UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL SCIENCE AND ENGINEERING

Electronics and Computer Science

Trust-Based Algorithms for Fusing Crowdsourced Estimates of Continuous Quantities

by

Matteo Venanzi

Thesis for the degree of Doctor of Philosophy

August 2014

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCE AND ENGINEERING
Electronics and Computer Science

Doctor of Philosophy

TRUST-BASED ALGORITHMS FOR FUSING CROWDSOURCED ESTIMATES OF CONTINUOUS QUANTITIES

by Matteo Venanzi

Crowdsourcing has provided a viable way of gathering information at unprecedented volumes and speed by engaging individuals to perform simple micro–tasks. In particular, the crowdsourcing paradigm has been successfully applied to participatory sensing, in which the users perform sensing tasks and provide data using their mobile devices. In this way, people can help solve complex environmental sensing tasks, such as weather monitoring, nuclear radiation monitoring and cell tower mapping, in a highly decentralised and parallelised fashion. Traditionally, crowdsourcing technologies were primarily used for gathering data for classifications and image labelling tasks. In contrast, such crowd–based participatory sensing poses new challenges that relate to (i) dealing with human–reported sensor data that are available in the form of continuous estimates of an observed quantity such as a location, a temperature or a sound reading, (ii) dealing with possible spatial and temporal correlations within the data and (ii) issues of data trustworthiness due to the unknown capabilities and incentives of the participants and their devices. Solutions to these challenges need to be able to combine the data provided by multiple users to ensure the accuracy and the validity of the aggregated results.

With this in mind, our goal is to provide methods to better aid the aggregation process of crowd–reported sensor estimates of continuous quantities when data are provided by individuals of varying trustworthiness. To achieve this, we develop a trust–based information fusion framework that incorporates latent trustworthiness traits of the users within the data fusion process. Through this framework, we develop a set of four novel algorithms (MaxTrust, BACE, TrustGP and TrustLGCP) to compute reliable aggregations of the users' reports in both the settings of observing a stationary quantity (MaxTrust and BACE) and a spatially distributed phenomenon (TrustGP and TrustLGCP). The key feature of all these algorithm is the ability of (i) learning the trustworthiness of each individual who provide the data and (ii) exploit this latent user's trustworthiness information to compute a more accurate fused estimate. In particular, this is achieved by using a probabilistic framework that allows our methods to simultaneously learn the fused estimate and the users' trustworthiness from the crowd reports.

We validate our algorithms in four key application areas (cell tower mapping, WiFi network mapping, nuclear radiation monitoring and disaster response) that demonstrate the practical impact of our framework to achieve substantially more accurate and informative predictions compared to the existing fusion methods. We expect that results of this thesis will allow to build more reliable data fusion algorithms for the broad class of human–centred information systems (recommendation systems, peer reviewing systems, student grading tools, etc.) that are based on making decisions upon subjective opinions provided by their users.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

$\propto$	Proportional–to operator
$\sim$	Sampling operator
$\mathcal{N}(\cdot,\cdot)$	Normal distribution
Poisson$(\cdot)$	Poisson distribution
$U[\cdot,\cdot]$	Uniform distribution
i	Item index
k	User index
j	Class index
o	Observation index
s	Iteration index of a looping algorithm
d	Input dimension of a continuous function
M	Number of items
F	Number of emergency categories
K	Number of users
J	Number of classes
N	Number of observations
$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of natural numbers
$\boldsymbol{C}$	Set of discrete labels
$\mathbf{R}$	Set of the crowd reports
$\boldsymbol{\Sigma}$	Covariance matrix of the Gaussian distribution
μ_i	True class of item i
$\pi_{jj'}^{(k)}$	Probability of user k of reporting the label j' for an item of class j
$\boldsymbol{\pi}_j^{(k)}$	Probability vector of user k in labelling an item of class j
$c_i^{(k)}$	Label reported by user k for item i
p_j	Probability of a generic item to belong to class j
f	Gaussian process latent function
x	Input variable of f
X	Set of inputs of f
y	Output variable of f
ϵ	Noise parameter of y
$\hat{y}$	True value of f

σ^2	Variance parameter
θ	Precision parameter
t	Trust parameter
$\boldsymbol{\mu}$	Mean parameter
$\boldsymbol{e}_{cons}^{(s)}$	Consensus estimate computed by RM at the s–th iteration
$\boldsymbol{e}$	Gaussian estimate with mean and precision parameter
$\hat{\boldsymbol{e}}_{CI}$	Fused estimate of Covariance Intersection
$\hat{\Sigma}_{CI}$	Fused precision matrix of Covariance Intersection
$\hat{\boldsymbol{e}}_{CU}$	Fused estimate of Covariance Union
$\hat{\Sigma}_{CU}$	Fused precision matrix of Covariance Union
β	Threshold parameter of the Reece method (RM)
μ_{cons}	Mean of the consensus estimate computed by RM
θ_{cons}	Precision of the consensus estimate computed by RM
t_{cons}	Trustworthiness of the consensus estimate computed by RM
acc	Accuracy threshold parameter of RM
$epochs$	Number of training epochs of RM
F_s	Fisher ratio
$E(\cdot)$	Expected value of a random variable
$Cov(\cdot,\cdot)$	Covariance of a pair of random variables
$K(\cdot,\cdot)$	Covariance function of the Gaussian process
$m(\cdot)$	Mean function of the Gaussian process
$\lambda(\cdot)$	Intensity function of the Poisson distribution
$z(\cdot)$	Logarithmic function of λ

Acknowledgements

First and foremost, I owe sincere gratitude to my supervisors Nick Jennings and Alex Rogers. Nick has given me the invaluable opportunity to undertake this research in an internationally acclaimed research group under his guidance. He has always been supportive and has created the best work environment around me to pursue my research. Alex has provided me with precious insights, he encouraged me all the way and he has always been there when I needed him. Thanks both for helping me make the most out of my PhD.

I am also grateful to my mentors at the Microsoft Research Cambridge Lab: John Guiver, Gabriella Kazai, Pushmeet Kohli and Milad Shokouhi, for their kindness and patience in walking me through the beauty of machine learning and information retrieval. Thanks goes to Emine Yilmaz, Katja Hofmann and Di Wu and all the members of the Infer.NET team, for offering their invaluable professional and emotional support in the last mile towards the completion of this thesis.

I also thank my co-authors and all my past and present colleagues and friends of the Agents, Interactions and Complexity group of the University of Southampton, particularly Long Tran–Thanh,Victor Naroditskiy, Gopal Ramchurn, Lampros Stavrogiannis, Muddaser Allam, Ramachandra Kota, Ruben Stranders, Sam Miller, Oliver Parsons, Sebastian Stain, Maria Polukarov and Luke Teacy for taking part to insightful discussions around my work and all the fun time spent together.

I wish to thank my former mentors: Daniele Nardi, Michael Wooldridge, Rino Falcone and Cristiano Castelfranchi, with whom I had excellent and enjoyable work experiences prior to my PhD that helped me become better at what I do.

Thanks to Dafni Anna Boula and her family for taking care of me with their encouragements and smiles during the ups and downs of my PhD journey.

Finally, I would like to thank my parents Anna and Vincenzo and my sisters Marta and Cecilia for their continued support and love that allowed me to do what I have made so far. To them I dedicate this thesis.

This thesis work was funded by the UK Research Council through the ORCHID project, grant EP/I011587/1 and the Electronic and Computer Science PhD fellowship programme of the University of Southampton.

Declaration of Authorship

I, Matteo Venanzi, declare that the thesis entitled *Trust-Based Algorithms for Fusing Crowdsourced Estimates of Continuous Quantities* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as: (Venanzi et al., 2013b) and (Venanzi et al., 2013a).

Signed:.....

Date:.....

Chapter 1

Introduction

Over the last decade, *crowdsourcing* has emerged as a revolutionary way to provide data at unprecedented volumes and speed by harvesting the power of human computation taken into the loops of computer systems (Law and Ahn, 2011). First introduced by Jeff Howe (2006), crowdsourcing is the outsourcing of micro-tasks through an open call to the undefined network of the 2.4 billion internet users[1] (34.3% of the global population). These tasks are typically jobs that are still hard to solve for computers, yet, they are simple and easy to perform even for untrained human users. These jobs include activities such as classifying images, rating objects, or sensing the environment. Following this principle, crowdsourcing has become a viable way to provide fast and inexpensive information gathering services where, collectively, such crowd generated data can facilitate large–scale and highly decentralised information gathering more efficiently and at less cost than can typically be achieved by a single individual or organisation.

In general, the crowdsourcing model is based on a web-based interaction between two types of agents. The first, *task requestor*, agent is a company, an organisation or a single individual who wishes to outsource a number of tasks to the public. The second, *task executor*, agent is a user who responds to these tasks and provides the requested outputs. This information exchange typically occurs either on a monetary basis, i.e., a contracted reward is paid by the task requestor to the users for each response, or a voluntary basis, i.e., the task executors perform tasks motivated by an intrinsic or social reward. To support access of the task requestors to crowdsourcing marketplaces, there are now a number of web platforms, such as Amazon Mechanical Turk (`www.mturk.com`), Crowdflower (`www.crowdflower.com`) and oDesk (`www.odesk.com`) that allow task requestors to automatically post tasks, collect the answers and pay task executors.

The success of crowdsourcing has followed the growth of the internet population and the time that people dedicate to internet activities[2]. Following this trend, many companies

[1]Source: Internet World Stats `www.internetworldstats.com` (2014)

[2]A recent survey stated that U.S people spent an average of 11.6 hours per week online (October 2013). Source: `www.businessinsider.com`

and corporations are increasingly crowdsourcing parts of their everyday business operations to profit from low cost crowd labour and parallelised crowd–powered information services. Examples of successful crowdsourcing applications range from web information retrieval (see the Bing vs. Google crowdsourcing search relevance judgments campaign, `www.bingiton.com`) to fashion design (see `www.threadless.com`) and general problem solving (see `www.innocentive.com`).

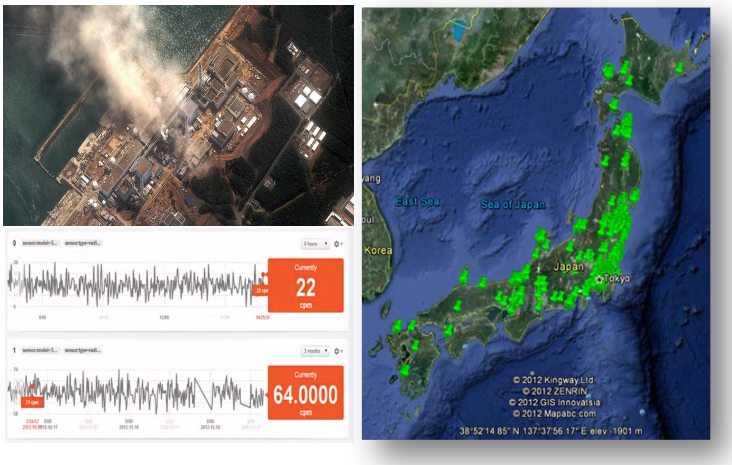
In particular, an important application area of the crowdsourcing paradigm is *participatory sensing*, which is centred on the use of crowdsourced data to solve highly distributed environmental monitoring tasks (Burke et al., 2006). In participatory sensing, task executors perform sensing tasks, by providing environmental data using readily available sensors such as microphones, cameras and Global Positioning Systems (GPS), that are embedded in their smartphones. Technologically, this paradigm is particularly driven by the reality of the 2.1 billion people (29.5% of the global population, 72.4% of the internet's users)[3] that currently have access to the internet through mobile devices. By using their smartphones as an on-board sensor platform, people participate in a ubiquitous sensor network that is able to provide valuable data from remote areas quickly and cost effectively. Importantly, participatory sensing moves crowdsourcing towards the new perspective of using the crowd to collect not only simple piece of information, such as discrete image labels or numeric object ratings, as in the traditional view of crowdsourcing tasks (Whitehill et al., 2009; Welinder et al., 2010; Tran-Thanh et al., 2013). Rather, crowds can now also provide complex data consisting of sensor estimates that represent *continuous quantities*, such as location estimates, signal strength readings or nuclear radiation estimates (Pino and Pezoa, 2012; Yasuhiko, 2011). Collectively, such crowd generated sensor data can help task requestors solve challenging tasks such as globally estimating complex continuous quantities represented by signal coverage maps, disaster maps and nuclear radioactivity maps, over large geographical areas and time periods.

Successful participatory sensing applications focusing on the sensing of continuous quantities have started to emerge. Firstly, a number of companies, including Google, Microsoft–Nokia and OpenSignal, are involved in building signal coverage maps and cell-tower maps from signal strength readings provided by the crowd of mobile devices connected to the network (Figure 1.1a). Using this technology, it has been possible to produce cell-phone signal coverage maps, cell tower maps and WiFi hotspots maps for over 200 countries with the participation so far of more than 2 million users contributing data[4]. Secondly, a key application area of participatory sensing is *disaster response*, where crowdsourcing technologies are becoming an important way to gather live emergency information from local responders. Some example of crowdsourcing and participatory sensing applications are illustrated in Figure 1.1. In particular, a notable example is

[3]Source: The MobiThinking `www.mobithinking.com` (June 2013)
[4]Source: OpenSignal `www.opensignal.com`

(a) Cell-tower, WIFI hotspots and mobile signal coverage mapping

(b) Haiti emergency mapping (2010)



(c) Fukushima radiation monitoring (2011)

Figure 1.1: Some examples of crowdsourcing and participatory sensing applications.

the Haiti-Ushahidi crowdsourced disaster mapping that took place after the devastating earthquake in Haiti, 2011. An open-source crisis mapping platform was set up by the Ushahidi team (`www.ushahidi.com`) to allow people to fill a map with reports of the estimated locations of disaster events around their area, such as trapped persons or damaged buildings. This created a live disaster map with more than $600,000$ emergency reports that became a key resource for the first responders to coordinate their rescue operations (Figure 1.1b). In a similar scenario, during the nuclear emergency in Japan subsequent to the 2011 Fukushima disaster, 557 Geiger counters were deployed by private individuals to collect live radiation sensor readings to help monitor the spread of the nuclear cloud (Figure 1.1c). Many of these sensors were based on open-hardware boards such as Arduino (`www.arduino.cc`) or low cost computers such as Raspberry pi (`www.raspberrypi.com`). This entirely crowdsourced sensor network came to life in less than two weeks after the disaster and became a key resource for the public to gather live nuclear radiation data from the contaminated areas (Teraguchi et al., 2011).

A number of key aspects emerging from these scenarios, and from many other besides, are important for the scope of our work. The first aspect is that, when crowdsourcing sensor estimates, we are facing a new scenario in which the crowd reports sensor estimates of continuous quantities as opposed to discrete data. This means that each report contains information about the uncertainty of the user surrounding the reported value. We can then say that, in participatory sensing settings, the reports include both a reported value and the precision of such a value, i.e., the *reported uncertainty* of the user (Quinonero-Candela et al., 2006). For example, it is common for users to report the precision of each observation as a confidence value estimated through self-appraisal, as the precision of the measuring tool or as the variance of a series of repeated measurements as part of their reports. In particular, when reporting GPS data, the precision of the location is automatically provided by the GPS device itself estimated on the basis of the number and configuration of satellites providing the fix (Brown, 1994).

The second key aspect is the fact that crowdsourcing and participatory sensing paradigms are indistinguishably used for estimating values of both *stationary* and *non-stationary* quantities. More precisely, for the purposes of this work, the concept of stationarity is referred to the value of the item[5] being crowdsourced. That is, stationary quantities are items whose value remains constant, are uncorrelated to any extra dimensions (such as space and time) and are uniquely defined within a specific range. For example, in crowdsourcing applications, stationary quantities are typically location and fix point targets, such as cell-tower locations, the WiFi hotspot locations. By contrast, non-stationary quantities have non–constant values that may vary across one or several dimensions. As a result, the crowd reports related to such quantities are correlated to these dimensions. For example, non-stationary quantities estimated through crowdsourced estimates are

[5]We will use the terms quantity and item interchangeably.

continuous spatial-temporal functions, i.e., functions varying across spatial and temporal dimensions, where the observation values reported by each user depend on a specific location and timestamp. Understanding the different types of the data generated by observing stationary or non-stationary quantities is important for a task requestor to make sense and effective use of such crowdsourced information.

A third challenging aspect of participatory sensing is the fact that the ground truth of the crowdsourced items is typically unknown by the task requestor. Indeed, the main purpose of crowdsourcing processes is usually to allow task requestors to retrieve information about such an unknown item's value by outsourcing the information retrieval to the largest community of potential sources. This is particularly true for all the above mentioned scenarios where, for example, the true positions of the cell-towers are unknown to the crowdsourcing companies, since they do not have access to the official cellular network data that is owned by the network infrastructure providers. Similarly, people involved in tracking radiation levels in Japan did not have knowledge of the true radiation levels over the contaminated areas that they were trying to monitor. Inevitably, the lack of ground truth of the value of such quantities generates a great deal of uncertainty in the crowdsourcing process when the goal of the task requestor is to estimate such true values. To cope with this uncertainty, it is common within crowdsourcing models to introduce one basic assumption, conventionally referred to as the *majority assumption* (Karger et al., 2011; Tran-Thanh et al., 2013; Kamar et al., 2012). This assumption states that, on average, the majority of the crowd reports is somehow related to the ground truth. That is, most of the reports are informative for the correct estimation of the true item's value. In a way, this assumption reflects the task requestor's belief that the crowdsourcing process is overall useful, and not misleading, to learn correct knowledge about the crowdsourced quantities. However, even after introducing this assumption, there is still uncertainty related to how to identify such a majority of good reports within the crowdsourced dataset.

Generally speaking, these aspects highlight a trade–off between the benefits of crowdsourcing in providing large amounts of information produced by the mobilisation of people to report data, and the uncertainty surrounding such information generated by missing ground truth and the individual uncertainty of the reported estimates. In fact, while crowdsourced information can be a key contribution to successfully track stationary and non-stationary targets, the inefficient management of its uncertainty can result in counterproductive outcomes that can potentially invalidate the utility of these tools (see the case of the Boston marathon bombing where crowdsourcing reports of suspects led to identifying the wrong people (Bodden, 2014)) To help address this challenging problem, we identify a key question of interest to task requestors in participatory sensing applications related to how to *fuse* multiple crowdsourced sensor estimates to accurately estimate a generic (i.e. stationary or non-stationary) quantity, without knowing its ground truth. At first glance, this question could be thought of as an instance of a

standard data fusion problem in sensor networks, where the problem of fusing multiple sensor estimates to learn a single output is prominent. In more detail, the sensor fusion problem relates to fusing multiple readings provided by hard sensors under the uncertainty of the possible inaccuracies, e.g., biases, gains and offsets, of individual sensors. To address this problem, many approaches try to model sensor's faults and biases to recover the unbiased readings (Brooks and Iyengar, 1998). However, in our crowdsourcing setting, we are dealing with human-generated sensor readings that may have gain and offset errors that may be correctly calibrated but exhibit greater noise than reported due to the over-confidence of the human user (Hall and Jordan, 2010). They may even have position and timestamp errors with all the above features again, and they may also be subject to malicious reporting. Therefore, as this thesis will show, a potential drawback of the traditional sensor fusion techniques employed in crowdsourcing settings is the difficulty of learning accurate sensor models for a heterogeneous and arbitrarily large system of crowdsourced sensors given the sparse data typically available. Since this issue can have a detrimental effect on the quality of the fusion results, we seek alternative fusion methods that can be more effective for crowdsourcing scenarios by abstracting from specific sensor models. To do so, we shall first discuss the key aspects related to data trustworthiness in crowdsourcing contexts which lie at the foundations of the problems that we address and the solutions that we will present in this thesis.

1.1 Trust Issues with Crowdsourced Information

The openness of crowd systems inevitably exposes the data produced by such systems to issues of uncertain trustworthiness of the single crowd report. This issue relates to the unknown capabilities and individual reliabilities of the task executors and their mobile devices. Collectively, this uncertainty does not allow task requestors to easily identify the reliable content among the set of crowd responses. For example, data trustworthiness issues were reported by the Ushahidi team during the Haiti disaster mapping (Figure 1.1c) where many people maliciously misreported the true needs, category and priority of their emergency. Similarly, in both the signal coverage mapping (Figure 1.1b) and the radiation monitoring scenario (Figure 1.1d), the reliability of the reports varies depending on the user's behaviour as a reporter, as well as the noise and the sensitivity of the device used for taking measurements. Therefore, when drawing conclusions from such data by fusing the crowd reports together in a single output, it is necessary to take these data trustworthiness issues into account to ensure the accuracy and validity of the final results (Hall and Jordan, 2010).

