166667</value>
        <value at="2011-07-30T11:59:39.987595Z">0.18333334</value>
        <value at="2011-07-30T19:10:39.382951Z">0.116666675</value>
        <value at="2011-07-31T17:53:47.445835Z">0.14166668</value>
        <value at="2011-08-01T11:59:34.912675Z">0.075</value>
        <value at="2011-08-01T22:59:57.363402Z">0.116666675</value>
        <value at="2011-08-02T11:59:23.359377Z">0.10000001</value>
        <value at="2011-08-02T23:14:51.511688Z">0.116666675</value>
        <value at="2011-08-03T11:59:34.050613Z">0.075</value>
        <value at="2011-08-03T23:59:44.540151Z">0.083333336</value>
        <value at="2011-08-04T11:37:58.142513Z">0.075</value>
        <value at="2011-08-04T23:59:12.339526Z">0.13333334</value>
        <value at="2011-08-05T11:59:04.226102Z">0.14166668</value>
        <value at="2011-08-05T20:14:11.982138Z">0.26666668</value>
        <value at="2011-08-06T11:59:17.013373Z">0.13333334</value>
        <value at="2011-08-06T23:59:19.511640Z">0.14166668</value>
        <value at="2011-08-07T08:23:44.017187Z">0.116666675</value>
        <value at="2011-08-07T23:59:04.459241Z">0.10833334</value>
        <value at="2011-08-08T07:52:31.086786Z">0.15</value>
        <value at="2011-08-08T23:59:19.166662Z">0.10833334</value>
        <value at="2011-08-09T11:59:38.688544Z">0.125</value>
        <value at="2011-08-09T20:50:22.391484Z">0.10833334</value>
        <value at="2011-08-13T11:59:37.708707Z">0.10000001</value>
        <value at="2011-08-13T19:01:46.534556Z">0.09166667</value>
        <value at="2011-08-14T11:59:27.203382Z">0.13333334</value>
        <value at="2011-08-14T23:59:44.121526Z">0.09166667</value>
        <value at="2011-08-15T11:59:01.963549Z">0.116666675</value>
        <value at="2011-08-15T23:59:22.979722Z">0.19166668</value>
        <value at="2011-08-16T11:59:40.605287Z">0.14166668</value>
        <value at="2011-08-16T18:01:19.606337Z">0.058333337</value>
        <value at="2011-08-17T08:02:48.710350Z">0.116666675</value>
        <value at="2011-08-17T23:59:17.301783Z">0.13333334</value>
        <value at="2011-08-18T11:19:33.509494Z">0.09166667</value>
        <value at="2011-08-18T21:50:28.700705Z">0.083333336</value>
        <value at="2011-08-19T11:59:52.292633Z">0.10833334</value>
        <value at="2011-08-19T21:55:34.910898Z">0.09166667</value>
        <value at="2011-08-20T11:59:59.405274Z">0.10833334</value>
        <value at="2011-08-20T20:11:57.677557Z">0.083333336</value>
        <value at="2011-08-21T03:59:55.135269Z">0.083333336</value>
        <value at="2011-08-21T23:59:52.765817Z">0.09166667</value>
        <value at="2011-08-22T10:57:21.091489Z">0.125</value>
        <value at="2011-08-22T23:59:53.335037Z">0.083333336</value>
        <value at="2011-08-23T11:59:23.872506Z">0.13333334</value>
        <value at="2011-08-23T16:15:24.313347Z">0.075</value>
        </datapoints>
</feed>


<feed id="25342"> <!-- Bad Unit -->
        <title>radiation in Mitaka, Tokyo</title>
        <lat>35.7015333818623</lat>
        <lon>139.559712409973</lon>
        <unit>?Sv/h</unit>

        <datapoints>
        <value at="2011-06-26T14:36:47.427950Z">0.318</value>
        </datapoints>
</feed>
```

```
        <feed id="29324"> <!-- Single Datapoint -->
                <title>Radiation @ Futomi</title>
                <lat>43.1882581168454</lat>
                <lon>141.438689608967</lon>
                <unit>uSv/h</unit>
                <elevation>0</elevation>