To assist the task requestor in this challenging task of fusing crowdsourced reports, we advocate the role of a third, software, agent, which we call the *output mediator agent*, within crowd-based information systems. As defined by Wooldridge and Jennings (1999), an intelligent software agent is a computer system that is capable of autonomous action

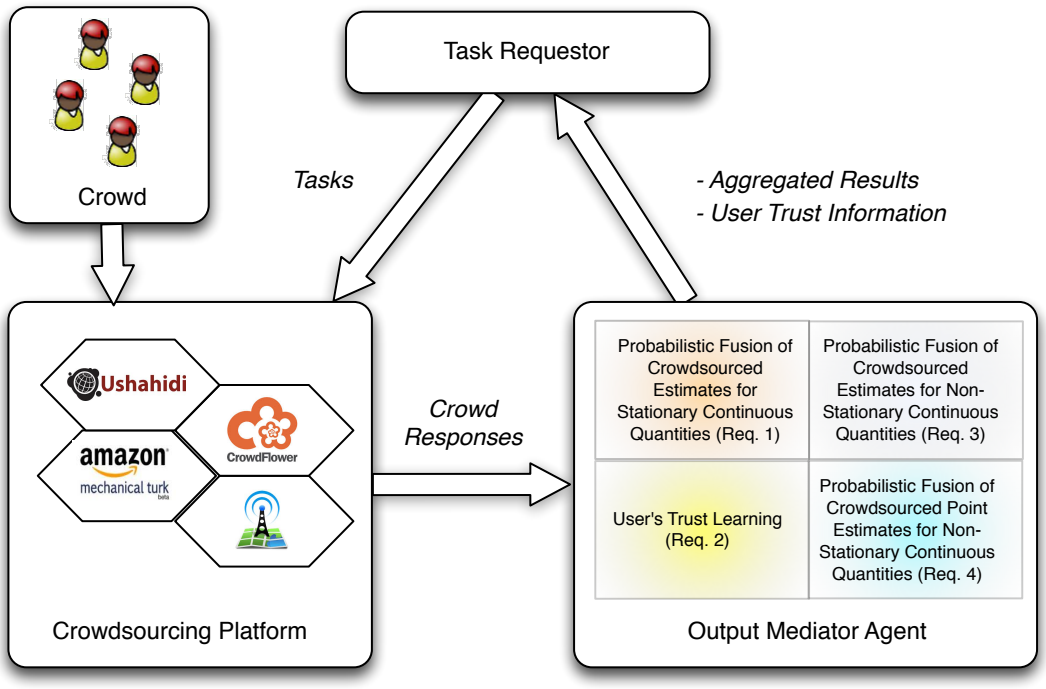


Figure 1.2: The architecture of a crowd–based information system including an output mediator software agent.

within an environment. In particular, an agent is capable of (i) reactive behaviour (i.e. respond to changes in the environment), (ii) proactive behaviour (i.e. take initiatives and goal-driven actions when appropriate) and (iii) social behaviours (i.e. interact with other software and human agents) within this environment. In the context of crowd-based information systems, the primary goal of an output mediation agent is to compute estimates of the item's value by aggregating the set of crowd responses as accurately as possible. Its reactive behaviour relates to the capability of responding to the evolving structure of the information produced by the crowd. Its social behaviour relates to the interaction with the reports produced by other task executor agents while facing the uncertainty about their individual trustworthiness[6]. Figure 1.2 illustrates the architecture of such a system with the output mediator agent bridging the information outputs between the crowdsourcing platform and the task requestor. In particular, the agent must feed back to the task requestor information about (i) the aggregated output and (ii) the user's trustworthiness. In particular, the learning of a user's trustworthiness is required to yield the fusion process to the correct estimation of the item's true value. Furthermore, it is important for task requestors to have knowledge of user's trustworthiness to be able to design and allocate tasks to the best users and so maximise the throughput of their crowdsourcing tasks. To summarise, our output mediator agent must be able to operate in the following setting.

[6]We do not identify any strong proactive behavioural attitude in our output mediator agent.

(i) The crowd responses are given in the form of crowdsourced sensor estimates, i.e. each report contains a reported value and the precision of such a value.

(ii) The true value of the crowdsourced quantity is unknown.

(iii) The trustworthiness of each task executor is unknown.

In this setting, it is important to observe that, due to the lack of ground truth for both the user's trustworthiness and the items' value, the set of crowd responses is the *only* resource for the output mediator agent to recover the most likely aggregated result and the trustworthiness of each user. Given this, the aim of our research is to provide a new set of trust-based fusion algorithms that address the challenging task of aggregating crowdsourced estimates combined with the assessment of the user's trustworthiness. In summary, the key research question that we address in this thesis can be stated as follows:

How to reason about fused outputs and the trustworthiness of individuals in crowdsourcing estimates of stationary and non-stationary continuous quantities?

To address this question, we shall now describe the key requirements for a fusion method suitable to be employed in the design of an output mediator agent.

1.2 Output Mediator Agent Requirements

From our discussion so far concerning the problem of fusing untrustworthy estimates in participatory sensing domains, we outlined the architecture of a crowd-based information system. In this system, we described the role of an output mediator agent which is dedicated to solve the data fusion task (Figure 1.2). Now, such an output mediator agent must be able to merge crowdsourced estimates in the various settings that relate to the variety crowdsourcing settings described in our examples (Section 1). In particular, the agent should be able to perform data fusion tasks equally well for both stationary and non-stationary items. Given this, we identify the following requirements for the design of such an output mediator agent:

Req. 1: **Probabilistic Fusion of Crowdsourced Estimate of Stationary Continuous Quantities.**

The first requirement is to be able fuse crowdsourced estimates for stationary items. This requirement is relevant to the problem of estimating the location of a WiFi hotspot from crowdsourced observations in mobile sensing, or the location of a trapped person from crowdsourced reports in disaster response.

However, we discussed earlier the issue of dealing with uncertainty about the data trustworthiness which makes this requirement particularly challenging to meet. To efficiently deal with this issue, the agent should make use of notions from probability theory as a standard mathematical tool to model uncertainty in data fusion processes. In particular, when computing the fused estimate, a prerequisite for the agent is to provide full information about the uncertainty around the fused value, i.e., it should provide the *predictive uncertainty* of such a value. In fact, having information about the predictive uncertainty is relevant to many decision making tasks where task requestors take actions based on the confidence in the fused value (Quinonero-Candela et al., 2006). For example, in emergency response, a rescue operator (i.e., a task requestor) who is aware of the risk of inefficiently allocating the limited rescue forces, would prefer a prediction saying: "there is 40% probability that a person is trapped in this building" rather than the much less informative statement: "it is likely that there is a person trapped in this building". In these terms, it is important to compute a fused estimate with low uncertainty in order to provide the highest informative contribution to the task requestor's decision making.

Req. 2: **User's Trust Learning.**

The second requirement for the agent is to be able to estimate the trustworthiness of the individual users from the set of crowd responses. We identify this requirement as traversal to both the cases of fusing estimates for stationary and the non-stationary items. That is, each fusion algorithm deployed in crowdsourcing settings should be able to perform user trust learning as a prerequisite to produce most accurate fusions. Hereafter, we will refer to this task of the joint learning of user's trustworthiness and the fused output as the *trust-based fusion* task. For this requirement, due to the lack of knowledge of the ground truth of the user's reliability (Section 1.1), the process of learning such a user's trustworthiness is implicitly driven by the concept of *crowd consensus* (Sheshadri and Lease, 2013). Specifically, from the majority assumption stated earlier (Section 1), the agent can assume that there exists a trustworthy crowd consensus value given by the agreement of an unidentified majority of trustworthy reports. Therefore, such a crowd consensus can be used as evidence for identifying the reliability of each reports and, in turns, assess the trustworthiness of the individual users. Given this, the agent's trust learning mechanism can assess user's reliabilities on the basis of how much the user's reports agree with the consensus of the other users. In doing so, however, a crucial point relates to how to deal with the uncertainty about the crowd consensus value that is also unknown to the agent.

Req. 3: **Probabilistic Fusion of Crowdsourced Estimates for Non-Stationary Continuous Quantities.**

Extending our first requirement, this requirement is for the agent to perform probabilistic fusion of untrustworthy estimates for non-stationary items. Specifically, this requirement is relevant to the applications of participatory sensing of spatial–temporal function such as radiation sensing, temperature sensing, or weather sensing. In these applications, the fusion of the reports must be computed as an estimate of the continuous, spatial-temporal function to represent the non-stationary quantity observed by the crowd. To achieve this, the agent must be capable of assessing the trustworthiness of the individual user with respect to the spatial and temporal features of its reports and of relating such a user's trustworthiness to the values of the aggregated function. For example, the estimated function should be consistent with the value of the trustworthy reports and, at the same time, ignore the value of other untrustworthy reports. In this setting, the report's low or high trustworthiness might be related to specific geographical areas and time ranges that are particularly easy or difficult to observe for the user. Therefore, the capability of analysing the user's trustworthiness from spatial-temporal estimates is required for the agent to perform the trust-based data fusion for non-stationary items.

Req. 4: **Probabilistic Fusion of Crowdsourced Point Estimates for Non-Stationary Continuous Quantities.**

In many cases, crowd reports are given in the form of of point estimates defined as an isolated point (e.g., latitude and longitude coordinates) with an associated precision (e.g., the uncertainty around the reported location) and a submission timestamp. This is the case for the crowdsourced emergency reports of the Haiti scenario (Section 1) where reports from the crowd were geo-located tweets and text messages, with associated GPS accuracy, that people sent to indicate the location and the time of an emergency event. In this setting, the data fusion task for the agent is to recover the spatial-temporal distribution of emergency events based on the *intensities* of the submitted reports. That is, the fused output is produced in the form of a continuous function representing the *expected* number of reports at any point in space and time. In particular, these reports' intensity functions are important to estimate the real distribution of damage in the disaster areas and to allocate rescue resources more efficiently (Goodchild and Glennon, 2010b). By knowing the expected intensity of reports in a certain area, a task requestor could infer the location of new rescue tasks or the boundaries of the new areas to inspect for rescue purposes. Therefore, the output mediator agent must be able to compute aggregations of point estimates in the form of a spatial-temporal intensity function (commonly referred to as spatial-temporal point processes, Diggle 2013) in the typical crowdsourcing setting where there is uncertainty about the trustworthiness of the sources (i.e. the point reporters).

1.3 Research Challenges

From the requirements described in Section 1.2, our research aims to develop new solutions to efficiently tackle the data fusion problem within a number of crowdsourcing and participatory sensing applications. To do this, there are a number of challenges that must be addressed:

1. When simultaneously estimating an item's true value and the trustworthiness of the individual users without having knowledge to the ground truths of such quantities, a challenging aspect for the fusion algorithm is indeed the impossibility to validate the correctness of its learning outputs. In more detail, the fusion algorithm must be able to learn the correct assignments of trustworthiness among the users, and accordingly estimate the fused output in such a way that the trust value reflects the contribution of each user's report to the fused estimate. However, as discussed earlier, this learning task must be performed in an unsupervised setting where there is no access to *gold standards* (i.e. the ground truth) for either the user's trustworthiness or the item's value throughout the learning process.

2. When putting the previous challenge in the context of fusing crowdsourced estimates for non-stationary items, another key challenge is to also take into account extra correlations, e.g., spatial-temporal correlations, of the set of reported estimates within the fusion process. In more detail, the difficulty of the algorithm is to be aware of the dependencies of the reports from the user' actual location and time of submission when processing the value of different reports. In particular, when evaluating the user's trustworthiness, the trust values learned by the algorithm must be related to the consistency of the user's reports with the estimated function over a specific space and time range.

3. The third challenge is to solve the trust-based fusion task with spatial-temporal point estimates. Similarly to the previous challenge, the algorithm must be able to simultaneously perform the user trust learning and the data fusion task with datasets of spatial-temporal point reports. Recall, when dealing with point reports, the fused estimate is typically computed as an intensity function that expresses the expected number of reports at any location and time of interest (Diggle, 2013). However, an extra difficulty of this fusion task lies in the problem of relating user trustworthiness to the intensities of submissions of the user's reports, rather than the reported values themselves. This difficulty is due to the need to analyse correlations between the intensities of reports sent by each user. In this setting, each untrustworthy user can be regarded as a single reporter whose number of submissions does not follow, i.e., is uncorrelated to, the main intensity pattern drawn by the unknown portion of trustworthy users. Given this, the key question is how to estimate the underlying intensity function spatially and temporally and,

at the same time, learn the user's trust values correlated to this function, without having knowledge of either of these quantities.

From these challenges, a number of research communities from various fields, including human computation, citizen science, machine learning and multi-agent systems, have been trying to find solutions to various aspects of the overarching problem. In particular, reasoning about trust under uncertainty to support decision making by autonomous agents has always been a central topic within multi-agent systems research (Castelfranchi and Falcone, 1998). In this respect, a number of computational approaches to support decision making by autonomous agents in situations where sources of information are of varying degree of trustworthiness have been devised. These include trust-based argumentation frameworks, (Parsons et al., 2013), cognitive trust models (Castelfranchi and Falcone, 2010) to probabilistic methods based on the Dempster-Shafer theory (Yu and Singh, 2002). These approaches look at ways of manipulating trust under uncertainty within the artificial minds of intelligent agents with the goal to build reliable multi–agent interactions. In these approaches, however, mechanisms for deriving trust from interactions are needed which is one of the technologies that this thesis aims to provide.

Another line of work on computational models of trust in multi-agent systems focuses on ways of learning the trustworthiness of individuals (Marsh, 1994; Ramchurn et al., 2003; Huynh et al., 2004; Teacy et al., 2006). These models consider a rich set of variables regarding, among others, the context, the competence and the risk of the interactions with the trustees. However, most of this work computes trust based on data acquired from direct observations or past interactions with the potential trustees. In our work, we are primarily concerned with trust evaluation for a undistinguished and heterogeneous set of information sources, where trust is computed by comparing the opinions (i.e., the reports) of an individuals with the opinions of all the individuals. As this appears as a new problem to most of these trust models, our work will look at extending the key concepts of the existing trust models to crowdsourcing settings.

Closer to the sphere of crowdsourcing research, prior work has addressed the problem of fusing crowd reports of discrete quantities, combined with the assessment of a user's trustworthiness, in a number of crowdsourcing applications including, image labelling for medical diagnosis (Dawid and Skene, 1979), natural image classification (Clow and Makriyannis, 2011) and galaxy classification (Kamar et al., 2012). The existing solutions range from simple techniques based on majority voting or weighted majority voting (Tran-Thanh et al., 2013), to methods that take into account factors that affect the data reliability, such as user's trustworthiness and task difficulty (Welinder et al., 2010). However, since all these fusion methods are tailored to discrete data, they cannot easily be applied to continuous data that is a core requirement of our problem (Req. 1). In the research of fusion methods for crowdsourcing continuous data, another line of work has dealt with the fusion of single-value reports in crowdsourcing settings related to various

applications, including IQ testing (Bachrach et al., 2012a), quality scoring (Kazai, 2011) and image rating (Kittur et al., 2008). However, most of the current solutions are based on non-trust based fusion methods that treat the reports as equally trustworthy (Hurley, 2002). By doing so, these solutions fail to meet our requirement of user's trust learning that we identify as a crucial element to achieve accurate fusions (Req. 2). In more detail, these non-trust based fusion methods often overcome the possible reliability issues in the data at the extra cost of gathering more reports from the crowd in order to increase the reports' redundancy and possibly reduce the uncertainty in the fused output. Given this, in this thesis, we will focus on the search of alternative fusion methods that are able to deal with the uncertainty in data trustworthiness by putting in place a more efficient user's trust learning mechanism finalised at providing a more accurate fusion with the *same* dataset, i.e., without forcing the task requestor to necessarily request extra reports from the crowd (see Chapter 2 for more details).

Related to the challenge of fusing estimates of non-stationary continuous quantities, there are a number of methods produced by the research on probabilistic regression and random point processes for spatial-temporal data (Cressie and Wikle, 2011; Brix and Diggle, 2001). In this area, effective data fusion methods are typically based on model-based statistical machine learning approaches such as Kalman filters, Markov random fields, basis function models and Gaussian processes, among others (see Cressie and Wikle (2011) for an overview). However, since these methods are typically designed for general purpose applications, they do not explicitly consider user trustworthiness as a crucial requirement of their model. By doing so, their shortcoming of the fact of explaining the inaccuracies of the data through a homoskedastic, single-variance, noise model that captures the general noise of the crowd reporting process but ignores the individual trustworthiness of the sources. In contrast, by modelling the concept of user trustworthiness within such models, our approach will aim to provide solutions that handle the noise of spatial-temporal data in crowdsourcing settings more efficiently, with the target of improving the accuracy of the fusion.

Against this background, we now detail how we addressed the shortcoming of the current methods and describe the contributions of our research.

1.4 Thesis Contributions

The description of our requirements identifies three different core problems related to managing untrustworthy information in crowdsourcing processes. These problems relate to (i) fusing reported observations of stationary items (Req. 1), (ii) fusing reported observations of non-stationary item's reports and (iii) fusing non-stationary point reports (Req 4), with all these problems combined with learning users' trustworthiness (Req 2). In this context, our work will devise a set of new trust-based fusion algorithms,

	MaxTrust (Chapter 3)	BACE (Chapter 4)	TrustHGP (Chapter 5)	TrustLGCP (Chapter 6)
Probabilistic Fusion of Crowdsourced Estimate for Stationary Continuous Quantities (Req. 1)	+	++	-	-
User's Trust Learning (Req. 2)	+	++	+	+
Probabilistic Fusion of Crowdsourced Estimates for Non-Stationary Continuous Quantities (Req. 3)	-	-	+	-
Probabilistic Fusion of Crowdsourced Point Estimates for Non-Stationary Continuous Quantities (Req. 4)	-	-	-	+

Table 1.1: An overview of our contributions mapped against the requirements of the output mediator agent for crowd–based information systems. The symbol '+' ('++') means that the requirement is (strongly) satisfied by the algorithm. The symbol '-' means that the requirement is not satisfied by the algorithm.

each specialised on addressing one of these problems, that will provide four sets of contributions that gradually address all our requirements.

In more detail, as a common basis for all our contributions, we develop a new user trust model for crowd reported estimates based on uncertainty scaling techniques. We use this model to relate the noise of the reports to the trustworthiness of each user (see Chapter 3 for more details). Using this model, we derive our first contribution that is the first algorithm (MaxTrust) for fusing crowd reports for stationary items (satisfying our Req. 1) using a frequentist learning approach. This algorithm applies a maximum likelihood approach to estimate the values of the users' trustworthiness, and in turn the parameters of the fused estimate (see Chapter 3). Subsequently, we present a second algorithm (BACE) that addresses the same problem of fusing crowd reports for stationary items while now using a Bayesian learning approach (see Chapter 4). Specifically, the use of a Bayesian learning framework within BACE allows us to significantly improve the qualities of MaxTrust in several ways. Firstly, the use of prior probability distributions over the random variables describing the user's trustworthiness and the item's value allows us to account for the uncertainty in the estimation user trustworthiness which improves on the accuracy of MaxTrust's fusion. Secondly, BACE provides a richer set of learning outputs by computing probabilistic estimates of user trustworthiness (in contrast to the single-value estimates computed by MaxTrust), that provide numeric information about the uncertainty around the estimated user's trust values. Subsequently, we present our third contribution that is the first model addressing the problem of fusing estimates for non-stationary items (satisfying Req. 3). This model builds upon the heteroskedastic Gaussian process regression framework (Goldberg et al., 1997), from which, we derive a new trust-based heteroskedastic Gaussian process (TrustHGP) designed to represent individual user's trustworthiness in spatial-temporal regression. Within this model, we provide an inference algorithm that is able to perform

the trust-based fusion task with spatial-temporal crowdsourced estimates. Finally, we address our last requirement (Req. 4) by introducing our fourth contribution consisting of an algorithm (TrustLGCP) for merging untrustworthy point estimates again combined with the assessment of user trustworthiness (see Chapter 6). This model is based on an extension of the standard Log Gaussian Cox Process model (i.e., a non-homogeneous Poisson process model with a random log-intensity function generated by a Gaussian process, Møller et al. (1998)) where we introduce input-dependent noise terms to deal with the individual trustworthiness of the reports. In summary, all our contributions and their mapping against our requirements is reported in Table 1.1. Furthermore, to demonstrate the impact and wide applicability of our work, we evaluate our algorithms in a number of key crowdsourcing applications. These applications are inspired by the scenarios presented earlier in this chapter (Section 1) and consider the applications of cell-tower localisation and WiFi hotspot localisation for evaluating MaxTrust and BACE, respectively, radiation monitoring for evaluating TrustHGP and disaster mapping for evaluating TrustLGCP.

In what follows, we highlight the most salient features our each contribution, along with the numerical results of our experimental evaluations.

1. **A Frequentist Trust Model for Fusing Crowdsourced Continuous Data (Chapter 3):** To address the shortcomings in existing research related to fusion of untrustworthy estimates for crowdsourced stationary items, we present a first frequentist approach to model individual user's trustworthiness (satisfying Req. 2) in the probabilistic fusion of stationary data (satisfying Req. 1). In doing so, we make the following contributions:

    - We present the first approach for jointly fusing untrustworthy estimates of stationary continuous items in crowdsourcing settings. Our approach consists of using unobserved trustworthiness parameters to model user's reliabilities with respect to the the Gaussian noise of their estimates.
    - We derive an efficient inference algorithm (MaxTrust) for our model that implements a Jacobi numeric optimisation scheme to compute maximum likelihood estimates of the trustworthiness parameters, from which the fused estimate is automatically derived.
    - Using the OpenSignal (`www.opensignal.org`) dataset containing cell-tower detections collected from Android mobile phones, we show that our algorithm outperforms four existing methods in both absolute accuracy, gaining up to 22%, and predictive uncertainty, gaining up to 21%. Furthermore, we also show through simulations that our algorithm achieve comparable accuracy with 10% more untrustworthy users within the crowd.

2. **A Bayesian Trust Model for Fusing Crowdsourced Continuous Data (Chapter 4):** To further enhance the performance MaxTrust, we present a second

approach to the problem fusing crowdsourced estimates for stationary continuous items. In detail, we apply a Bayesian treatment to the same MaxTrust model defining prior distributions over the trust parameters and the item's true value and deriving the posterior distributions of such random variables. In doing so, we make the following contributions:

- We present the first Bayesian approach to model the fusion of continuous estimates of stationary items (e.g., locations or fixed values) in crowdsourcing settings. Extending our previous MaxTrust's model, this new model (BACE) is able to (i) integrate prior domain knowledge over an item's value and the trustworthiness of a user, (ii) naturally adapt to online learning and sequential data and (iii), most importantly, to achieve *transfer learning*, whereby the reliability of a participant's reports about one item, can be used as evidence about the reliability of its reports about other items.
- We derive an efficient Gibbs sampling–based algorithm to perform approximate Bayesian inference within our model and compute the posterior estimates of the users' trustworthiness and the items' value from the set of crowd reports.
- Using a second OpenSignal dataset of WiFi hotspot detections collected from Android mobile phones, we shows that BACE outperforms MaxTrust and other three state-of-the-art fusion methods by up to 45%. Furthermore, we show that BACE achieves comparable accuracy to existing methods even with 20% more untrustworthy users through experiments on simulated crowdsourcing scenarios.

3. **A Trust-based Heteroskedastic Gaussian Process Model for Fusing Crowdsourced Spatial Data (Chapter 5):** To address the shortcomings of the existing research related to trust-based fusion for crowdsourcing non-stationary items, we present the first work that models spatial and spatial-temporal data using a heteroskedastic Gaussian process approach[7]. In detail, we make the following contributions:

   - We present the trust-based heteroskedastic Gaussian process: the first model for fusing untrustworthy spatial and spatial-temporal estimates in crowdsourcing settings. This method is based on an integration of our user trust model with the heteroskedastic Gaussian process. From this, we derive a new Gaussian process method that is able to aggregate crowdsourced spatial reports while also learning the individual user's trustworthiness.
   - We show that our method significantly improves the accuracy of the predictions of other GP and HGP methods in an application of crowdsourced

[7]Notice that, although in this thesis we present both the TrustHGP and TrustLGCP in the context of spatial data, the GP framework adopted by these models allows us to easily apply them to both spatial and spatial-temporal data by selecting the appropriate GP kernel. See Chapter 5 for more details.

radiation monitoring using real-world data from the 2011 Fukushima nuclear disaster. In particular, we show that our method outperforms other non-trust based Gaussian process methods by up to 23% in terms of accuracy. We also provide an in-depth analysis of the performance using synthetic data showing that our method achieves performance comparable to other methods with up to 30% more untrustworthy users.

4. **A Trust-Based Log Gaussian Cox Process Model for Fusing Crowdsourced Point Data (Chapter 6):** To address the shortcomings of existing research related to fusing untrustworthy point estimates, we present the first work that models report's categorical trustworthiness in random point processes for crowdsourced information. In more detail, we make the following contributions:

   - We introduce the trust-based Log Gaussian Cox Process (TrustLGCP), the first model for learning random spatial point processes from crowdsourced point estimates. Our method is able to perform the trust-based learning of the spatial intensities of the point process together estimating the trustworthiness of sets of reports with respect to their input features (e.g., categories and types).
   - We show that our TrustLGCP model outperforms the standard, non-trust LGCP through experiments on point process estimations using point reports obtained from simulated crowds. We also demonstrate that our TrustLGCP can efficiently learn intensity maps from crowdsourced emergency reports and also learn the trustworthiness of each emergency category with an application to the Ushahidi dataset collected during the 2010 Haiti earthquake.

A number of these contributions have been presented in refereed publications:

M. Venanzi, A. Rogers, N.R. Jennings. Trust-Based Fusion of Untrustworthy Information in Crowdsourcing Applications *In the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2013, 829-836 (See Chapter 3).

M. Venanzi, A. Rogers, N.R. Jennings. Crowdsourcing Spatial Phenomena Using Trust-Based Gaussian Processes *In the 1st International Conference on Human Computation and Crowdsourcing (HCOMP)*, 2013, 182-189 (See Chapter 5).

Additional publications inspired by this work, whose abstract is given in Appendix D are:

M. Venanzi, J. Guiver, G. Kazai, P. Kohli, M. Shokouhi. Community-Based Bayesian Aggregation Models for Crowdsourcing. *In the 23rd International World Wide Web Conference (WWW)*, 2014. *Best paper runner up.* Microsoft, one of the partners of the ORCHID project (`www.orchid.ac.uk`), has registered the algorithm presented in this paper under a US patent. MS ref: 340522.01.

L. Tran-Thanh, M. Venanzi, A. Rogers, N.R. Jennings (2013) Efficient Budget Allocation with Accuracy Guarantees for Crowdsourcing Classification Tasks. *In the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2013, 901-908.

S. Ramchurn, T. D. Huynh, M. Venanzi, B. Shi. Collabmap: Crowdsourcing Maps for Emergency Planning. *In the 5th Annual ACM Web Science Conference*, 4, (2), 2013, 326-335.

A. Rutherford, M. Cebrian, I. Rahwan, S. Dsouza, J. McInerney, V. Naroditskiy, M. Venanzi, N. R. Jennings, J.R. deLara, E. Wahlstedt, S. U. Miller. Targeted Social Mobilization in a Global Manhunt. *PLoS ONE*, 2013, 8(9): e74628.

H. T. Dong, M. Ebden, M. Venanzi, S. Ramchurn, S. Roberts, L. Moreau. Interpretation of Crowdsourced Activities Using Provenance Network Analysis. *In the 1st International Conference on Human Computation and Crowdsourcing (HCOMP)*, 2013, 78-85.

V. Capraro, M. Venanzi, M. Polukarov, N.R. Jennings. Cooperative Equilibria in Iterated Social Dilemmas. *In, 6th International Symposium on Algorithmic Game Theory (SAGT)*, 2013, 146-158.

M. Venanzi, M. Piunti, R. Falcone, C. Castelfranchi. Facing Openness with Socio Cognitive Trust and Categories. *In the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2011, 400-405.

R. Falcone, M. Piunti, M. Venanzi, C. Castelfranchi, From manifesta to krypta: The relevance of categories for trusting others. *ACM Transactions on Intelligent Systems and Technology (TIST), special issue on Trust in Multi-Agent Systems*, 2011, 1-24.

With our work, we expect to provide an important set of solutions to the problem of crowdsourced data fusion that will help make crowdsourcing tools more reliable and robust for real–world applications.

1.5 Thesis Outline

The remainder of this thesis is structured as follows.

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

In Chapter 3, we introduce our MaxTrust model for the trust-based fusion of crowdsourced location estimates. Following this, we present an empirical evaluation of MaxTrust using both synthetic data and real-world location data with an experiment of cell tower localisation.

In Chapter 4, we introduce our BACE model that improves the performance of MaxTrust through the adoption of Bayesian statistics. We show an empirical comparison of BACE against our set of benchmarks using both synthetic data and real data for WiFi hotspot locations.

In Chapter 5, we present our TrustGP model for the trust-based fusion of spatial data. We also describe the steps of the Bayesian inference to jointly estimate the underlying spatial function and the user's trustworthiness. We then show its application to the crowdsourced radiation monitoring problem using data from the Fukushima nuclear disaster.

In Chapter 6, we present our TrustLGCP model for learning spatial intensity maps from untrustworthy point reports. We show an application of this model to the Ushahidi dataset of crowdsourced emergency reports for the Haiti disaster scenario.

In Chapter 7, we summarise the results of our research and draw our conclusions. We also outline directions for future work to broaden the scope of this research and increase its applicability to a wider spectrum of crowdsourcing applications.

Chapter 2

Literature Review

In this chapter, we review the key background research related to the problem of fusing crowdsourced information outlined in Chapter 1. This background will provide the theoretical basis for the models and the algorithms presented in the subsequent chapters. Specifically, the chapter begins with an overview of research on the topic of trust in information sources (Section 2.1) and computational models of trust in the context of crowdsourcing systems (Section 2.2). Subsequently, we will review the state–of–the–art fusion approaches to crowdsourced information that are relevant to our requirements related to managing crowdsourced estimates of stationary items. Specifically, we will divide the discussion by first considering the class of non-trust based fusion approaches (Section 2.3) and then the class of trust-based approaches (Section 2.4). We consider both the cases of discrete and continuous data for each of these. In the second part, we survey fusion approaches related to crowdsourced information of non-stationary items, particularly in the context of crowdsourcing spatial information. In this respect, we will discuss some state–of–the–art approaches to spatial regression (Section 2.5) and spatial point processes (Section 2.6) that are related to our set of problems.

2.1 Trust in Information Sources

In the previous chapter, we discussed how the problem of making crowdsourced information more reliable is primarily concerned with issues of building and managing trust in information sources. As this topic has been a central component of research in computer science for several decades, we provide an overview of existing trust research that will help us position the work of this thesis within this active research area.

Trust is a widely used concept within diverse research areas of computer science ranging from security and semantic web to multi–agent systems and social sciences. As trust is generally intended as an integral component of many types of interactions within humans

and artificial agents, its definition often differs among researchers and application areas. In this respect, several papers explore in-depth many different aspects of trust and survey a number of definitions that emphasize its role as both a degree of belief and acceptance that allows people to make decisions with the risk of negative consequences (Artz and Gil, 2007; Jøsang et al., 2007; Misztal, 2013). In our work, we follow the widely cited definition from Gambetta (1988), that captures trust as a particular level of the subjective probability with which an agent will perform a particular action. In our crowdsourcing context, this concept may be referred to the action of reporting information reliably.

With this in mind, there is a wide variety of trust literature that aims to find ways and theories to build, establish and manage trust in interactions among mixed teams of individuals. In their survey, Artz and Gil (2007) organise this literature into four majour areas: i) general models of trust, ii) policy–based trust, iii) trust management and iiii) reputation–based trust. The first area comprises models of trust that describe the factors and the conditions that play a role in making trust decisions. A number of works on modelling trust propose ways of assessing trust based on both the *abilities* and *willingness* of the trustee with respect to the outsourced task, and the *external context* of the interaction (Marsh, 1994; Castelfranchi and Falcone, 2010). However, many applications, including crowdsourcing, do not always follow these models due to the difficulty of finding values for some of these variables.

In the area of policy–based trust, many works focus on the use of policies to establish trust based on exchanging credentials with the users (Kagal et al., 2003). For example, a policy is the process of logging into a computer system where a user must provide a valid user name and password in order for the system to trust his/her identity. Along this line, research of trust management addresses the problem in establishing trust by using credentials as that may incur a loss of privacy or control for the users in revealing information (Jøsang et al., 2007). In general, both the approaches to policy–based trust and trust management do not deal with the problem of computing trust from interactions with individuals in a crowdsourcing context. In fact, most of the existing work has focused on the effective manipulation of trust beliefs that are computed in some way. Therefore, we aim to provide new models that can solve the problem of computing trust in a broad range of crowdsourcing applications that is currently not addressed by most of this work.

Finally, more related to our problem is work on reputation–based trust, which looks at ways of trusting users based on direct or indirect interactions with other agents that happen over time. As research on reputation–based trust does address the problem of computing trust from interactions with virtual information providers, we explore it in more detail together with other approaches to trusting crowd users in the next section.

2.2 Computational Approaches to Trusting Crowd Users

We now discuss a number of computational approaches to trust evaluation that we regard as tightly connected to the context of crowdsourcing applications. In particular, we focus our discussion on the three main trust approaches that are commonly adopted by crowdsourcing systems. These are (i) reputation–based trust, (ii) gold data–driven trust and (iii) consensus–based trust.

The first class of approaches to trust evaluation that we review is *reputation–based trust*. These approaches are based on assessing a user's trustworthiness by relying on historical data about previous interactions of the trust assessor with the user (Resnick et al., 2000). This data might include other reputation reports and opinions received from third parties which further strengthen the evidence to support the trust formation (Ramchurn and Jennings, 2005; Pinyol and Sabater-Mir, 2013). In general, the advantage of this approach is that the user's trustworthiness evaluation is supported by some empirical evidence of how the user behaved in the past with the system, either by interacting directly with the trust assessor or with other users. However, one difficulty in employing reputational trust approaches in crowdsourcing stems from the openness of the crowd. That is, participants can join and leave the crowd at any time. In fact, such openness can facilitate *whitewashing*, i.e. a crowd member who anonymises or regenerates its identity, and other forms of attack that can be a hinderance for these trust mechanisms in building reliable user reputations (Feldman et al., 2006). In general, the fact that multiple encounters with the same users in open crowd systems are scarce means that use of reputational trust is not well suited to our problem (Huynh et al., 2006). Moreover, building trust based on historical data in a crowdsourcing setting is exposed to the threat of strategic reporting behaviours by the users who may use their reports to build a deceptive image of their reputation in the eyes of the task requestor (Archak and Sundararajan, 2009). Since strategic behaviour of this kind has been observed in a number of human reports (e.g. in the "Red Balloon" challenge, Naroditskiy et al. (2012), the Boston marathon event, Bodden (2014), and the Tag challenge, Rutherford et al. (2013)), in this work we do not consider reputational trust approaches.

A second approach to trust evaluation is *gold data–driven trust*. This approach aims to identify unreliable users using a set of data about which there is a predefined ground truth, or *gold standard* (Oleson et al., 2011). In more detail, by asking the user to perform a set of gold tasks (i.e., a set of tasks for which a gold standard answer is known), the task requestor can estimate the user's trustworthiness based on the discrepancy between their answers and the correct ones. Currently, many crowdsourcing platforms adopt gold–based mechanisms to provide assurance of data reliability to task requestors. For example, both Amazon Mechanical Turk (`www.mturk.com`) and Crowdflower (`www.crowdflower.com`) offer the feature of specifying gold standards when creating tasks. Also, ReCAPTCHA, a tool that uses human answers to digitalise text, performs the

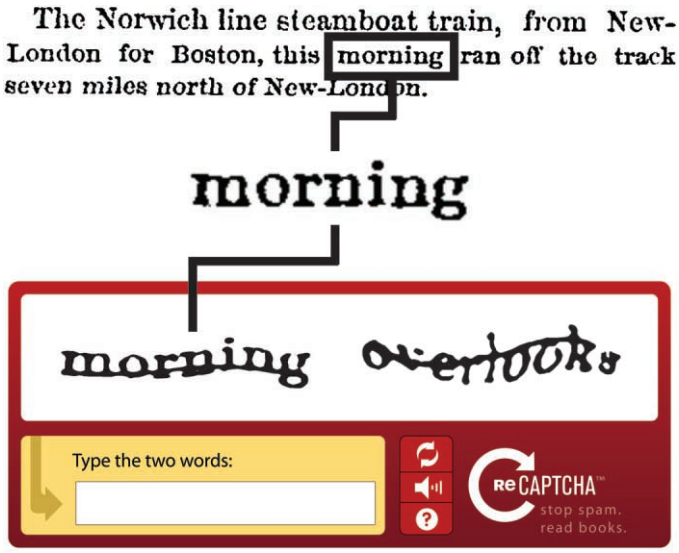


Figure 2.1: Example of gold data-driven trust assessment performed by ReCAPTCHA. The trustworthiness of the users to type the unknown challenge word (morning) is evaluated on the basis of their answer to the known control word (overlooks).

control word test to decide whether an input answer of a user who types an unknown word is trustworthy or not (Figure 2.1). Specifically, the ReCAPTCHA test consists of presenting two words to the user. One of the these words is the actual challenge word, that is unknown, and the other is the control word, that is known. Since the two words are placed in a random order and 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.

In general, this gold–data driven trust approach is meaningful in situations where it is relatively easy for the task requestor to acquire gold standards. The advantage of this approach is that it allows the task requestor to form trust beliefs about the user's abilities based on the evidence of how it has performed on the same task in a real crowdsourcing environment. On the downside, there is an extra cost of training users with gold standards which is not always supported by an absolute guarantee of substantial gain in the quality of the data (Ipeirotis, 2010). That is, even after gaining evidence of the individual user's trustworthiness in performing gold tasks correctly, there is no absolute guarantee that this will convert into a more trustworthiness final outcome. In particular, the users might perform differently on the actual tasks, especially when these are not well aligned to the gold tasks. For this reason, and also for the reasons stated by Req. 2 in which we set our work in the unsupervised setting where gold standards are not available, in this thesis we choose not to consider gold–driven trust methods as a solution to our trust–based fusion problem.

A third approach to trust evaluation is *consensus–based trust*. This involves computing a user's trustworthiness based on the number of other independent observations from other users that match the one reported by the evaluated user. This is a concept formalised by Kamar and Horvitz (2012) under the name of *consensus task*. As introduced in Section 1, this consensus–based trust approach is typically applied when the *majority assumption* holds, i.e. the majority of the crowd's opinions will eventually agree on the ground truth. As a result, the consensus value is likely to reveal such a ground truth value of the crowdsourced task. In general, this consensus-based trust approach to crowdsourcing tasks is supported by the fact that it is relatively easy to create data redundancy, and thus increase the strength of the consensus, by gathering extra reports at low cost. For this reason, many existing models advocate the use of this approach to trust in crowdsourcing settings (e.g., Kamar et al. (2012); Raykar et al. (2010); Whitehill et al. (2009)). However, it is important to notice that, in practice, consensus is easy to compute with discrete data that, for example, is generated from crowdsourced classification tasks. In this setting, consensus can be reduced to a standard voting problem where each reported answer can be seen as a vote on a certain outcome related to the item being classified. Then, the consensus outcome related to the class of that item is chosen as the most voted one. Furthermore, consensus is also fairly easy to compute for simple continuous problems where the consensus value can be taken as the average of the reported values. However, it is less straightforward to compute consensus with crowdsourced sensor estimates which is the problem we address in this work (Req 1). Indeed, the difficulty lies in the fact that the reports are given in the form of continuous probability distributions which do not allow a straightforward averaging or majority voting analysis. Therefore, while considering consensus-based trust as a suitable approach to meet our requirement (Req. 2), we will also look into consensus methods for crowdsourced continuous estimates within the field of data fusion (see Section 2.4). Before this, however, we continue the discussion with the review of more general, non-trust based fusion approaches for crowdsourced data.

2.3 Non-Trust Based Fusion Approaches for Crowdsourced Data

In this section, we discuss approaches that do not attempt to embed learning a user's trust into their data fusion methods. As such, we refer to these as non-trust based fusion approaches. In particular, it is important for us to look at this class of approaches in order to understand the potential of using trust-based fusion in crowdsourcing and also identify suitable non-trust based fusion benchmarks for the evaluation of our approach. Specifically, we now review existing methods from the two main classes of non-trust based fusion approaches to crowdsourcing discrete data (Section 2.3.1) and to crowdsourcing continuous data (Section 2.3.2)

2.3.1 Dealing with Discrete Data

Methods for fusing crowdsourcing discrete data represent the largest body of fusion methods for crowdsourcing that are available to date (see Sheshadri and Lease (2013) for an overview). Generally speaking, these methods are designed for situations in which the set of observations reported by the crowd can be enumerated by a finite set of possible values, or classes. As discussed above, in situations where the majority assumption holds, this fusion problem reduces to the search of the consensus class based on the votes of the crowd. In such cases, majority voting methods are simple ways (yet broadly used in practice) to solve this fusion problem with discrete data.

In more detail, majority voting methods are a frequentist approach to estimating the consensus class from a set of discrete votes. The consensus class is usually estimated as the one that received the highest number of votes. In some cases, there are variants of the standard majority voting method that are applied to more efficiently deal with the uncertainty in the vote distribution, such as ties and situations of weak consensus. In particular, one of these variants is based on taking a random guess between the two (or more) most voted classes, when ties occur. We refer to this method as *semi-weighted majority voting*. For example, assume that the array of the vote counts for an item is $\{3, 2, 3\}$ for the three possible classes $\{A, B, C\}$. Then, using the semi-weighted majority voting, we will select the final class with a uniformly random draw between A and C. More formally, $x \sim \mathit{Discrete}(0.5, 0, 0.5)$ where x is the item's class and Discrete is a probabilistic function with the parameters representing the probability of the item belongong to each class. Alternatively, another majority voting method, *weighted majority voting*, takes into account the full distribution (i.e., it does not ignore the two vote for C) and draws the item's class based on the probabilities defined by the normalised vote counts. That is, in our previous example, this method will draw the final class as $x \sim \mathit{Discrete}(0.37, 0.25, 0.37)$.

Despite the simplicity of the majority voting methods, there are several problems with them when they are applied to crowdsourcing problems. Firstly, the tie breaking rule based on random draws creates instability in the results, especially when there is a high uncertainty due to a low consensus in the vote distribution. In such cases, the task requestor may need to break ties by requesting more votes from the crowd. However, in addition to incurring additional costs, such extra votes do not guarantee to reduce the uncertainty in the vote distribution in situations where the classification task is intrinsically hard to solve. Secondly, this approach implies that all users' votes count with equal weight in the vote distribution, i.e., they are all equally trustworthy. By so doing, it does not account for the different reliabilities of some votes which are based on different levels of trustworthiness (failing to meet our Req. 2). This might, in turn, have a detrimental impact on predicting the final class. Thirdly, as already mentioned, it is non–trivial to define majority voting methods for continuous estimates which are

part of our requirements. For example, when considering the case of spatial-temporal estimates which are part of our Req. 3, we observe that it is not possible to average the estimates when these individually relate to different locations and timestamps.

In general, for the case of discrete data, majority voting methods are typically outperformed by a number of trust-based fusion approaches. That is, approaches that take into account the user's trustworthiness or accuracy, as extensive related work in this area has showed (Raykar et al., 2010; Whitehill et al., 2009; Bachrach et al., 2012a; Welinder et al., 2010; Kamar et al., 2012). In this vein, our work will help further demonstrate the benefits of using trust-based fusion approaches for the case of continuous data. To this end, a prerequisite for us is to identify a suitable majority voting method that is applicable to datasets of continuous estimates which we discuss in the following subsections.

2.3.2 Dealing with Continuous Data

Methods for merging crowdsoured continuous data, which are relevant to our Req. 1, can be found in the literature concerned with data fusion in sensor networks. Specifically, research in this domain studies how to combine estimates from multiple sources to achieve more efficient inference in sensing problems (Thrun et al., 2001). Typically, sensor fusion models consider information sources as physical sensors that are employed, for example, in a target monitoring task where each sensor provides observations of the target in its monitoring area. Then, data fusion algorithms deal with how to aggregate the multiple sensor readings into one single estimate that predicts the target position. In addition, since sensors are noisy, the requirement for such algorithms is to filter the sensor's noise in the fused estimate. Based on the approach taken by these algorithms in modelling the sensor's noise, we can distinguish the two main classes of approaches that relate to non-trust based fusion methods and trust-based fusion methods, respectively. In more detail, this distinction relates to the feature of whether or not the sensor's noise is represented using a trust model. To this end, we will discuss the trust-based sensor fusion approaches later in this chapter (Section 2.4.2). For now, however, we focus on non–trust based methods.