                <datapoints>
                <value at="2011-07-16T04:47:37.376689Z">3.39</value>
                </datapoints>
        </feed>

        <feed id="25885"> <!-- Empty Dataset -->
                <title>Airborn radiation on 4F roof in Arakawa, Tokyo (uSv/h)</title>
                <lat>35.7305931286104</lat>
                <lon>139.79763507843</lon>
                <unit>uSv/h</unit>
                <elevation>12</elevation>

                <datapoints>
                </datapoints>
        </feed>

        <feed id="26485"> <!-- Multiple Datapoints -->
                <title>Mejiro Radiation Meter</title>
                <lat>35.7203154126837</lat>
                <lon>139.701633453369</lon>
                <unit>uSv/h</unit>
                <elevation>33.89</elevation>

                <datapoints>
                <value at="2011-08-24T16:38:34.356060Z">0.130</value>
                <value at="2011-07-26T23:59:12.132096Z">0.138</value>
                </datapoints>
        </feed>

        <feed id="22524"> <!-- Bad Values -->
                <title>Monitoring data at Fukushima Daiichi Nuclear Power Stations: MP-1</title>
                <lat>37.441609604785</lat>
                <lon>141.028575897217</lon>
                <unit>uSv/h</unit>

                <datapoints>
                <value at="2011-06-12T12:00:00.000000Z">?????????</value>
                </datapoints>
        </feed>

        <feed id="25972"> <!-- Single Datapoint -->
                <title>Geiger Counter Feeds from Fukushima,JAPAN</title>
                <lat>37.5577104682266</lat>
                <lon>139.85312461853</lon>
                <unit>uSv/h</unit>
                <elevation>182</elevation>