As introduced in Section 1.2, from the traditional data fusion perspective, the human user is primarily considered as an interpreter of the processing result that ultimately transforms the fused estimate into knowledge, and only rarely is input data from human observers considered. However, crowdsourcing introduces a new perspective 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 in the study of the applicability of the current sensor fusion algorithms to human information. In this vein, Hall and Jordan (2010) point out a number of important differences between human and sensor information that needs to be taken into account in the fusion process.

Particularly, they highlight the different types of noise between the two data sources, arguing that the inaccuracies of a sensor reading typically depend on the faults that temporarily or permanently affect the functioning of the sensor, while it is unrealistic to think that the sensor would deliberately misreport its observation as may well 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 data noise features that relate to subjectivity, expertise, bias and other inaccuracies of the human observers.

From this observation, we identify sensor fusion techniques as a suitable methodology for devising fusion methods that are also applicable for crowdsourced data. However, as stated by Req. 1, we need fusion methods that can effectively merge human data. To this end, we now provide an overview of methods for fusing probabilistic estimates by drawing from the two main areas of research that relate to problems of single-hypothesis (i.e., estimates of static values) and the multi-hypothesis (i.e., estimates of non-static values) data fusion. For our purposes, we focus on the two most common methods for these two classes that are (i) covariance intersection for the single-hypothesis fusion and (ii) covariance union for multi-hypothesis fusion.

2.3.2.1 Covariance Intersection

Covariance intersection (CI) is the standard method for fusing a set of probabilistic Gaussian estimates within the single-hypothesis setting (Julier and Uhlmann, 2001). Specifically, this setting assumes that the reported estimates relate to only one correct answer for the item's true value that we wish to estimate. Related to our examples from Section 1.1, a typical case of single-hypothesis fusion is the crowdsourcing of stationary items, such as cell tower mapping or search for victims in disaster response, in which the hypothesis estimated by the fusion relates to the actual location of the tower or the victim. In these cases, CI performs the fusion of the set of estimates in the following manner.

Given two normally distributed estimates, $\boldsymbol{e}_1 = (\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\boldsymbol{e}_2 = (\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, where $\boldsymbol{\mu}$ is the multivariate *mean* and $\boldsymbol{\Sigma}$ is the *covariance matrix*, the CI fused estimate $\boldsymbol{e}_{CI} = (\hat{\boldsymbol{\mu}}_{CI}, \hat{\boldsymbol{\Sigma}}_{CI})$ is computed as:

$$\hat{\boldsymbol{\Sigma}}_{CI}^{-1} = \boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1} \tag{2.1}$$

$$\hat{\boldsymbol{\mu}}_{CI} = \hat{\boldsymbol{\Sigma}}_{CI}(\boldsymbol{\mu}_1\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\mu}_2\boldsymbol{\Sigma}_2^{-1}) \tag{2.2}$$

That is, the parameters of the fused Gaussian estimate are estimated through a linear combination of the means $\boldsymbol{\mu}_i$ weighted by the *precision matrices* $\boldsymbol{\Sigma}_i^{-1}$ (i.e., the inverse covariance matrices). It is easy to notice that, by fusing the estimates as weighted by their individual precision, CI encodes the fact that the observations reported with

higher confidence have a higher contribution to the fused estimate. By so doing, CI takes into account the contribution of the reported uncertainties (i.e. the precision of each report) in a proper way within the fusion, which makes it suitable for us to satisfy our Req. 1. Furthermore, it is important to notice that, by computing $\hat{\boldsymbol{\Sigma}}_{CI}^{-1}$ as the cumulative sum of these reported precisions, the CI fusion will result in a globally higher precision. That is, the CI estimate becomes more confident about its predicted mean value as more reports are added to the fused set. However, the potential issue of doing this is that, without accounting for the trustworthiness of the estimates, the fusion may estimate the true value over confidently or even converge to the wrong value. In this aspect, CI resembles the feature of majority voting in considering all the estimates as equally trustworthy in a continuous space. However, as per the critique raised for majority voting, it is likely that the standard CI fusion will fail to provide the best results in fusing crowdsourced datasets because it ignores the heterogeneous range of data reliabilities. To rectify this, and so address our Req. 2, we need an extension of CI that accounts for a user's trustworthiness in the computation of the fused estimate. This extension will be elaborated upon in Chapter 3.

2.3.2.2 Covariance Union

Covariance union (CU) is the standard method for merging a set of probabilistic Gaussian estimates in the multi-hypothesis case. This case assumes that, due to the variance in the set of reported estimates, there is more than one hypothesis which could be the correct answer. For example, a case of the multi-hypothesis fusion is the setting in which the crowd observes a moving target and reports estimates of its position in different time instants. In this case, a conservative fusion approach to estimate the target position is to merge the reports in such a way that none of the hypotheses are discarded. That is, the fused estimate is computed as the most general output obtained by taking the union of all the reported estimates. Specifically the CU method to unify two Gaussian estimates is described as follows (Reece and Roberts, 2010):

Given two multivariate normally distributed estimates, $\boldsymbol{e}_1 = (\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\boldsymbol{e}_2 = (\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, where $\boldsymbol{\mu}$ is the multivariate mean and $\boldsymbol{\Sigma}$ is the covariance matrix, then the CU estimate $\boldsymbol{e}_{CU} = (\hat{\boldsymbol{\mu}}_{CU}, \hat{\boldsymbol{\Sigma}}_{CU})$ is any Gaussian estimate defined by the following constraint:

$$\begin{cases} \hat{\boldsymbol{\Sigma}}_{CU} & \geq & \boldsymbol{\Sigma}_1 + (\hat{\boldsymbol{\mu}}_{CU} - \boldsymbol{\mu}_1)(\hat{\boldsymbol{\mu}}_{CU} - \boldsymbol{\mu}_1)^T \\ \hat{\boldsymbol{\Sigma}}_{CU} & \geq & \boldsymbol{\Sigma}_2 + (\hat{\boldsymbol{\mu}}_{CU} - \boldsymbol{\mu}_2)(\hat{\boldsymbol{\mu}}_{CU} - \boldsymbol{\mu}_2)^T \end{cases} \tag{2.3}$$

In this definition of CU, the inequalities of Equation 2.3 are defined based on the observation that if $\boldsymbol{A} > \boldsymbol{B}$, then $\boldsymbol{A} - \boldsymbol{B} > 0$, meaning that $\boldsymbol{A} - \boldsymbol{B}$ is positive semi–defined (i.e., it has no negative eigenvalues) (Bochardt and Uhlmann, 2010). More specifically, these

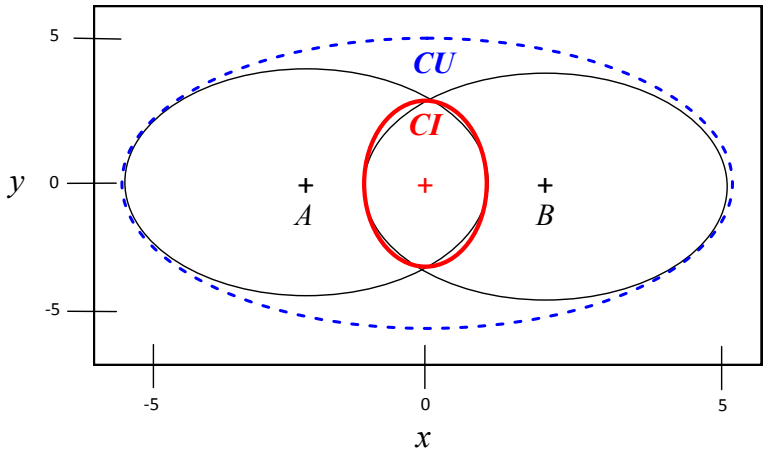


Figure 2.2: Merging two Gaussian estimates, represented by the two circles, using CI (solid line) and CU (dashed line).

inequalities encode the property that if the estimate $\boldsymbol{e}_1$ is consistent, then the translation of $\boldsymbol{\mu}_1$ to $\hat{\boldsymbol{\mu}}_{CU}$ will require its covariance $\boldsymbol{\Sigma}_1$ to be at least as large as the squared error $\hat{\boldsymbol{\mu}}_{CU} - \boldsymbol{\mu}_1$ (Liggins II et al., 2008). The same reasoning applies if the estimate $\boldsymbol{\mu}_2$ is consistent.

Then, among the family of the Gaussians in the space defined by Equation 2.3, the one that minimises some measurement of the size of $\hat{\boldsymbol{\Sigma}}_{CU}$, e.g., $|\hat{\boldsymbol{\Sigma}}|$, or the ratio $|\hat{\boldsymbol{\Sigma}}|/\hat{\boldsymbol{\mu}}$, is usually chosen. Specifically, the CU method performs hypothesis merging through increasing the variance (in the univariate case) of the fusing estimate to include all the possible hypotheses. By doing so, the CU estimate has the property of always being consistent with all the possible hypotheses, as opposed to the CI estimate that is potentially inconsistent with some of them. In more detail, Figure 2.2 shows an example of two Gaussian estimates, A and B, fused through CI and CU. In this example, it can be seen that, in contrast to CU, 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 the CU estimate is not very informative due to its high level of uncertainty. Thus, it does not satisfy part of our Req. 2. Given this, as we seek fusion methods with a good trade–off between accuracy and low predictive uncertainty (Req. 2), the CU method will only be considered as a conservative fusion benchmark to be compared against our approach.

2.4 Trust-Based Fusion Approaches for Crowdsourced Data

In contrast to the non-trust based fusion methods discussed so far, we now consider alternative approaches that involve modelling a user's trustworthiness in the data fusion

processes. Similarly to the previous discussion, our review covers the two classes of trust-based fusion algorithms for discrete data (Section 2.4.1) and continuous data (Section 2.4.2).

2.4.1 Dealing with Discrete Data

Methods for fusing discrete data have emerged in several crowdsourcing domains that involve classification of tasks (see the examples cited in Section 2.3.1). In general, a standard approach to this problem is to use model–based machine learning to derive the correct classification answer and some additional information about the task and the users from crowdsourced dataets through statistical inference models. In particular, such a machine leaning approach consists of designing a statistical model of a crowd reporting process and then applying inference to estimate unobserved quantities based on the data gathered from the crowd (Dawid and Skene, 1979; Whitehill et al., 2009; Bachrach et al., 2012b). As these models are relevant to our requirements of fusing crowdsourced data in unsupervised settings (Req. 1) combined with learning the user's trustworthiness (Req. 2), we review them in detail in the following sub–sections.

2.4.1.1 Classifier Combination

Classifier combination is one of the first fusion approaches that emerged from the crowdsourcing literature concerned with in combining discrete crowd responses. In particular, the first classifier combination algorithm designed to intelligently combine the classification labels from the reports of different users (i.e., the classifiers) was proposed by Dawid and Skene in 1979 (well before the advent of crowdsourcing) to study the advantage of using low-cost noisy classification data produced by untrained users for unsupervised learning algorithms. In their model, each user k has an individual probability vector of reporting the true label for each class j denoted as $\boldsymbol{\pi}_j^{(k)} = \{\pi_{j1}^{(k)}, \ldots, \pi_{jJ}^{(k)}\}$, where J is the maximum class. The set of these probability vectors, one for each class, gives a $J \times J$ confusion matrix that defines the accuracy profile of the individual user. To infer a particular user's confusion matrix from a given set of user's labels $\boldsymbol{C}$ containing I labels, they define p_j as the probability of any image in the set belonging to class j, and take the data likelihood for a number of independent and identically distributed reported labels $c_i^{(k)} \in \boldsymbol{C}$ as:

$$p(\boldsymbol{C}, \tau_i | \boldsymbol{\pi}_j^{(k)}, p_j) = \prod_{i=1}^{I} p_{\tau_i} \left\{ \prod_{k=1}^{K} \pi^{(k)}_{\tau_i c_i^{(k)}} \right\}$$

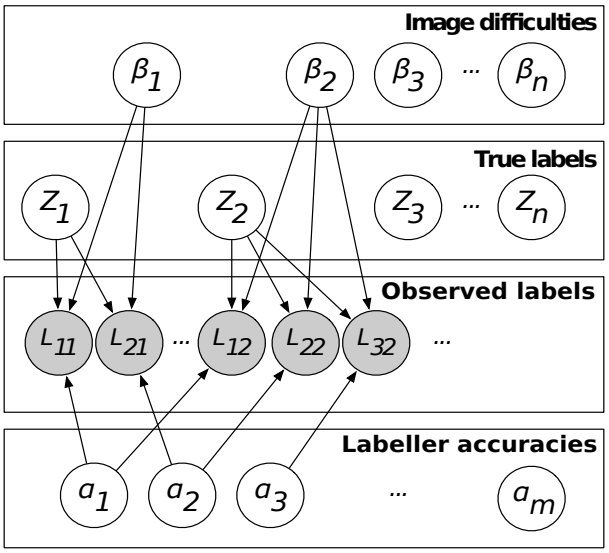
where τ_i is the (unobserved) true class of image i. Then, using expectation-maximisation (EM), i.e. a standard iterative method for approximating inference in statistical models (Dempster et al., 1977), they are able to estimate the p_j and $\pi_{ij}^{(k)}$ parameters for each

$i, j = 1, \ldots J$. In more detail, their EM-based inference is based on having an expectation step in which the correct answer is estimated from the data based on the current user's confusion matrix parameters, and then a maximisation step that updates the confusion matrix parameters to their maximum likelihood values under the updated item's classes. Ultimately, other algorithms were inspired by Dawid and Skene's method (DS) and applied to several crowdsourcing problems such as image labelling, galaxy classification and annotation tasks (Ipeirotis et al., 2010; Snow et al., 2008; Simpson et al., 2012). In particular, Kim and Ghahramani (2012) introduced the Bayesian version of DS, i.e., Bayesian classifier combination (BCC). In more detail, BCC defines prior probabilities over the random variables and computes approximate posterior estimates of the user's confusion matrix and the item's true class. In these terms, BCC is particularly useful for incorporating some prior knowledge about the user's trustworthiness and the item's classification in the fusion process. For example, we might know that some users are more reliable than others and so chose the appropriate priors in order to achieve a more accurate fusion; as was shown by Raykar et al. (2010) in a crowdsourcing application of classifying cancer diagnoses in the medical domain. More generally, DS, BCC and other models of this kind are 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 performing trust-based inference over crowdsourced data (Req. 1), we review them in the next section.

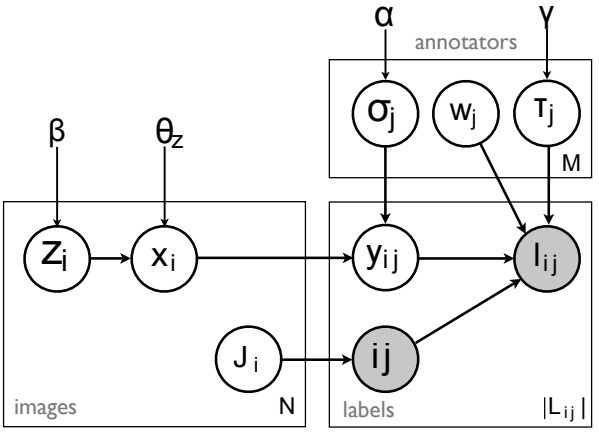
2.4.1.2 Graphical Models

In machine learning, graphical models are tools for representing probability density functions in the form of a graph that highlights its factorisation properties (Koller and Friedman, 2009). Specifically, a graphical model is a directed graph with a set of random variable represented as nodes, distinguishing between observed nodes (shaded) and unobserved or latent nodes (unshaded). The directed links represent the probabilistic dependencies between pairs of nodes. In these models, inference flows through the graph based on the conditional dependencies defined by the links. In such a way, it is possible to estimate the probability densities of the latent (unobserved) nodes based on the data feeding the observed nodes.

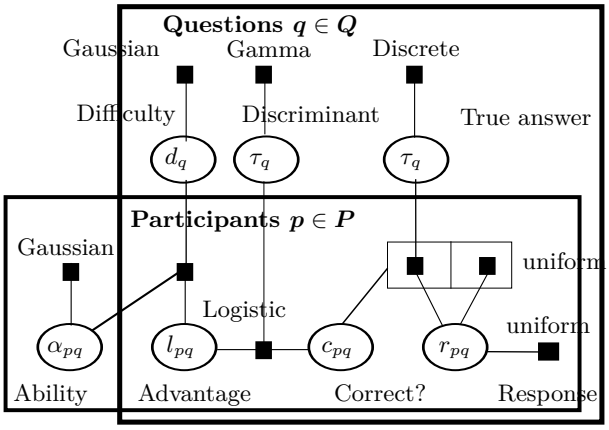
A number of trust-based data models for the joint estimation of the user's accuracy and the item's true value with discrete crowd responses based on graphical models have recently been presented. Some of these are shown in Figure 2.3. In detail, Whitehill et al. (2009) model an image labelling task considering 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_{kj} reported by user k for the image j depends on the true binary label Z_j, the image difficulty β_j and the expertise of the user α_i (Figure 2.3a). Then,



(a) Whitehill et al. (2009)



(b) Welinder et al. (2010)



(c) Bachrach et al. (2012b)

Figure 2.3: Examples of probabilistic graphical models proposed in previous work for crowdsourced image labelling (a and b) and IQ testing (c)

putting Gaussian priors on α and β, they compute maximum-a-posteriori inference of the posterior parameters. Furthermore, Welinder et al. (2010) extend this model to consider the accuracy of the user in a multidimensional space, with variables representing the competence, the expertise and the bias of user, i.e. the α, w and τ parameters respectively (Figure 2.3b). Finally, Bachrach et al. (2012b) 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 2.3c). Importantly, all these algorithms were shown to empirically outperform majority voting which promotes trust-based fusion approaches as a valid solution to crowdsourcing problems.

In general, machine learning algorithms based on graphical models are the first concrete solution for the trust-based data fusion problem with crowdsourced information. However, the main issue of this approach is that these models are normally designed for a particular problem they are trying to solve (Ghahramani, 2004). Unfortunately, the design of such models does not trivially extend to other types of crowdsourcing problems such as, for example, our problem of modelling user's trustworthiness in fusion processes with crowdsourced estimates. Additionally, another issue is the complexity of performing inference on such models. In fact, while these models can be arbitrarily complicated by adding new random variables (modes) to the graph, inference can rapidly become analytically intractable. That is, it is impossible to derive the exact Bayesian update expressions for the posterior distributions of these variables. This issue is partially alleviated by a number of techniques to approximate posterior inference on graphical models such as expectation-propagation (Minka, 2001), variational methods (Winn et al., 2005) and sampling methods (Gilks, 2005). However the problem of such approximation techniques is that they are prone to finding sub–optimal solutions in non-convex problems or to requiring many samples to achieve a good level of approximation.

Specifically for our problem, none of the discussed models considers observations as continuous estimates as we require (Req. 1). By contrast, they focus only on modelling single value observations (i.e. without reported precision). Therefore, using a similar probabilistic approach, in our work we need to take a step forward in designing new probabilistic models with computationally tractable inference for the case of fusing crowdsourced estimates of continuous quantities in Chapters 3 and 4.

2.4.2 Dealing with Continuous Data

Methods for fusing continuous data based on a trust model of a user are closely related to our set of problems (in particular to Reqs. 1 and 2). Along the line of our discussion in Section 2.3.2, sensor fusion is a promising research area where methods for fusing sensor estimates can be found. Interestingly, some of these sensor fusion methods consider the sensor's trustworthiness as part of the requirements of their data fusion processes.

However, we already discussed that, due to the key differences between sensor data and human–generated data, an open question is how these sensor fusion methods are able to maintain their efficiency when applied to crowdsourced information. Since we are not aware of any sensor fusion algorithm evaluated in a crowdsourcing context, we will have to review these trust-based sensor fusion approaches, particularly discussing their ability to solve the trust-based data fusion problem with crowdsourced information. In this respect, we discuss two main approaches related to outlier detection approach and other trust–based sensor fusion that can form a basis for a method to achieve this objective.

2.4.2.1 Outlier Detection

A possible way to identify untrustworthy estimates is based on outlier detection. As defined 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 concept fits the view of having untrustworthy estimates which significantly deviates from the crowd consensus. Notice that, however, Hawkins's definition only captures certain kinds of outliers, namely those points that outlie relative to the global dataset. For this reason, they are referred to as "global" outliers. In contrast, Breunig et al. (2000) provides a more general definition of a density-based outlier as "the points that outlie with respect to their neighbouring points". These are often referred to as "local" outliers. Since the Breunig et al.'s definition is more appropriate for complex data structures, we will adopt this in the discussion that follows.