                <datapoints>
                <value at="2011-08-24T16:38:06.617218Z">0.217</value>
                </datapoints>
        </feed>
</feeds>
```

References

Shane Ahern, Marc Davis, Simon King, Mor Naaman, and Rahul Nair. Reliable, user-contributed gsm cell-tower positioning using context-aware photos, 2006.

N. Archak and A. Sundararajan. Optimal design of crowdsourcing contests. *ICIS 2009 Proceedings*, 200, 2009.

Y. Bachrach, T. Graepel, T. Minka, and J. Guiver. How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing. *Arxiv preprint arXiv:1206.6386*, 2012.

C.M. Bishop. *Pattern recognition and machine learning*, volume 4. Springer New York, 2006.

N. Black and S. Moore. Gauss-seidel method. *From MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld. wolfram. com/Gauss-SeidelMethod. html*, 2006.

Ed Borden. Crowdsourcing data accuracy, 2011. URL: http://blog.cosm.com/2011/06/crowdsourcing-data-accuracy.html.

M.M. Breunig, H.P. Kriegel, R.T. Ng, J. Sander, et al. Lof: identifying density-based local outliers. *Sigmod Record*, 29(2):93–104, 2000.

A.P. Dawid and A.M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, pages 20–28, 1979.

M.H. DeGroot. *Optimal statistical decisions*, volume 82. John Wiley & Sons, 2004.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

U. Endriss, S. Kraus, J. Lang, and M. Wooldridge. Incentive engineering for boolean games. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2602–2607. AAAI Press, 2011.

T. Gneiting and A.E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

P.W. Goldberg, C.K.I. Williams, and C.M. Bishop. Regression with input-dependent noise: A gaussian process treatment. *Advances in neural information processing systems*, 10:493–499, 1997.

P. Groot, A. Birlutiu, and T. Heskes. Learning from multiple annotators with gaussian processes. *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 159–164, 2011.

K. Guan, S. Dehnie, L. Gharai, R. Ghanadan, and S. Kumar. Trust management for distributed decision fusion in sensor networks. In *Information Fusion, 2009. FUSION'09. 12th International Conference on*, pages 1933–1941. IEEE, 2009.

L.A. Hageman and D.M. Young. *Applied iterative methods*. Dover Pubns, 2004.

D.L. Hall and J.M. Jordan. *Human-centered information fusion*. Artech House Publishers, 2010.

D.M. Hawkins. *Identification of outliers*, volume 11. Chapman and Hall London, 1980.

J. Heinzelman and C. Waters. Crowdsourcing crisis information in disaster-affected haiti. *Special report (United States Institute of Peace)*, page 252, 2010.

Panos Ipeirotis. Worker evaluation in crowdsourcing: Gold data or multiple workers?, 2010. URL: http://www.behind-the-enemy-lines.com/2010/09/worker-evaluation-in-crowdsourcing-gold.html.

S.J. Julier and J.K. Uhlmann. General decentralized data fusion with covariance intersection (ci). *Handbook of Data Fusion*, 2001.

E. Kamar, S. Hacker, and E. Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.

E. Kamar and E. Horvitz. Incentives and truthful reporting in consensus-centric crowdsourcing. Technical report, Technical report, MSR-TR-2012-16, Microsoft Research, 2012.

L.M. Kells, W.F. Kern, and J.R. Bland. *Plane and spherical trigonometry*. McGraw-Hill, 1951.

K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard. Most likely heteroscedastic gaussian process regression. In *Proceedings of the 24th international conference on Machine learning*, pages 393–400. ACM, 2007.

J. Kohonen and J. Suomela. Lessons learned in the challenge: making predictions and scoring them. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 95–116, 2006.

S. Kullback and R.A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

M. Lázaro-Gredilla and M. Titsias. Variational heteroscedastic gaussian process regression. In *Proceedings of the 28th international conference on Machine learning*, 2011.

M. Momani, S. Challa, and R. Alhmouz. Bayesian fusion algorithm for inferring trust in wireless sensor networks. *Journal of Networks*, 5(7):815–822, 2010.

D. Oleson, A. Sorokin, G. Laughlin, V. Hester, J. Le, and L. Biewald. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Proc. HComp*, 2011.

J.A. Quinn, K. Leyton-Brown, and E. Mwebaze. Modeling and monitoring crop disease in developing countries. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

J. Quinonero-Candela, C. Rasmussen, F. Sinz, O. Bousquet, and B. Schölkopf. Evaluating predictive uncertainty challenge. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 1–27, 2006.

S.D. Ramchurn and N. R. Jennings. Trust in agent-based software. In R. Mansell and B.S. Collins, editors, *Trust and Crime in Information Societies*, pages 165–204. Elgar Publishing, 2005. URL: http://eprints.soton.ac.uk/260823/.

C. Rasmussen. Gaussian processes in machine learning. *Advanced Lectures on Machine Learning*, pages 63–71, 2004.

V.C. Raykar, S. Yu, L.H. Zhao, G.H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *The Journal of Machine Learning Research*, 11:1297–1322, 2010.

S. Reece and S. Roberts. Generalised covariance union: A unified approach to hypothesis merging in tracking. *Aerospace and Electronic Systems, IEEE Transactions on*, 46(1): 207–221, 2010.

S. Reece, S. Roberts, C. Claxton, and D. Nicholson. Multi-sensor fault recovery in the presence of known and unknown fault types. In *Information Fusion, 2009. FUSION'09. 12th International Conference on*, pages 1695–1703. IEEE, 2009.

Y. Saad and Y. Saad. *Iterative methods for sparse linear systems*, volume 20. PWS publishing company Boston, 1996.

B.W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–52, 1985.

E. Snelson and Z. Ghahramani. Variable noise and dimensionality reduction for sparse gaussian processes. In *22nd Conference on Uncertainty in Artificial Intelligence, Boston (USA)*, 2006.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, 2001.

Bryan Walsh. After the quake comes the disease, can haiti cope?, jan 2010. URL: http://www.time.com/time/specials/packages/article/0,28804,1953379_1953494_1953675,00.html.

Watchboard. Perry authorizes more border security funding, virtual border watch program, jun 2006. URL: http://governor.state.tx.us/news/press-release/4909/.

P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *Neural Information Processing Systems Conference (NIPS)*, volume 6, page 8, 2010.

J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems*, 22:2035–2043, 2009.

P. Wolfe. Convergence conditions for ascent methods. *SIAM review*, pages 226–235, 1969.