Given this background, one fairly simple idea to solve our problem is to use outlier detection to identify and remove untrustworthy estimates (satisfying Req. 2). Subsequently, we can compute the fusion of the remain inlier estimates (satisfying our Req. 1) using any non-trust based fusion methods, e.g., CI. To describe this methodology, we refer to the standard density–based outlier detection method of the *local outlier factor* (LOF) (Breunig et al., 2000). Such a method is based on assigning a LOF score to each point as an indicator of its outlier level which is computed by measuring the relative density of the point compared to its neighbours. That is, the method seeks to identify the outliers by measuring whether the density around each estimate is significantly different from the density around its neighbours. The procedure for computing LOF scores is detailed in Algorithm 2.1. In more detail, the algorithm first computes the reachability distance of each report from its neighbour based on the parameter k that defines the locality region of r, i.e., the set of its k nearest neighbours (step 2). For this step, the distance between two probabilistic estimates can be measured using the *Kullback-Leibler divergence* (KL)(Kullback and Leibler, 1951). In particular, KL is a standard metric for measuring the distance between probability densities that, for the

Algorithm 2.1 Local Outlier Factor

Variables :

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

Algorithm *LOF(**R**, k, l)*

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

case of two multivariate Gaussian densities of dimension d, is:

$$KL(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)||\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)) = \frac{1}{2}\Big(tr(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) - \log\left(\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|}\right) - d\Big) \tag{2.4}$$

Next, the algorithm computes the local reachability distance $ldr(r)$ as the inverse of the mean reachability distance between r and its neighbours (Step 3). The LOF(r) is computed as the ratio of its local reachability of r and the one of its neighbours (step 4). Then, the algorithm returns the mean $\hat{\boldsymbol{\mu}}_{LOF}$ and the covariance matrix $\hat{\boldsymbol{\Sigma}}_{LOF}$ of the fused estimate obtained by merging the inliers' reports, i.e., the reports with LOF lower than the threshold l (Step 6). In this context, this means that the LOF score can be interpreted as the trustworthiness of each report.

Using this method, we can compute the trust-based fusion of a crowdsourced dataset by using LOF to filter the outliers that are likely to represent untrustworthy estimates. However, in order to apply such a method, the threshold l must be appropriately chosen. Notice that setting l to the right value is important to make the algorithm not too selective (i.e., l is too small) or too permissive (i.e., l is too large) in selecting the outlier set. Unfortunately, it is not trivial to make such an optimal choice of l for each dataset.

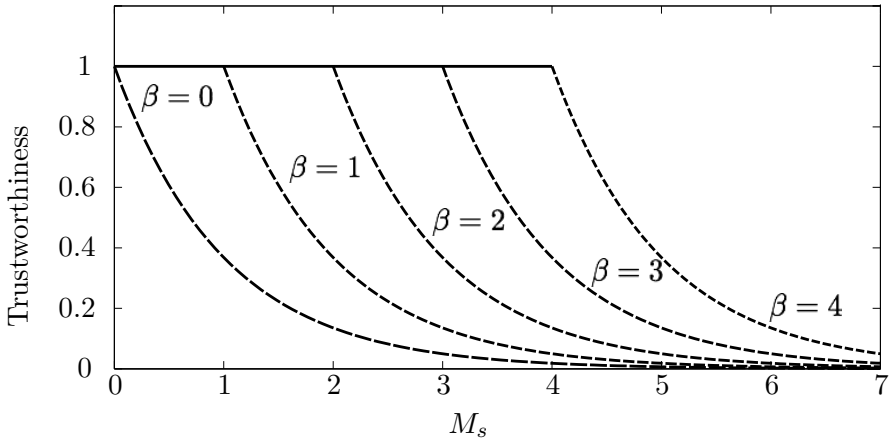


Figure 2.4: Plot of the RM trustworthiness function varying the β parameter.

Alternatively, we seek a more flexible parameter-free method that can learn the report's trustworthiness without relying on any outlier thresholds. Therefore, we will consider LOF as a benchmark in the evaluation of our approach.

2.4.2.2 Trust-Based Sensor Fusion

In the sphere of trust-based sensor fusion, there are methods that deal with the uncertainty in sensor fusion through sensor trust models (Reece et al., 2009; Momani et al., 2010; Guan et al., 2009). In particular, Reece et al. presented a algorithm (RM) for sensor noise recovery which handles unknown sensor's fault types by modelling the sensor's trustworthiness. In more detail, RM is based on recovering the readings from the sensor's noise in two stages. In the first stage, each sensor uses a pre-defined set of fault models to identify some known fault types in the reading[1]. In the second stage, the algorithm computes the trustworthiness of each sensor based on a consensus rule, which is close to the idea of the consensus–based trust model introduced in Section 2.4. In more detail, RM estimates the sensor's trustworthiness based on the distance between their estimates from the consensus estimate. Thus, RM contributes to our work in two ways. Firstly, it provides a consensus rule for fusing estimates reported by untrustworthy sensors. Secondly, it describes a distance–based method for estimating the sensor's trustworthiness from its reported readings. These two steps of RM are described in what follows[2].

Trust-Based Consensus Rule for Fusing Gaussian Estimates: Given two univariate Gaussian estimates, $\boldsymbol{e}_1 = \langle \mu_1, \theta_1 \rangle$ and $\boldsymbol{e}_2 = \langle \mu_2, \theta_2 \rangle$, where μ and θ are the mean and precision of the Gaussian distribution respectively, and given $t_1, t_2 \in [0, 1]$ as the

[1]Specifically, they consider drift, spike, shock and echo faults.
[2]For simplicity, our description of RM assumes an univariate case.

trustworthiness of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ respectively, then the consensus estimate between e_1 and e_2 is the Gaussian estimate $\boldsymbol{e}_{cons} = \langle \mu_{cons}, \theta_{cons} \rangle$, with trustworthiness t_{cons} given by:

$$\tilde{\mu} = (\theta_1 \mu_1 + \theta_2 \mu_2)/(\theta_1 + \theta_2) \tag{2.5}$$

$$\mu_{cons} = t_1 t_1 \tilde{\mu} + t_1(1 - t_2)\mu_i + t_2(1 - t_1)\tilde{\mu} \tag{2.6}$$

$$\tilde{\theta} = \theta_1 + \theta_2 \tag{2.7}$$

$$\theta_{cons} = t_1 t_2(\tilde{\theta} - (\tilde{\mu} - \mu_{cons})^2) + t_1(1 - t_2)(\theta_1 - (\mu_1 + \mu_{cons}^2)) + t_2(1 - t_1)(\theta_2 - (\mu_2 + \mu_{cons}^2)) \tag{2.8}$$

$$t_{cons} = t_1 + t_2 - t_1 t_2 \tag{2.9}$$

Fisher ratio–Based Trust Evaluation: Given the consensus estimate $\boldsymbol{e}_{cons} = \langle \mu_{cons}, \theta_{cons} \rangle$, the trustworthiness of a univariate Gaussian estimate $\boldsymbol{e}_k = \langle \mu_k, \theta_k \rangle$ is defined as follows. Let F_s be the *Fisher ratio* between $\boldsymbol{e}_{cons}$ and $\boldsymbol{e}_k$ given by[3]:

$$F_s = \frac{(\mu_{cons} - \mu_i)^2}{(\sigma_i + \sigma_{cons})} \tag{2.10}$$

Then, the trustworthiness t_k for the estimate $\boldsymbol{e}_k$ is:

$$t_k = \begin{cases} 1 & \text{if } F_s < \beta \\ \exp(-(F_s - \beta)) & \text{otherwise} \end{cases} \tag{2.11}$$

where β is the parameter which sets the point after which the sensor's trustworthiness decreases exponentially. In more detail, Figure 2.4 shows the plot of the trust evaluation function used by RM (Equation 2.11) for different β values. In particular, by iterating between these two steps, RM is able to estimate the fused estimate together with the sensor's trustworthiness, after appropriately tuning the β parameter. This iterative procedure of RM are also described in Algorithm 2.2.

Against this background, RM can potentially be a solution for a crowdsourced sensor setting which would meet our requirements. However, the parametric trust model used by RM, which is natively designed for sensor fusion, might not be able to handle the uncertainty of human reporters as efficiently as in the sensor setting due to the its static trust evaluation function. Therefore, while pursuing our goal of defining a novel trust-based data fusion framework more suitable for crowdsourced human sensors, we will compare our approach to RM. In doing so, we contribute to the evaluation of this trust-based fusion algorithm in a crowdsourcing scenarios.

[3]In the original paper by Reece et al. (2009), the expression of Equation 2.10 was erroneously called the Mahalanobis distance.

Algorithm 2.2 Reece Method

Variables :

$\boldsymbol{R}$: report set.
acc : accuracy bound.
$epochs$: number of training epochs.

Algorithm *ReeceMethod(**R**)*

1: Start with uniform max trust values on all the reports:
$$\boldsymbol{t}^{(0)} =< 1, \ldots, 1 >$$
2: **while** ($|\boldsymbol{t}^{(s-1)} - \boldsymbol{t}^{(s)}| < acc$ **or** $s > epochs$) **do**
3: Fuse the observations using the consensus rule based on $t^{(s-1)}$:
$$\boldsymbol{e}_{cons}^{(s)} = consensus(\boldsymbol{R}, t^{(s-1)})$$
4: Update trustworthiness parameters based on $f^{(s)}$:
for $k = 1 : K$ **do**
$$t_k^{(s)} = F_s(\boldsymbol{R}, f^{(s)})$$
end for
5: **end while**
6: **return** $(\boldsymbol{t}^{(s)}, \boldsymbol{e}_{cons}^{(s)})$

2.5 Crowdsourcing Spatial Data

So far, our discussion has covered the state–of–the–art approaches that relate to our first set of requirements of fusing crowdsourced estimates of stationary items (Req. 1) when there is uncertainty about the user's trustworthiness (Req. 2). Now, in order to address our second set of requirements that relate to the fusion of crowdsourced estimates of non-stationary items (Req. 3 and 4), we extend our discussion to work related to crowdsourcing settings in which the reported values are correlated to a number of input features. In particular, we focus on work related to crowdsourcing spatial functions which is an application emerging across several domains, including disaster response, disease mapping, weather forecasting and radiation monitoring (Heinzelman and Waters, 2010; Quinn et al., 2011; Overeem et al., 2013; Venanzi et al., 2013a). In these applications, untrustworthy crowd reports are problematic in the sense that their data can lead to incorrect spatial predictions in the same modalities that we discussed for the stationary case (Section 2.4). However, the spatial correlations in the dataset make the fusion problem more challenging, particularly in terms of learning the fused output as a continuous spatial function. To deal with this case, we now look at the class of learning models for spatial regression.

2.5.1 Spatial Regression Approaches

In the class of data regression approaches, a number of spatial regression models are available, ranging from linear and polynomial regression to neural networks, latent force

models and kernel methods. In more detail, linear regression models are simple and computationally efficient methods to infer a spatial phenomenon through the linear relationship between an environmental variable of interest (such as temperature or nuclear radioactivity) and some explanatory variables (such as location and time) (Stranders, 2010). However, more complex spatial phenomena typically follow (strongly) non–linear distributions that are difficult to express through linear regression. Alternatively, neural networks, latent force models and kernel methods are more powerful models that can deal with non–linear phenomena more efficiently. In particular, these models offer a hybrid approach to incorporate the physical laws of a spatial phenomenon in data–driven processes (Alvarez et al., 2009). Their attractiveness steams from combining noisy observations of a spatial phenomenon with a physical model of the system. However, the design of such models require a deep understanding of the system dynamics and its physical laws. Furthermore, their inference becomes easily intractable due to the complexity of deriving close form predictive distributions under such statistical models. Generally speaking, there is typically a trade–off between the expressiveness of these regression models and their inference tractability (Bishop, 2006). In this space, we identify the Gaussian process from the family of the kernel methods as a rare exception of a regression model that is analytically tractable and at the same time very flexible (Rasmussen and Williams, 2006). In particular, Gaussian process regression have been widely used in Geostatistics to model various spatial phenomena such as ultra fine particle concentrations, sound levels, and weather–related events (Li et al., 2014). In fact, their advantage is that they do not require any prior knowledge of the dynamics of the phenomenon that are instead inferred from the data through a principled non–parametric framework. Since the Gaussian process appears as a suitable basis for a solution to our problem, we review it in detail in the next section.

2.5.2 Gaussian Process Spatial Regression

The Gaussian process (GP) is a Bayesian non–parametric model widely used for spatial regression in many real–world applications (Rasmussen and Williams, 2006). This model is able to deal with spatial regression problems which lie in the following setting. Given a dataset of n geo–located observations of an unknown spatial function $f(\boldsymbol{x})$, where an observation normally consists of a pair with a location (latitude and longitude) and an observed value, i.e. $\mathcal{D} = \{\boldsymbol{x}_o \in \mathbb{R}^2, y_o \in \mathbb{R} : o = 1, \ldots, N\}$, where N is the total number of observations, and the objective is to determine $f(\boldsymbol{x})$ from this dataset. The GP approach to this problem is based on assuming that the joint distribution of any subset of a function's outputs is a Gaussian density and that y_o is a noisy measurement, with zero–mean Gaussian noise, of the actual function value, $\tilde{y}_o$, at the location $\boldsymbol{x}_o$. Formally:

$$y_o = \tilde{y}_o + \epsilon, \qquad \tilde{y}_o = f(\boldsymbol{x}_o), \qquad \epsilon \sim \mathcal{N}(0, \sigma_N^2) \tag{2.12}$$

In this setting, the GP is able to perform Bayesian inference in the function space by defining a prior distribution over f specified by a *mean function* $m(\boldsymbol{x}) = E[f(\boldsymbol{x})]$ and a *covariance function*, or kernel $K(\boldsymbol{x}, \boldsymbol{x}') = cov(\boldsymbol{x}, \boldsymbol{x}')$. That is:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x}')) \tag{2.13}$$

where

$$\begin{aligned} m(\boldsymbol{x}) &= E[f(\boldsymbol{x})] \\ K(\boldsymbol{x}, \boldsymbol{x}') &= Cov(f(\boldsymbol{x}), f(\boldsymbol{x}')) \end{aligned}$$

Specifically, the mean function represents the default value of f in the regions where no observations are available (and it is often conventionally taken to be zero for simplicity). The covariance function models the correlation between the input points. In particular, this covariance function may have some free parameters, called *hyperparameters*, to control the smoothness and noise properties of the covariance function in modelling correlations. For this reason, such a covariance function is a key element in a GP predictor that needs to be appropriately chosen for a specific dataset. In particular, a standard set of covariance functions for GPs is represented by the class of stationary functions, where stationarity means the function's value depends on the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$. Among these, we describe the squared–exponential covariance function that is a standard kernel for modelling smoothly changing quantities:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f \exp\Big(-\frac{1}{2l^2}(\boldsymbol{x} - \boldsymbol{x}')^2 \Big) \tag{2.14}$$

This function has two hyperparameters that are the signal variance σ_f and the length scale l, respectively. The set of these hyperparameters is denoted as γ. Given this, if we wish to predict the value of f at a new test location $\boldsymbol{x}_*$, and let such a value be y_*, assuming that y and y_* are Gaussian random vectors, we can write the joint distribution at the test location as:

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N}\left(m(\boldsymbol{x}), \begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_N^2 I_N & K(\boldsymbol{x}, \boldsymbol{x}_*) \\ K(\boldsymbol{x}_*, \boldsymbol{x}) & K(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{bmatrix} \right) \tag{2.15}$$

where I_N is the $q \times q$ identity matrix. Then, 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:

$$p(y_*|\boldsymbol{x}, y, \boldsymbol{x}_*) = \mathcal{N}(E[y_*], \sigma^2(y_*)) \tag{2.16}$$

where

$$E[y_*] = m(\boldsymbol{x}) + K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_N^2 I_N]^{-1} y \tag{2.17}$$

$$\sigma^2(y_*) = K(\boldsymbol{x}_*, \boldsymbol{x}_*) - K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_N^2 I_N]^{-1} K(\boldsymbol{x}, \boldsymbol{x}_*) \tag{2.18}$$

Specifically, Equations 2.17 and 2.18 denote the mean and the variance of the Gaussian predictive distribution of $f(\boldsymbol{x}_*)$. In particular, the predictive variance is useful for estimating the predictive uncertainty, which is useful for us to meet our Req. 2. Also, by integrating the likelihood, $p(y|f, \boldsymbol{x}) = \mathcal{N}(y|f, \sigma_N)$, over the GP prior, $p(f) = \mathcal{N}(f|0, K(\boldsymbol{x}, \boldsymbol{x}'))$, we can obtain the closed form equation of the *marginal likelihood*, i.e. the data likelihood marginalised over the latent function:

$$\begin{aligned} \log p(y|\boldsymbol{x}) = &-\frac{1}{2} y^T [K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_N^2 I_N]^{-1} y \\ &-\frac{1}{2} \ln |K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_N^2 I_N| - \frac{N}{2} \ln(2\pi) \end{aligned} \tag{2.19}$$

Notice, that we use the logarithm of marginal likelihood which simplifies the products as sums while still preserving the monotonicity of the function. In particular, the log marginal likelihood is useful for model training (Rasmussen and Williams, 2006). That is, by maximising this function, we can find the appropriate values of the hyperparameters.

Importantly, the GP predictive distribution described above is derived under the assumption of a process noise ϵ having a constant variance σ_N. This means that all the estimates reported by the users have the same level of noise, which in statistics is referred to as *homoskedastic* regression (Silverman, 1985). In a crowdsourcing setting, this assumption does not allow us to deal with situations in which users have varying trustworthiness, as we require (Req. 2), using a standard GP. In particular, by having users with different levels of trustworthiness, the reported data will necessarily have different noise variances which is a problematic case for a homoskedastic GP. To address this shortcoming, we now introduce the *heteroskedastic* variant of the Gaussian process which is based on modelling non-constant noise levels on the input set[4] .

2.5.3 Heteroskedastic Gaussian Processes

In heteroskedastic regression, we deal with a situation in which the signal noise varies across the inputs. Formally:

$$y_o = f(\boldsymbol{x}_o) + \epsilon_o \tag{2.20}$$

[4]Heteroskedastic models are also referred to as heteroscedastic in the literature. In this thesis, we choose the latter spelling that is conventionally used in econometrics papers.

In particular, if we assume that the noise terms are independent and normally distributed, we obtain the same model studied by Goldberg et al. (1997). That is:

$$\epsilon_o \sim \mathcal{N}(0, \sigma^2(\boldsymbol{x}_o)) \tag{2.21}$$

and defining $\Sigma_x = diag\{\sigma^2(\boldsymbol{x}_o)\}$, the predictive Gaussian distribution of the model is:

$$E[y_*] = K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x]^{-1} y \tag{2.22}$$

$$\sigma^2(y_*) = K(\boldsymbol{x}_*, \boldsymbol{x}_*) - K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x]^{-1} K(\boldsymbol{x}, \boldsymbol{x}_*) \tag{2.23}$$

In general, the heteroskedastic Gaussian process (HGP) model poses an inference challenge related to the noise function σ^2 that defines input-specific noise rates that are typically unknown and so the values must be estimated in some way. To deal with this problem, various works advocate a hierarchical modelling approach that makes use of a stack of GPs to model σ^2 (Goldberg et al., 1997; Kersting et al., 2007). Since this hierarchical GP makes the inference no longer tractable analytically as in the case of the single GP, research has tended to focus on approximate inference for 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 the analysis of this model. Firstly, in our setting, the noise rates are known since they relate to the reported precision of the user's observations (Req. 2). Secondly, in crowdsourcing settings, it is reasonable to assume that such noise rates are reported independently by the users. Given these assumptions, we are able to separate σ by its individual noise terms, i.e. $\sigma(\boldsymbol{x}_i) = \sigma_i$ and derive the predictive Gaussian distribution as described by Equations 2.22 and 2.23. However, the limitation of the current HGP model is that it does not consider the individual trustworthiness of the users who report the observations, thus presenting the same issue of equally trusted reports discussed for CI in Section 2.3.2. In the attempt to address this limitation, a GP model designed to deal with multiple noisy estimators was presented by Groot et al. (2011) . To handle the individual noise of each estimator, they use a standard GP with a rational quadratic covariance function:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f \exp\Big(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^T A^{-1} (\boldsymbol{x}' - \boldsymbol{x})\Big) \tag{2.24}$$

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. Thus, their model can deal with the individual trustworthiness of the observations coming from different estimators by setting appropriate values of their length scales. However, a potential problem of this model applied to our setting is that the length scale parameters grow with the number of inputs, which yields the intractability problem of training this model over large datasets. Therefore, in Chapter 5, we detail an alternative HGP model that learns

trust parameters at the user level, thus still considering individual reliabilities of the reports with a smaller and more tractable parameter set. In addition, our model will also address the requirement of considering the precisions of the reported observations (Req. 3) which is currently not considered in the discussed models.

2.6 Crowdsourcing Spatial Intensity Functions

To address our last set of requirements related to crowdsourcing non-stationary quantities based on reported point estimates, we now discuss work that relates to the inference of spatial intensity functions. In particular, we discuss the state–of–the–art statistical approaches to analyse spatial patterns that are expressed as continuous spatial intensity functions across location data. These approaches define the theories of the models employed in real-world applications concerned with point–based spatial data. For example, using one of these approaches, Diggle et al. (2005)'s work analyses the spatial incidence of disease reports from people's calls to the UK national health system reporting service (NHS direct) for the purpose of detecting unexpected variations in gastroenteric symptoms. In general, since these approaches emerging from this and other domains are closely related to our requirements of computing fused outputs from human–reported spatial point estimates (Req. 4), we review them in detail with respect to our crowdsourcing context.

2.6.1 Spatial Point Processes

The main challenge in dealing with spatial point reports is inferring spatial patterns in the possibly noisy point estimates reported by the crowd. These patterns are required to compute a fused output that predicts the distribution or points over a monitored region. Spatial point processes are commonly used to accomplish this objective in applications that range from monitoring seismic events, plant ecology, astronomic events, etc. To date, the most solid basis of a spatial process model to analyse non–stationary random points is represented by non–homogeneous Poisson processes (Cressie and Wikle, 2011) that we discuss in the next sub–section.

2.6.2 Non–Homogeneous Poisson Processes

We start by introducing the key mathematical concepts of a non–homogeneous Poisson process (NHPP) for analysing spatial point patterns. Specifically, we are given a set of point estimates (or location estimates) and we assume that the spatial distribution of such points is a random process described by a spatial intensity function $\lambda(\boldsymbol{x})$, where $\boldsymbol{x}$ is the variable representing the space of locations. In general, $E[\lambda(X)]$, i.e. the

expected number of reports within a certain region X, is considered to be realisation of a stochastic process represented by a NHPP (Cressie and Wikle, 2011):

$$p(E[\lambda(X)]) = \text{Poisson}\Big(\int_X \lambda(X)\Big)$$

That is, a NHPP assumes that the number of points in X is Poisson distributed with the intensity of such a Poisson distribution given by the integral of the λ function in that region. In particular, the non-homogeneity of the process relates to the fact λ is a non-constant function. Since λ is itself an unknown function, we assume that λ is generated from an independent stochastic process. This whole doubly stochastic NHPP is referred to as Cox process (Cox and Isham, 1980).

Based on various ways of modelling the stochastic process generating λ using parametric or non-parametric statistics, there are different types of Cox processes that can be defined. For our purposes, we focus on a standard case of Cox process named log-Gaussian Cox process (LGCP) which is closely related to Gaussian processes and, as such, it allows us to re-use the GP theory also for spatial point processes (Møller et al., 1998).

2.6.3 Log Gaussian Cox Processes

The LGCP is a special case of spatial Cox process where the the log of the intensity function $z(\boldsymbol{x}) = \log(\lambda(\boldsymbol{x}))$ is assumed to be generated by a Gaussian process. That is:

$$z(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x'})) \tag{2.25}$$

where

$$m(\boldsymbol{x}) = E[z(\boldsymbol{x})]$$
$$K(\boldsymbol{x}, \boldsymbol{x'}) = Cov(z(\boldsymbol{x}), z(\boldsymbol{x'}))$$

In particular, the GP prior allows us to model the latent intensity in a non-parametric framework with the same properties described for GP regression (Section 2.5.2). In particular, consistently with what we described for GP regression, we use a zero-mean function $m(\boldsymbol{x}) = 0$ and a covariance function $K(\boldsymbol{x}, \boldsymbol{x'})$ with hyperparameters $\theta = \{\sigma_s, l_s, \}$ modelling the mean structure and the covariance of the log-intensities between points, respectively. In particular, the use of zero-mean function means that the process assigns zero log-intensity in areas where no reports are available[5].

[5]While the use of zero-mean GP prior is counterintuitive as the model would predict log-zero, i.e. one report ($z = 0$) in areas where no reports has been observed, the whole model is generally easier to analyse and this side–effect is easy to filter in practice.

Thus, drawing the two processes together, the data likelihood for a set of N reported point $\boldsymbol{R} = \{\boldsymbol{x}_i : i = 1, \dots N\}$ is given by:

$$p(\boldsymbol{R}|z) = \exp\Big(- \int_X \exp z\Big) \prod_{i=1}^{n} \exp(z(\boldsymbol{x}_i)) \tag{2.26}$$

where $\boldsymbol{x}_i$ is the reported location and θ_i is its precision. Unfortunately, we cannot get a closed-form posterior inference in the LGCP due to the non-conjugate form of the Gaussian prior over z with the Poisson likelihood and also due to the stochastic integral over z involved in Equation 2.26. Therefore, a first standard approximation is made by discretising the space into a grid of resolution Δ_x, i.e. $\boldsymbol{X} = \cup \boldsymbol{X}_i : \boldsymbol{X}_i = \boldsymbol{X}_{i-1} + \Delta_x$. In particular, let X_i be a single cell of the partitioned space region. If we assume that the intensity z_i of points in $\boldsymbol{X}_i$ is approximately constant in $\boldsymbol{X}_i$ and independent from z_j with $i \neq j$ (i.e. the Poisson process is homogeneous in $\boldsymbol{X}_i$), the likelihood can now be written as:

$$p(\boldsymbol{R}|\boldsymbol{z}) \approx \prod_{i=1}^{n} \text{Poisson}\Big(\phi(X_i)|\exp(z_i)\Big) \tag{2.27}$$

with $\boldsymbol{z} = \{z_1, \dots, z_i\}$ and $\phi(X_i)$ is the point count of X_i, i.e. number of points falling in X_i.

Another intractability problem comes from computing the posterior distribution over $\boldsymbol{z}$ which requires the integration of the Gaussian prior (Equation 2.25) over the approximate Poisson likelihood (Equation 2.27). To get around this problem, there are Monte-Carlo sampling methods that can be used to approximate such a posterior density, although these methods can be computationally expensive depending on the number of samples requires to achieve a good approximation. To improve the efficiency of computing posterior updates, we consider the Laplace approximation (Friston et al., 2007) which is faster, although potentially less accurate, and approximates the posterior with a normal distribution obtained by taking the second-order Taylor expansion of the likelihood around its mode $\hat{z}$. That is:

$$p(\boldsymbol{z}|R) \approx \mathcal{N}(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\Sigma}}^{-1}) \tag{2.28}$$

where

$$\hat{\boldsymbol{z}} = \arg\max_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{R}) \tag{2.29}$$

$$\hat{\boldsymbol{\Sigma}} = - \bigtriangledown \bigtriangledown \log p(\boldsymbol{z}|\boldsymbol{R})|_{\boldsymbol{z}=\hat{\boldsymbol{z}}} \tag{2.30}$$

are the maximum a posteriori (MAP) estimates of the parameters obtained by the non-Gaussian posterior (see Rasmussen and Williams (2006), Section 3.4 for more details). Under such an approximation, we can compute predictive distribution of the

log-intensities over the entire spatial grid by integrating the Poisson likelihood (Equation 2.27) of the test points $z_* = z(\boldsymbol{x}_*)$ over the posterior distribution of $\boldsymbol{z}$ (Equation 2.28). This density is approximately multivariate normal with p.d.f.:

$$p(z^*|R,\theta) \approx \mathcal{N}(E[z_*], \sigma^2(z_*)) \tag{2.31}$$

where

$$E[z_*] = K(\boldsymbol{x}_*, \boldsymbol{x})K(\boldsymbol{x}, \boldsymbol{x})^{-1}\hat{\boldsymbol{z}} \tag{2.32}$$

$$\sigma^2(z_*) = K(\boldsymbol{x}_*, \boldsymbol{x}_*)(K(\boldsymbol{x}, \boldsymbol{x}) + \hat{\boldsymbol{\Sigma}}^{-\mathbf{1}})^{-1}K(\boldsymbol{x}_*, \boldsymbol{x}) \tag{2.33}$$

Finally, we can also derive the approximate marginal likelihood that is useful for model training as:

$$\begin{aligned}\log p(\boldsymbol{z}|\boldsymbol{R}) = &-\frac{1}{2}\hat{\boldsymbol{z}}^T(K(\boldsymbol{x}, \boldsymbol{x}))^{-1}\hat{\boldsymbol{x}} + \log p(\boldsymbol{R}|\hat{\boldsymbol{z}}) \\ &-\frac{1}{2}|I_N + \hat{\boldsymbol{\Sigma}}^{\frac{1}{2}}K(\boldsymbol{x}, \boldsymbol{x})\hat{\boldsymbol{\Sigma}}^{\frac{1}{2}}|\end{aligned} \tag{2.34}$$

where I_q is a $q \times q$ identity matrix. By maximising this expression, we can estimate the hyperparameters of the model in the same way described for the standard GP.

With this, we described the key equations for estimating the spatial intensity field of a set of point reports using the LGCP. However, similarly to the what discussed for the HGP, also this models has the shortcoming of not considering the trustworthiness of each report which is a key requirement of our problem (Req. 2). Therefore, in Chapter 6, we will show how to make the LGCP more robust against the presence of untrustworthy reports and so introduce our new trust-based LGCP.

2.7 Summary

In this chapter we introduced the key notions within the literature for dealing with fusing untrustworthy information in crowdsourcing applications. Specifically, we began by discussing various approaches to trust evaluation which are currently employed within crowdsourcing systems. Discussing the properties of these approaches with respect to our problem, we discarded the reputational approach and the gold-data driven approach as the former is not reliable in open crowd systems and the later is based on gold standards which are not part of our requirements. The consensus–based trust approach was identified as a suitable basis for a solution to work in our setting due to its property of estimating trust based on the consensus of the crowd which does not use gold standards. However, we discussed that one important limitation approach of the consensus–based trust approach is the lack of a consensus method for continuous estimates which is what we need according to our Req. 1. To rectify this, in Chapter 3 we will define a

novel consensus method that is suitable for our continuous crowdsourcing setting (thus satisfying this requirement).

Then, we discussed various approaches for fusing crowdsourced information. In particular, we distinguished between fusion approaches for discrete data and for continuous data and, for each of these classes, we discussed the approaches that make use of trust–based versus non–trust based fusion models. For the class of non-trust based methods, in both the discrete and the continuous data case, we criticised the fact that they might be inefficient in managing crowdsourced information due to considering reports as equally trustworthy. For the class of trust-based fusion methods for discrete data, we highlighted the fact that these methods are typically more efficient than non-trust based that is a key finding emerging from various previous works. However, we discussed the problem of extending the existing models for discrete data to continuous data which involves a new model and inference design. Therefore, our work will contribute to define such new models for the trust–based fusion of crowdsourced continuous estimates.

In the class of non-trust based fusion methods for continuous data, we discussed two standard CI and CU methods for merging Gaussian estimates. In particular, CI was identified as suitable basis for fusing estimates by taking into account the reported precisions under the assumption of a Gaussian noise. However, CI needs to be extended to consider the individual trustworthiness of the user to meet our Req. 2. Instead, CU was identified as the conservative fusion method due to its property of merging estimates by increasing the variance of the fused estimate in order to achieve the consistency of the fused output with all the reported estimates. Therefore, CU will only be referred to as a benchmark for our algorithms presented in Chapter 3 and Chapter 4.

In the class of trust-based fusion methods for continuous data, we first discussed the outlier detection approach as a possible methodology to deal with untrustworthy estimates in crowdsourced datasets. However, one drawback of this method is the sensitivity to the choice of the the outlier threshold and the locality parameter. Therefore, while seeking an alternative parameter-free data fusion approach for crowdsourcing, we will consider outlier detection as a trust-based fusion benchmark. Subsequently, we described RM which is an algorithm specialised in fusing sensor estimates in presence of unknown sensor faults. This algorithm was identified as a solution that could potentially meet our requirements for the crowdsourcing stationary setting due to its ability to learn the sensor's trustworthiness and merge the sensor estimates accordingly. However, the fact that its underlying parametric model natively defined for sensor fusion settings, which requires less flexibility than a crowdsourcing setting, suggests that we can potentially improve it by using more flexible parameters–free trust approaches. As such, we will use RM as another benchmark in the evaluation of our approach.

For the requirement of fusing crowdsourced estimates of non-stationary items, particularly in the context of crowdsourcing spatial estimates, we reviewed techniques for heteroskedastic spatial regression. In particular, we introduced the heteroskedastic Gaussian process which was identified as a a rare exception of a powerful but also tractable regression model which is particularly suitable to satisfy our requirements. However, its main limitation was identified in the fact that it's constant-noise variance assumption is unsuitable to models datasets of varying trustworthiness within the inputs. To address this, we will study a new trust-based HGP model based on having individual user's trustworthiness associated with the data in Chapter 5.

For the requirement of crowdsourcing spatial intensity fields, which is a special case of crowdsourcing non-stationary information, we introduced the key mathematical background of spatial point processes that related to the fusion of crowdsourced point estimates. In particular, we discussed the Log Gaussian Cox (LGCP) process which is a spatial point process with similar properties to Gaussian processes. In the same vein, we highlighted the key shortcoming of the LGCP in solving our problem which is due to ignoring the fact that reports might be of different trustworthiness. To address this shortcoming, in Chapter 6, we will devise a new trust-based LGCP inspired by the same theory as our trust-based HGP.

Chapter 3

A Frequentist Trust Model for Fusing Crowdsourced Estimates of Stationary Continuous Quantities

As per the objectives outlined in Chapter 1, we wish to build a reliable information fusion framework for crowdsourcing and participatory sensing applications. In this chapter, we focus on the first challenge faced by the development of such a framework which is the fusion of crowdsourced continuous estimates of stationary quantities (Req. 1). To address this challenge, we rely on the trust–based fusion paradigm which entails reasoning about the trustworthiness of individuals to improve the statistical accuracy of the data fusion models (Req. 2).

As discussed in Chapter 2, existing approaches to merging continuous estimates produced by possibly unreliable sources are mostly inspired by the sensor network domain. In particular, existing sensor fusion methods can be trivially adapted to crowdsourcing scenarios by considering human users as traditional hard sensors. However, these approaches typically assume some specific noise structures within the data that are more characteristic of sensor devices than human sources. For example, it is common for these sensor models to represent the data noise as drifts, spikes and shocks of the reporting sensors (Reece and Roberts, 2010). Yet, in crowdsourcing, we know that it would be unrealistic to assume that a human sensor would produce inaccurate data in a sensor–like manner (i.e. as an effect of a spike or a shock), or that a sensor can deliberately misreport observations in a human–like manner (Hall and Jordan, 2010). In general, these and the other shortcomings that we discussed in Chapter 2 (Section 2.4) relate to the fact that current sensor fault models do not appropriately represent the range of

human errors. Therefore, we need to investigate new fusion models for crowdsourced information.

In Chapter 2 (Section 2.4.2), we also discussed the idea of combining sensor fusion techniques with simple parametric trust models based on outlier detection of distance–based trust functions. This can be regarded as the first intuitive trust–based approach to achieve more efficient information fusion models in crowdsourcing. However, the inconvenience of this approach is the need for their parametric trust models to find the optimal parameters for each dataset being analysed.

Against this background, we develop a new trust–based fusion method which addresses these shortcomings through the combination of a principled statistical model of a human reporter's trustworthiness for fusing crowdsourced estimates of stationary quantities. Specifically, our approach is based on constructing a likelihood model of the fusion process based on having a set of user trust parameters scaling the uncertainty of the reported estimates. Then, using an efficient frequentist learning approach based on maximum likelihood inference, we can learn the values of the trust parameters from the reports coming from the crowd. This learning step is performed through an approximate inference scheme implemented by the MaxTrust algorithm that we provide.

In more detail, our contributions are the following:

- We present the first trust–based crowdsourcing model for jointly fusing untrustworthy estimates of stationary continuous quantities and learning the trustworthiness of the users in participatory sensing applications.
- We derive an efficient inference algorithm (MaxTrust) for our model, which is able to statistically estimate the user's trustworthiness and the fused output.
- Using the OpenSignal (`www.opensignal.org`) dataset containing cell–tower detections collected from Android mobile phones, we show that our algorithm outperforms existing benchmark methods in both absolute accuracy, gaining up to 22%, and predictive uncertainty, gaining up to 21%. Furthermore, we also show through simulated experiments that MaxTrust can achieve comparable accuracy with 10% more untrustworthy users within the crowd.

The remainder of this chapter is structured as follows. In Section 3.1, we describe our model and its key components related to the user's trust model and the information fusion process. In Section 3.2, we provide the full details of our MaxTrust algorithm to perform inference in our model. In Section 3.3, we test our algorithm with both simulated and real–world experiments using the OpenSignal dataset in which we compare its performance to five state–of–the–art methods. Finally, Section 3.4 concludes the chapter with a summary of the results.

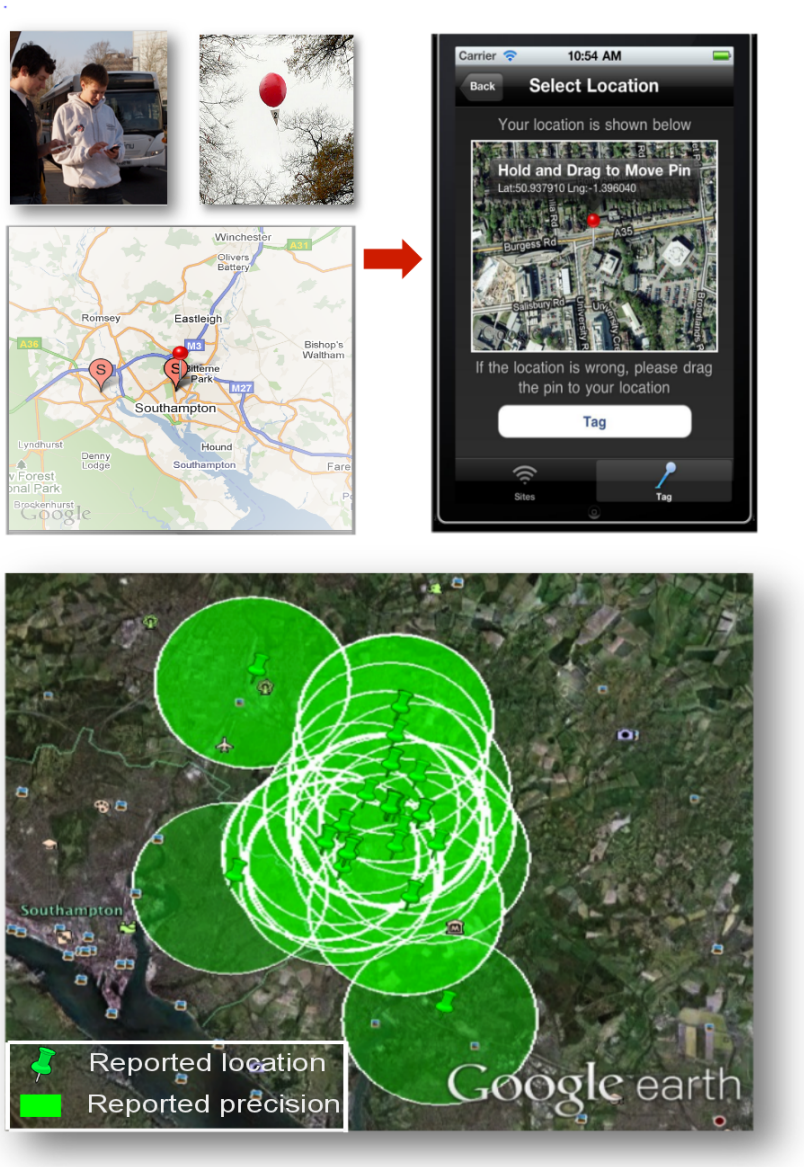


Figure 3.1: Illustration of scenario of crowdsourcing location data. The user reports GPS–based location estimates of a stationary target represented a red balloon (left most figure).

3.1 Model Description

In this section, we formally describe our user trust model (Section 3.1.1). Then, we detail our fusion method which embeds such a trust model (Section 3.1.2).

3.1.1 An Uncertainty Scaling User Trust Model

In this model, a crowd of K users reports observations of a stationary d-dimensional item with a true, unobserved value $\boldsymbol{\mu} \in \mathbb{R}^d$. Specifically, each user k reports p_k estimates of

$\boldsymbol{\mu}$ and each reported estimate $\boldsymbol{e}_{k,j}$ with $j = 1, \ldots, p_k$ comprises (i) the observation $\boldsymbol{x}_{k,j} \in \mathbb{R}^d$ and (ii) the precision of $\boldsymbol{x}_{k,j}$, i.e., $\theta_{k,j} \in \mathbb{R}_{>0}$. In particular, $\theta_{k,j}$ is the precision that user k reports about its observation, which is a key requirement of our crowdsourcing problem (Req. 1). Notice that, in practice, it is common for users to estimate $\theta_{k,j}$ as a self–apprised confidence about its observation, the precision of the measuring tool, or the variance of a series of repeated measurements. However, in situations where it is not possible to estimate the precision, this can also be set to a default value while still preserving the properties of our model. Therefore, our report set is $\boldsymbol{R} = \{\boldsymbol{e}_{k,j} | k = 1, \ldots, K; j = 1 \ldots p_k\}$ where each report $\boldsymbol{e}_{k,j} = \langle \boldsymbol{x}_{k,j}, \theta_{k,j} \rangle$ denotes that user k estimates $\boldsymbol{\mu}$ as $\boldsymbol{x}_{k,j}$ with precision $\theta_{k,j}$. In more detail, Figure 3.1 illustrates a typical scenario described by our model in which users report estimates (e.g., location estimates) of an observed stationary item (e.g. a cell tower position) collected through its own sensor device (e.g. the GPS of its phone). The set of reports is shown on a map as estimates (circles) of the item's value based on the reported observations and the precisions.

In our model, we assume we have a Gaussian distributed uncertainty around each user's observation associated with the reported precision. That is, for each report $\boldsymbol{e}_{k,j}$, the probability density function (p.d.f.) of its estimate is expressed by the following Gaussian density:

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{r_{k,j}}) &= \mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_{k,j}, \theta_{k,j}\mathrm{I}_d) \\ &= \left(\frac{t_k \theta_{k,j}}{2\pi}\right)^{d/2} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_{k,j})^T \theta_{k,j} \mathrm{I}_d (\boldsymbol{x} - \boldsymbol{x}_{k,j}) \right) \end{aligned} \tag{3.1}$$

where $\boldsymbol{x}$ is a generic point in $\mathbb{R}^d$ and $\theta_{k,j}\mathrm{I}_d$, with $\mathrm{I}_d = d \times d$, is the precision (or inverse covariance) matrix. In particular, by using a diagonal precision matrix, we assume an uncorrelated and equally distributed noise along the d dimensions of the input space.

Next, as part of our requirements (Req. 2), we consider the uncertainty of the individual user's trustworthiness. To this end, we formally state the conditions that define an untrustworthy estimate in our model. In detail, we assume that $\boldsymbol{e}_{k,j}$ is trustworthy w.r.t. $\boldsymbol{\mu}$ if the following condition holds:

$$\boldsymbol{x}_{k,j} \sim \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \theta_{k,j}), \qquad E[\boldsymbol{x}] = \boldsymbol{\mu}$$

That is, a report provided by a trustworthy user is assumed to be sampled from a Gaussian distribution $\mathcal{N}$, where its expected value is the item's true value and $\theta_{k,j}$ is the noise of the sampling process. This assumption can also be interpreted as the fact that $\boldsymbol{x}_{k,j}$ is generated as a noisy observation of $\boldsymbol{\mu}$ with noise correlated to $\theta_{k,j}$. By contrast, an untrustworthy report is assumed to be sampled from *any* other statistics that are not correlated to $\boldsymbol{\mu}$. For example, reports may have gain and offset errors, they may correctly calibrated but exhibit greater noise than what they report. In simple

settings, each of these errors can be modelled within the data aggregation process to be able of potentially recovering the most genuine estimates of the true item. However, in crowdsourcing settings, representing all these factors in a parametric model is not always feasible given the sparse data typically available. Thus, we deal with this set of known and unknown inaccuracies in the reports using a more general uncertainty scaling trust modelling approach.

More formally, given this, we define a set of *trust parameters*, $\boldsymbol{t} = (t_1, \ldots, t_k)^T$, to represent the reliability of each user in the range of $[0, 1]$. Specifically, t_k is the trustworthiness of user k with $t_k = 1$ meaning that the user is fully trustworthy and $t_k = 0$ meaning that it is completely untrustworthy. Then, we introduce the key feature of our trust model which is a new Gaussian p.d.f for a report $\boldsymbol{e}_{k,j}$ obtained by using t_k as the *scaling parameter* for $\theta_{k,j}$. Thus, Equation 3.1 is updated as follows:

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{e}_{k,j}, t_k) &= \mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_{k,j}, (t_k\theta_{k,j}\mathrm{I}_d)) \\ &= \left(\frac{t_k\theta_{k,j}}{2\pi}\right)^{d/2} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_{k,j})^T t_k\theta_{k,j}\mathrm{I}_d(\boldsymbol{x}-\boldsymbol{x}_{k,j})\right) \end{aligned} \tag{3.2}$$

In more detail, t_k regulates the uncertainty of the user's estimates in the following way. If a user is fully trustworthy ($t_k = 1$), then the uncertainty in its reported estimate corresponds to its reported precision, $\theta_{k,j}$. Otherwise, if a user is untrustworthy ($t_k \ll 1$) then the uncertainty in its reported estimate will increase up to having an approximately uniform probability density over $\boldsymbol{x}$ as t_k is close to 0. In this way, the model assumes that a trustworthy user must accurately report the uncertainty of its observations. In fact, having a user that confidently reports wrong observations is more problematic from the fusion point of view, compared to having wrong observations reported with low confidence. In this respect, by using trust parameters to scale the reported precisions, our user trust model is close to the idea of other existing trust approaches used in different non-crowdsourcing domains, such as service provision, web services and grid computing (Teacy, 2006). As an example, Figure 3.2 shows the scaling effect of a trustworthiness parameter for a one-dimensional Gaussian estimate, $\boldsymbol{e}_{k,j} = \langle 16, 3\rangle$, varying trustworthiness, $t_k = \{1, 0.5, 0.2\}$. In particular, it can be seen that the p.d.f. flattens on the x-axis as an effect of increasing its variance proportionally to t_k.

Having now defined our user's trust model, we discuss the integration of such a model in our fusion framework (thus, satisfying our Req. 1) in the following sub–section.

3.1.2 A Trust-based Fusion Model

As per the objectives laid down in Chapter 1, our ultimate goal is to fuse the set of crowdsourced reports into a single estimate that accurately predicts the item's value, i.e. $\boldsymbol{\mu}$. Crucially, such a fusion must be aware of the different trustworthiness levels

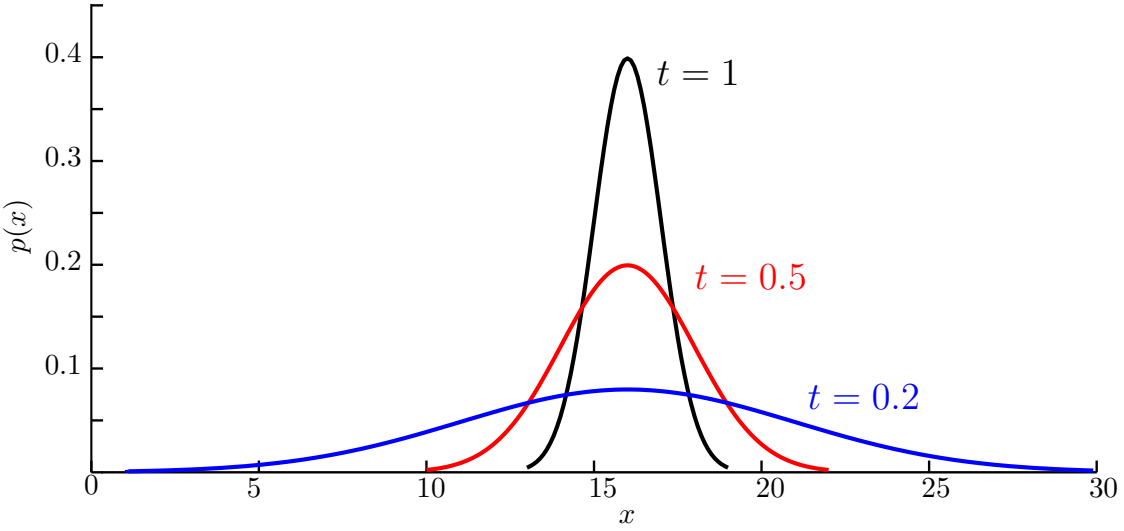


Figure 3.2: Effect of the trust parameter (t) as noise scaling factor of a Gaussian estimate.

of the users. To define such a method, we draw from the set of data fusion methods discussed in Chapter 2. In particular, we cast our problem in a single–hypothesis fusion setting (see Section 2.3.2). This choice is motivated by the fact that, in our setting, we deal with reports of a stationary item, therefore the unique fusion hypothesis relates to the true value of such an item. In this setting, we use covariance intersection (CI), that is the standard method for single-hypothesis fusion, as a baseline technique to derive our new trust-based fusion method, which we call trust–based CI. In more detail, the trust–based CI fusion of the estimates included in $\boldsymbol{R}$ given $\boldsymbol{t}$ is denoted as $f_{\boldsymbol{R}}(\boldsymbol{x}|\boldsymbol{t})$ and is expressed as follows:

$$f_{\boldsymbol{R}}(\boldsymbol{x}|\boldsymbol{t}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{x_f}, \theta_f \mathrm{I}_d) \tag{3.3}$$

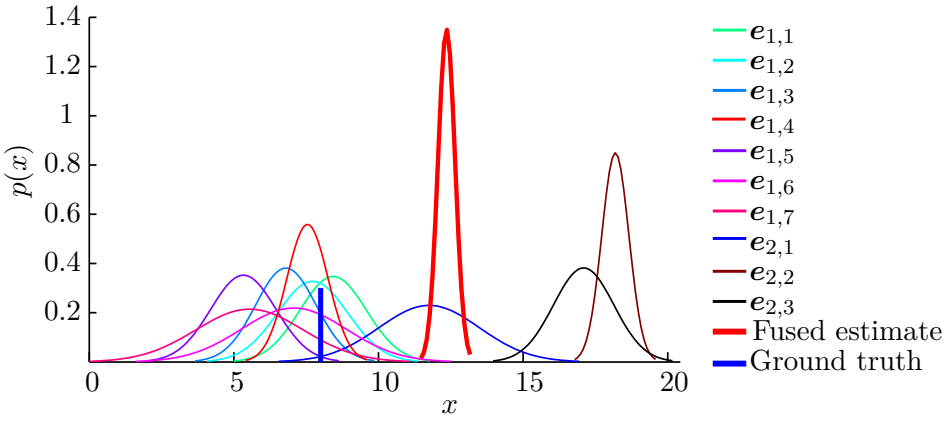
$$\theta_f = \sum_{k=1}^{K} t_k \sum_{j=1}^{p_k} \theta_{k,j} \tag{3.4}$$

$$\boldsymbol{x}_f = \theta_f^{-1} \sum_{k=1}^{K} t_k \sum_{j=1}^{p_k} \boldsymbol{x}_{k,j} \theta_{k,j} \tag{3.5}$$

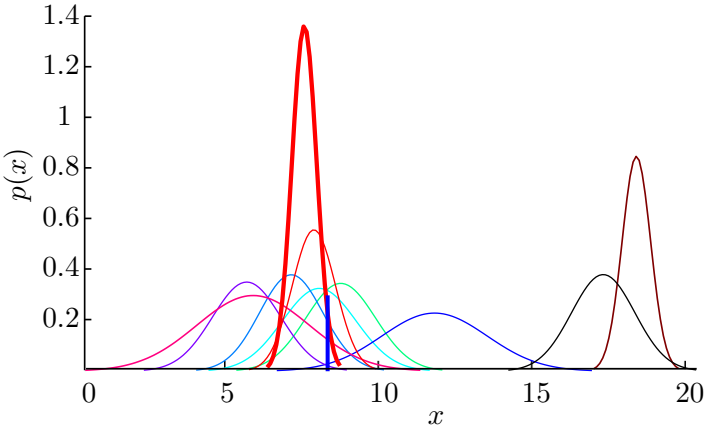
That is, the trust–based CI fusion of a set of Gaussian estimates is a new Gaussian p.d.f. over $\boldsymbol{x}$ with mean and variance weighted by the trust parameters of the user. In particular, for the bivariate case (i.e., $d = 2$), the trust–based CI equation of the fused mean $\boldsymbol{x}_f$ (Equation 3.5) can also be rewritten as follows:

$$x_{f,1} = \theta_f^{-1} \sum_{k=1}^{K} t_k \left(\sum_{j=1}^{p_k} x_{k,j,1} \theta_{k,j} \right)$$

$$x_{f,2} = \theta_f^{-1} \sum_{k=1}^{K} t_k \left(\sum_{j=1}^{p_k} x_{k,j,2} \theta_{k,j} \right)$$



(a) Standard CI fusion



(b) Trust-based fusion

Figure 3.3: Example of 10 Gaussian estimates fused through CI fusion (a) and trust-based CI (b).

where $\boldsymbol{x}_{k,j} = \langle x_{k,j,1}, x_{k,j,2} \rangle$ and $\boldsymbol{x_f} = \langle x_{f,1}, x_{f,2} \rangle$. Recall $\theta_{k,j}$ is always a univariate term since we use the same precision for all the input dimensions (see Section 3.1.1). Specifically, our trust–based CI merges the estimates as jointly weighted by θ_k and by t_k. This determines that the trustworthy reports are considered with a higher weight, while the untrustworthy ones are gradually downgraded in the fusion. In this respect, it could be argued that this way to define a trust–based fusion might be vulnerable to collusion attacks whereby the untrustworthy reports dominate over the trustworthy ones. However, in addition to the fact that this situation is excluded by the majority assumption underlying our model, we notice that collision attacks are not yet seen as a serious issue in crowdsourcing, due to the openness of the crowd which does not facilitate user–to–user agreements. This supports our choice of using CI in our fusion model for crowdsourced estimates that we described in Equations 3.4 and 3.5.

Thus, the property of trust–based CI of using users' trustworthiness to weight each report overcomes the limitation of the standard CI which considers all the estimates as equally trustworthy. Comparing these two fusion approaches, Figure 3.3 shows an example of fusing 10 one–dimensional Gaussian estimates submitted by two users with

$\boldsymbol{\mu} = 8$. Specifically, user 1 reports $\{\boldsymbol{e}_{1,1}, \ldots, \boldsymbol{e}_{1,7}\}$ and user 2 reports $\{\boldsymbol{e}_{2,1}, \ldots, \boldsymbol{e}_{2,3}\}$. In this example, it can be seen that the trust–based CI fusion (Figure 3.3 (b)) obtained by setting trustworthiness parameters to $t_1 = 1$ and $t_2 = 0$ is much closer to $\boldsymbol{\mu}$ compared the standard CI fusion (Figure 3.3 (a)). This is due to the correct assignment of trustworthiness which defines a zero weight for the estimates reported by user 2 that are inconsistent with $\boldsymbol{\mu}$. With this, we can see that the accuracy of this trust-based fusion method is strictly dependent on the right values of trustworthiness assigned to the users, which are not known beforehand. Thus, we next show a statistical method to estimate $\boldsymbol{t}$ from $\boldsymbol{e}$.

3.1.3 Maximum Likelihood Inference

We wish to infer the values of the trust parameters from the set of crowd reported estimates. In our model formulation, we do not specifically require to account for the uncertainty in the estimation of such parameters, i.e., they can be set to static values as opposed to estimating them through probability distributions. Therefore, we choose to adopt a maximum likelihood (ML) approach to our inference problem. Specifically, ML is a standard frequentist approach for learning the values of unobserved parameters in a statistical model by setting them to the values that maximise the probability of the observed dataset (likelihood) (Bishop, 2006). In order to apply ML to our model, we start by defining the likelihood of one trust parameter given a single report as follows. Without loss of generality, let us assume a bivariate setting (i.e. $d = 2$) and recall our notation $\boldsymbol{x} = \langle x_1, x_2 \rangle$, $\boldsymbol{x}_{k,j} = \langle x_{k,j,1}, x_{k,j,2} \rangle$ and $\boldsymbol{x}_f = \langle x_{f,1}, x_{f,2} \rangle$. Then, the likelihood of t_k given $\boldsymbol{e}_{k,j}$ and $f(\boldsymbol{x}|\boldsymbol{e}, \boldsymbol{t})$ is the joint product of the two p.d.f.s described by Equations 3.2 and 3.3, integrated over the two-dimensional space. Formally:

$$
\begin{aligned}
L(t_k|\boldsymbol{r_{k,j}}, f_{\boldsymbol{R}}) &= \int_{\mathbb{R}^2} p(\boldsymbol{x}|\boldsymbol{e}_{k,j}, t_k) f(\boldsymbol{x}|\boldsymbol{e}, \boldsymbol{t}) dx \\
&= \int_{x_1} \int_{x_2} \frac{t_k \theta_{k,j} \theta_f}{4\pi^2} \exp\Big(-\frac{1}{2} (t_k \theta_{k,j} (x_1 - x_{k,j,1})^2 \\
&\quad + t_k \theta_{k,j} (x_2 - x_{k,j,2})^2 + \theta_f (x_1 - x_{f,1})^2 \\
&\quad + \theta_f (x_2 - x_{f,2})^2 \Big) dx_1 dx_2
\end{aligned} \tag{3.6}
$$

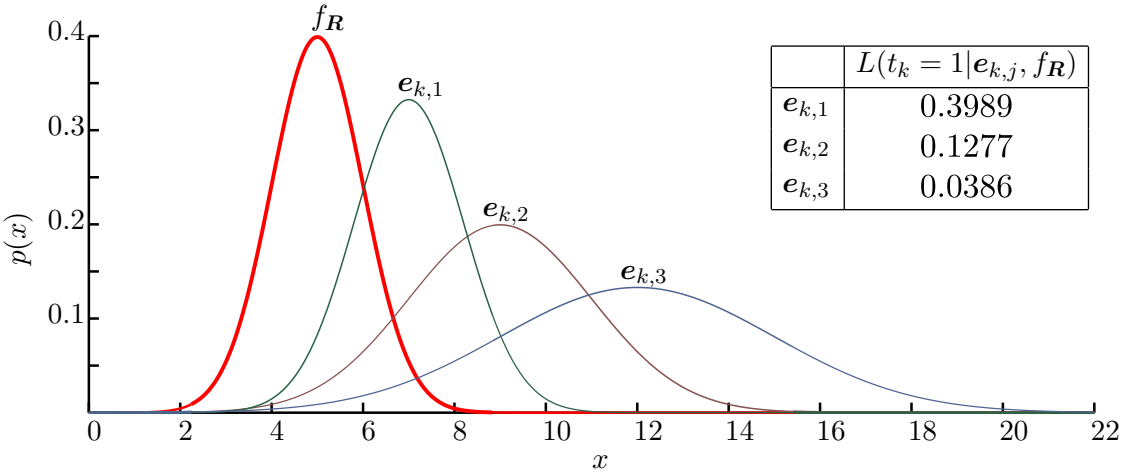


Figure 3.4: Likelihood of three reports $\boldsymbol{e}_{k,1}$, $\boldsymbol{e}_{k,2}$, $\boldsymbol{e}_{k,3}$ over the fused estimate $f_{\boldsymbol{R}}$.

Then, applying the basic rules of Gaussian integration, Equation 3.6 can be solved in closed form as follows:

$$\begin{aligned} L(t_k|\boldsymbol{e}_{k,j}, f_{\boldsymbol{R}}) = & \frac{1}{2\pi(\frac{1}{t_k\theta_{k,j}} + \frac{1}{\theta_f})} \exp\Bigg(- \frac{t_k\theta_{k,j}}{2}(x_{k,j,1} + x_{k,j,2})^2 \\ & + \frac{(t_k\theta_{k,j}x_{k,j,1} + \theta_f x_{f,1})^2 + (t_k\theta_{k,j}x_{k,j,2} + \theta_f x_{f,2})^2}{2(t_k\theta_{k,j} + \theta_f)} \\ & - \frac{\theta_f}{2}(x_{f,1} + y_{f,2})^2\Bigg) \end{aligned} \tag{3.7}$$

That is, the likelihood of t_k for a single report is taken as the product of the probabilities assigned by $\boldsymbol{e}_{k,j}$ and $f_{\boldsymbol{R}}$ 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 $\boldsymbol{x}_{k,j}$ and between $\boldsymbol{x}_{k,j}$ and $\boldsymbol{x}_f$, by $t_k\theta_{k,j}$ and θ_f respectively. In more detail, Figure 3.4 shows a numerical example of computing the likelihood of $t_k = 1$ with three reports, $\boldsymbol{e}_{k,1} = \langle 7, 0.7\rangle, \boldsymbol{e}_{k,2} = \langle 9, 0.25\rangle, \boldsymbol{e}_{k,3} = \langle 12, 0.11\rangle$ and $f_{\boldsymbol{R}} = \langle 5, 1\rangle$. In particular, it can be seen that the likelihood value is proportional to the area shared between $\boldsymbol{e}_{k,j}$ and $f_{\boldsymbol{R}}$ i.e. the further $\boldsymbol{e}_k$ is from $f_{\boldsymbol{R}}$, the lower is its likelihood. In this example $\boldsymbol{e}_{k,1}$ and $\boldsymbol{e}_{k,3}$ are the most and least likely estimate given $f_{\boldsymbol{R}}$, respectively.

Next, assuming independence between t_i and t_j for $i \neq j$, i.e. assuming that the users are independently trustworthy, the joint likelihood of $\boldsymbol{t}$ given $\boldsymbol{e}$ is the product of the individual likelihood terms. That is:

$$\begin{aligned} L(\boldsymbol{t}|\boldsymbol{R}) &= \prod_{k=1}^{K}\prod_{j=1}^{p_k} L(t_k|\boldsymbol{e}_{k,j}, f_{\boldsymbol{R}}) \\ &= \prod_{k=1}^{K}\prod_{j=1}^{p_k}\left(\int_{R^d} p(\boldsymbol{x}|\boldsymbol{e}_{k,j}, t_k) f_{\boldsymbol{R}}(\boldsymbol{x}|\boldsymbol{t})dx\right) \end{aligned} \tag{3.8}$$

Notice that the likelihood does not directly depend on $f_{\boldsymbol{R}}$ as this is implicitly derived from $\boldsymbol{e}$ and $\boldsymbol{t}$ which are already function parameters (see Equation 3.8). Then, by taking the logarithm of this function we obtain the following expression:

$$\begin{aligned}\ln\left(L(\boldsymbol{t}|R)\right) &= \sum_{k=1}^{K}\sum_{j=1}^{p_k}\ln\left(L(t_k|\boldsymbol{e}_{k,j}, f_{\boldsymbol{R}})\right) \\ &= -\,p\ln(2\pi) + \sum_{k=1}^{K}\sum_{j=1}^{p_k}\Bigg(\ln(t_k\theta_{k,j}+\theta_f) + \ln(t_k\theta_{k,j}\theta_f) \\ &\quad + \frac{(x_{k,j,1}t_k\theta_{k,j}+x_{f,1}\theta_f)^2 + (x_{k,j,2}t_k\theta_{k,j}+x_{f,2}\theta_f)^2}{2(t_k\theta_{k,j}+\theta_f)} \\ &\quad - \frac{t_k\theta_{k,j}}{2}(x_{k,j,1}+x_{k,j,2})^2 - \frac{\theta_f}{2}(x_{f,1}+x_{f,2})^2\Bigg)\end{aligned} \tag{3.9}$$

where $p = \sum_{k=1}^{K} p_k$ is the total number of reports. Finally, factoring in the expressions of θ_f and $\boldsymbol{x}_f$ (Equations 3.7, 3.8 and 3.9), and maximising this function, we obtain the ML estimates of $\boldsymbol{t}$. That is:

$$\boldsymbol{t}_{\mathrm{ML}} = \arg\max_{\boldsymbol{t}} \sum_{k=1}^{K}\sum_{j=1}^{p_k}\ln\left(L(t_k|\boldsymbol{e}_{k,j}, f_{\boldsymbol{R}})\right) \tag{3.10}$$

In so doing, we notice that there are two singularities in the function for $t_k = -\theta_f/\theta_k$ and $t_k = 0$ (see Equation 3.9). We discuss these two cases individually. Specifically, the case of $t_k = -\theta_f/\theta_{k,j}$ is excluded by our initial assumptions of having $\theta_{k,j}$ and t_k positively defined (see Section 3.1.1). The case of $t_k = 0$ implies that the trustworthiness value of any report cannot be set to zero as this would give an infinite variance which is not tractable numerically. To avoid this case, we can set the range of t_k to be open in 0, i.e. $t_k \in (0, 1]$, thus approximating the value of an untrustworthy user with a small value ϵ, i.e. $t_k \in [\epsilon, 1]$. Having now described our model formally, an algorithm for performing the optimisation step of the maximum likelihood inference of $\boldsymbol{t}$ is provided in the next section.

3.2 The MaxTrust Algorithm

In this section, we describe our algorithm, named MaxTrust, for the polynomial computation of ML estimates of the parameters $\boldsymbol{t}$, $\boldsymbol{x}_f$ and θ_f given $\boldsymbol{R}$. Before going into further detail, we discuss the following two computational aspects concerning the analysis of our model. Firstly, the non-linear expression of the likelihood described in Equation 3.9 does not allow a closed form analytical maximisation. Thus, we need to use numerical optimisation to carry out such a function maximisation. Secondly, the t_k terms are not separable in this function and there is a mutual dependency between the trustworthiness parameters, i.e. when we update t_k we also need to update the remaining $t_{(-i)}$ parameters. Thus, a natural way to solve this computationally is to iterate over setting values

of each t_k parameter until these converge. To do so, we use *Jacobi* iteration (Hageman and Young, 2004), which is a standard numerical technique for solving non-linear systems. In detail, Jacobi iteration sequentially updates only one system parameter at a time using the values of the previous iteration[1]. Drawing these two points together, our MaxTrust algorithm can now be described (see Algorithm 3.1).

In more detail, in step 1, the algorithm starts with an initial guess of t_k. For this step, random initialisations of the trust parameters in multiple runs of the algorithm are useful to avoid suboptimal solutions, although we found that the initial configuration with all the t_k parameters set to one provided faster convergence in our experiments. Then, steps 3-6 implement the Jacobi loop in which, at the h-th iteration, $t_k^{(h)}$ is updated through the line search maximisation of $f_{\boldsymbol{R}}$ with only t_k left as a free parameter using the values of $t_{-k}^{(h-1)}$ from the previous iteration (step 5). After convergence, that was empirically found to be reached in approximately 5 - 20 iterations, the algorithm returns the trustworthiness values $\boldsymbol{t}^{(h)}$ and the fused estimate $\langle \boldsymbol{x}_f, \theta_f \rangle$ from the last iteration (steps 7-8). In this way, MaxTrust computes the output in $O(epochs \times K|S|)$ polynomial time, where $|S|$ is the number of samples used to perform the line search function maximisation in step 5. While this procedure is not guaranteed to find the optimal solution, it can be computed in polynomial time which is more efficient than the optimal search of the optimal maximiser which has exponential time complexity ($O(|S|^K)$). From this, MaxTrust is suitable for multiple, cheaper, runs to improve the quality of the learning output. Having now described our algorithm, its empirical evaluation is presented in the next section.

3.3 Experimental Evaluation

In this section, we present the results of the evaluation of MaxTrust in performing multiple fusion tasks. Specifically, our first experiment aims to test the robustness of MaxTrust in various crowdsourcing settings with different levels of trustworthiness of the crowd using synthetic data (Section 3.3.2). Then, our second experiment aims to show the efficacy of MaxTrust in the real-world crowdsourcing application of cell–tower mapping (Section 3.3.3).

3.3.1 Experimental Setup

We compare the performance of MaxTrust in terms of accuracy and informativeness of its fusions agains several state–of–the–art approaches. To do so, we describe all the approaches that we evaluate and the metrics used to measure their performances.

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

Algorithm 3.1 MaxTrust

Variables :

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

Algorithm *MaxTrust*($\boldsymbol{R}$)

1: Start with uniform trustworthiness:
$\boldsymbol{t}^{(0)} := \langle 1, \ldots, 1 \rangle$
2: $h := 0$
3: **while** ($|\boldsymbol{t}^{(s-1)} - \boldsymbol{t}^{(s)}| < err$ **and** $h < epochs$) **do**
4: $s := s + 1$
5: **for** $k := 1 : K$ **do**
$t_k^{(s)} := \arg\max_{\boldsymbol{t}} L(\langle t, t_{-k}^{(s-1)} \rangle | \boldsymbol{e})$ (Equation 3.10)
end for
6: **end while**
7: $\theta_f = \sum_{k=1}^{K} t_k^{(s)} \sum_{j=1}^{p_k} \theta_{k,j}$ (Equation 3.4)
$\boldsymbol{x}_f = \theta_f^{-1} \sum_{k=1}^{K} t_k^{(s)} \sum_{j=1}^{p_k} \boldsymbol{x}_{k,j} \theta_{k,j}$ (Equation 3.6)
8: **return** $(\boldsymbol{t}^{(s)}, \boldsymbol{x_f}, \theta_f)$

3.3.1.1 Benchmarks

In our evaluation, we compare MaxTrust against five methods taken from the two main classes of trust–based and non–trust fusion approaches that we discussed in Chapter 2. These are described as follows:

- **Non-Trust Fusion Approaches**: These are the approaches that do not explicitly consider a user's trustworthiness in the fusion method. In this class, we consider the following two algorithms that are representative of the fusion approaches discussed in Chapter 2 (Section 2.3):
  - **Covariance Intersection (CI)**: The standard CI fusion method for the linear fusion of multiple Gaussian estimates as described in Section 2.3.2.1. Notice that this method is equivalent to MaxTrust without considering the trust parameters, i.e., $t_k = 1 \quad \forall k$.
  - **Covariance Union (CU)**: The standard CU fusion method that merges the reports by taking the union of their Gaussian estimates as described in Section 2.3.2.2.
- **Trust-Based Fusion Approaches**: These are approaches that, similarly to MaxTrust, compute the fusion of the reports combined with the learning of user's trustworthiness as described in Chapter 2 (Section 2.4). In this class, we consider the following three algorithms:
  - **Optimal Trust-Based Fusion (OptTrust)**: This is a *hypothetical* optimal algorithm given by our trust–based CI in which we assume we have correct knowledge of the trustworthiness of each user. Notice that we can only evaluate this algorithm for the case of synthetic data where the ground truth of the item's value is available.
  - **Local Outlier Factor Fusion (LOF)**: This algorithm is based on the density-based outlier detection approach described by Algorithm 2.1. Specifically, LOF applies outlier detection to the report set based on an outlier threshold $l = 1$ (using $k = 5$ as the number of nearest neighbours), and then applies CI to fuse the remaining inlier estimates (see Section 2.4.2.1).
  - **Reece Method (RM)**: This is the algorithm presented by Reece et al. (2009) for fusing sensor estimates combined with the inference of the sensor trustworthiness as described by Algorithm 2.2 (see Section 2.4.2.2). In our experiments, we set the trust threshold of RM to $\beta = 3$ which follows the setting that the authors suggest in their paper.

In summary, a set of six algorithms {CI, CU, LOF, OptimalTrust, RM, MaxTrust} were tested as representative benchmarks from both the trust and the non-trust based fusion approaches.

3.3.1.2 Accuracy Metrics

We are interested in measuring the accuracy of the fusion algorithms in terms of their average error and estimated uncertainty. In fact, a good fusion algorithm is expected to produce estimates with low error and uncertainty with respect to the item's true value. To this end, we consider two standard metrics that to measure the accuracy of probabilistic estimates with respect to a set of gold standards (i.e., the ground truth of the items) used for validation. These are the *root mean square error* (RMSE) and the *continuous rank probability score* (CRPS), respectively. Specifically, the RMSE is measured between the predictive mean $\boldsymbol{x}_f^i$ of the fused estimate and the items' true value $\boldsymbol{\mu}_i$, averaged over N items indexed by i. That is:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_f^i - \boldsymbol{\mu}_i)^2}$$

Notice that the RMSE is measured only based on the fused mean $\boldsymbol{x}_f^i$.

To take into account the full predictive distribution of an item that is computed by the fusion algorithm, we consider the CRPS as an additional , more comprehensive, accuracy metric that is computed from both the fused mean and the fused precision. This is is a non-local scoring rule particularly suitable for scoring probabilistic predictors that is commonly used in statistics (Wallsten et al., 1997) and increasingly in artificial intelligence (AI) (Quinonero-Candela et al., 2006). This is defined as follows:

$$CRPS = \frac{1}{N}\sum_{i=1}^{N}\int_{x=-\infty}^{x=+\infty}(F_{\boldsymbol{e}}^{(i)}(\boldsymbol{x}) - F_i^*(\boldsymbol{x}))^2 d\boldsymbol{x}$$

where $F_{\boldsymbol{e}}^{(i)}$ is the predictive cumulative density function (c.d.f) of the predictive estimate of $\boldsymbol{\mu_i}$ and F_i^* is the *Heaviside* step function, with step from 0 to 1 in $\boldsymbol{\mu}_i$. In particular, the CRPS has the properties of *properness* (i.e. the true generative distribution has the best score) and *distance-sensitive* scores (i.e. its score is proportional to predictive probability mass placed near the true value) (Kohonen and Suomela, 2006). In our experiments, we need to compute the CRPS of Gaussian predictions in the univariate case of the first experiment with synthetic data, and in the bivariate case of the second experiment with real location data. For the univariate case, the CRPS of N Gaussian predictions can be expressed in closed form as follows:

$$CRPS = \frac{1}{N}\sum_{i=1}^{N}\sigma_f^i\left(-\frac{1}{\sqrt{\pi}} + 2\varphi\left(\frac{\mu_k - x_f^i}{\sigma_f^i}\right) + \frac{\mu_k - x_f^i}{\sigma_f^i}\left(2\phi\left(\frac{\mu_k - x_f^i}{\sigma_f^i}\right) - 1\right)\right)$$

where φ and ϕ denote the probability density function and the cumulative distribution of a standard normal random variable, respectively. For the bivariate case, we use an

approximate version of the CRPS based on solving the CRPS integral via discretisation. The details on such a CRPS approximation are described in Appendix A.

Finally, for RM and MaxTrust, we also measure the error in their trustworthiness estimates in terms of the RMSE between the estimated t_k and the correct user's trust value t_k^*, which is known only for the case of synthetic data. That is:

$$t\_error = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(t_k - t_k^*)^2} \tag{3.11}$$

The results of our experiments are presented in the following sections.

3.3.2 Experiments on Synthetic Data

In this first experiment we evaluate MaxTrust on the task of fusing multiple reports in a continuous one–dimensional space using synthetic data. To simulate this scenario, we place a hypothetical item at $\mu = 8$. For the purpose of this experiment that aims to validate the performance of our fusion algorithm, we use a single–report setting in which each user reports only one estimate. However, the same experiment can be trivially extended to a multi-report setting. Then, we generate 50 Gaussian random estimates of μ, one for each user, i.e. $K = 50$ and $p_k = 1 \quad \forall k$ (for simplicity, we omit the index j in this description). Then, following the assumptions of the MaxTrust's model made in Section 3.1, we simulate a setting in which each user has a bounded random precision and each trustworthy observation is generated according to the user's precision. Specifically, we randomly sample the parameters of each report as $\theta_k \sim U[0.2, 10]$, which means that the most (least) accurate users estimate the item within a 18% (130%) error bound, and $x_k \sim \mathcal{N}(\mu, \theta_k)$. Next, we simulate a percentage ρ of randomly selected untrustworthy reports by adding a random bias w to x_k as follows:

$$\hat{x}_k = x_k + w \qquad w \sim \pm U[2, 10] \tag{3.12}$$

That is, by uniformly sampling w from $[2, 10]$, we avoid the situation in which two untrustworthy estimates are symmetric, i.e. their bias is $+w$ and $-w$ respectively, and so balance their noise in the linear fusion. In addition, by randomly choosing the sign of w in each run, we avoid the bias of considering only positive or negative noise terms in our results. In this setting, we can validate the trust values estimated by MaxTrust and RM against the true values of trust that we assume to be $t_k^* = 0$ for the untrustworthy reports and $t_k^* = 1$ for the trustworthy ones.

In more detail, Figure 3.5 shows the results of the six tested algorithms for $N = 600$ runs with a percentage of untrustworthy reports increasingly set to $\rho = \{10, 20, 30, 40, 50, 60\}$ (the error bars are not visible due to their very small values). Specifically, Figure 3.5 (a) shows that, as expected, the RMSE of all the algorithms increases for higher ρ values

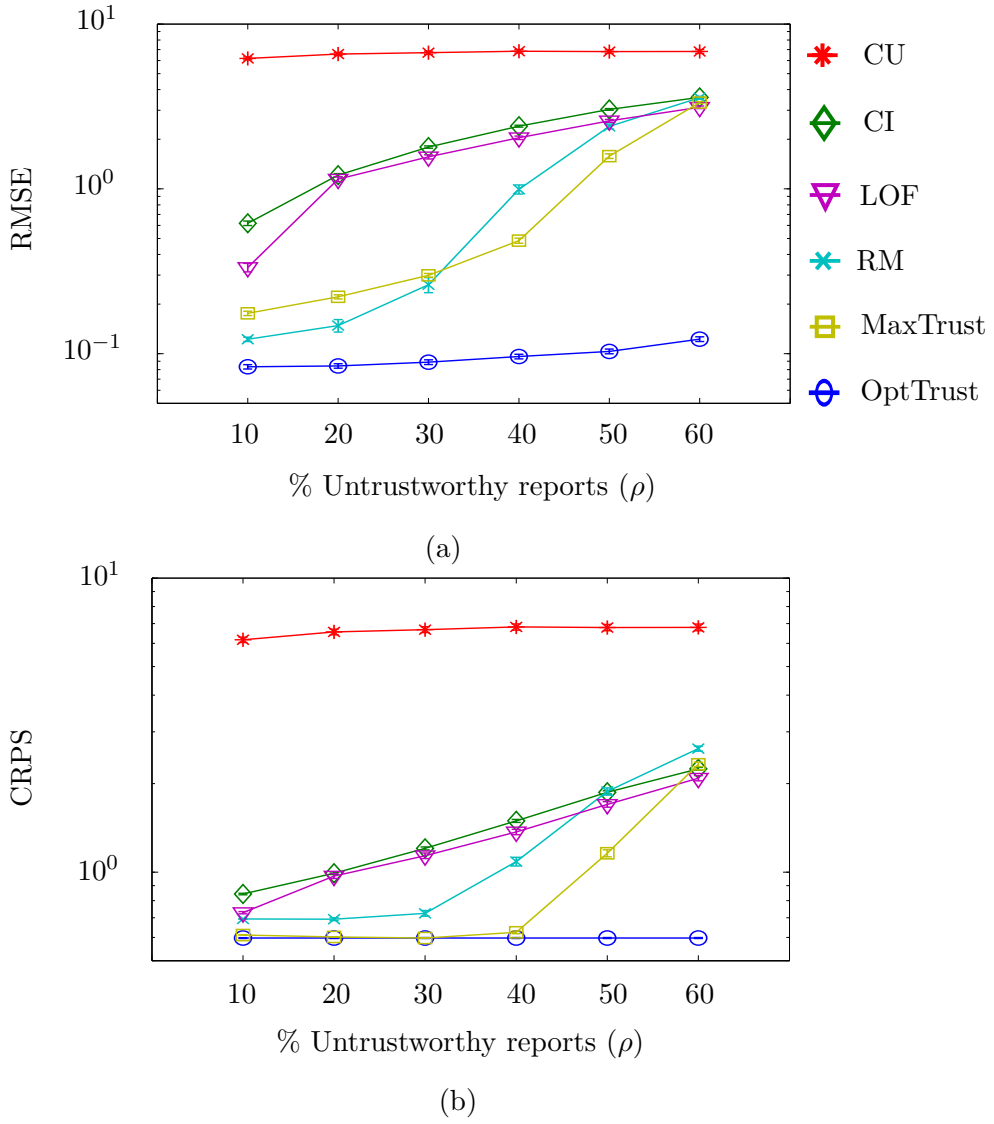


Figure 3.5: Plot of the RMSE (a) and the CRPS (b) of the six algorithms in the experiment with synthetic data. The x-axis is the percentage of untrustworthy reports controlled by ρ.

and that OptimalTrust has always approximately zero error due to its prior knowledge of the trust values. Importantly, the two trust–based fusion algorithms (MaxTrust and RM) outperform the non–trust algorithms (CI, CU) for the values of ρ between 10% and 50%. Notice that $\rho = 60\%$ corresponds to the extreme and least practical case when the majority of the reports is not guarantee to be trustworthy; thus providing no guarantees that we could extract correct knowledge about the item from such reports. As this case violates our initial majority assumption, it is less likely that the trust–based methods can make right decisions about identifying trustworthy users and in turn provide accurate estimate.

Comparing the two trust–based methods, MaxTrust's error is very close to RM for $\rho < 30\%$ and it is lower than RM for $30\% < \rho < 50\%$. In particular, its error is 51% lower than RM when $\rho = 40\%$. In particular, when $\rho = 60\%$, MaxTrust does

	RM	MaxTrust
t_error for trustworthy users ($\hat{t}_k = 1$)	0.67 ± 0.15	$\mathbf{0.31} \pm 0.10$
t_error for untrustworthy users ($\hat{t}_k = 0$)	0.26 ± 0.20	$\mathbf{0.24} \pm 0.11$
Average error	0.46 ± 0.27	$\mathbf{0.28} \pm 0.11$

Table 3.1: The error (*t_error*) of the estimated t_k for RM and MaxTrust evaluated on synthetic data. The lowest error of each row is highlighted in bold.

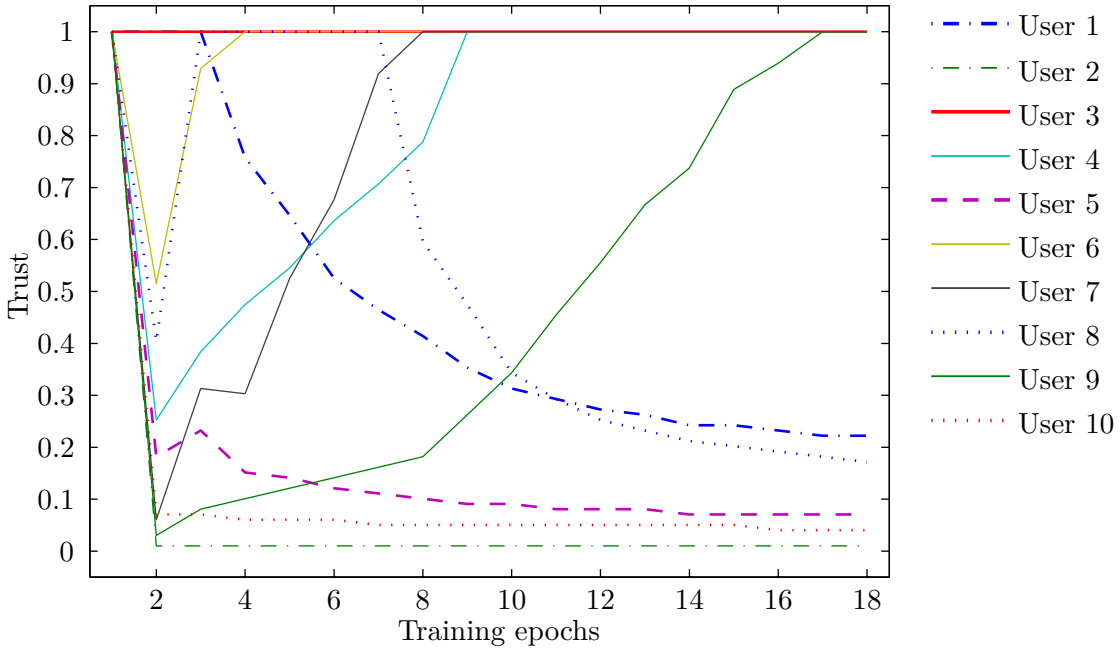


Figure 3.6: The plot of the trust values of each user estimated by MaxTrust at each training epoch from a synthetic dataset with 10 users.

no longer improves accuracy and conform to the errors of CI and RM. The differences between the RMSE of MaxTrust and RM were found to be statistically significant for all the values of ρ, except when $\rho = 30\%$ where the error bars overlap, from a t-test at $\alpha = 5\%$ ($p < 0.001$). Overall, MaxTrust is able to achieve comparable accuracy with a higher presence of untrustworthy users. Furthermore, the CRPS plotted in Figure 3.5 (b) shows MaxTrust has the best scores for all the configurations of $\rho < 60\%$. The statistical significance of these comparisons was also tested with a t-test at $\alpha = 5\%$ ($p < 0.001$) This means that its fusion provides the most informative estimate, i.e., it predicts the item's true value with the lowest uncertainty. In particular, this accuracy gain is due to the more accurate trust learning of MaxTrust, which is also demonstrated by the trust value's estimation errors reported in Table 3.1. In this table, we can see that MaxTrust estimates the trust values with an accuracy that is 39% higher than RM.

To further explain the behaviour of MaxTrust in learning such trust values, Figure 3.6 shows the graph of the values of t_k estimated at each epoch by the MaxTrust's iterative training process described in Algorithm 3.1. Specifically, we run the algorithm on a synthetic dataset with 10 users generated in the same setting described in Section

3.3 when $\rho = 50\%$. The graph shows that the algorithm converges in 18 iterations (convergence error = 10^{-2}) and the trust values change at each iteration based on the new configuration of the likelihood function. Crucially, in the last iteration the trust parameters converge to a set of values in which all the values of the trustworthy users (user 3, 4, 5, 7, 9) and untrustworthy users (user 1, 2, 5, 8, 10) are correctly learnt[2].

Furthermore, we can see from the error plots (Figure 3.5) that CU is consistently the algorithm with the highest error amongst the tested methods. This means that the conservative strategy of unifying the estimates is mostly inaccurate in our scenario. Moreover, LOF marginally improves over CI although its performance might improve by fine tuning its outlier threshold. However this threshold needs to be tuned for each ρ which makes LOF less flexible than MaxTrust which is threshold free. Finally, for $\rho = 60\%$ all the algorithms have comparable errors which is determined by the majority of untrustworthy reports that does not allow any further improvement by the trust-based fusion algorithms.

From these results, we contend that the trust-based approach improves the accuracy of the reports' fusion on our simulated datasets. In particular, MaxTrust is the algorithm that is the most robust against larger numbers of untrustworthy users. It also achieves a good accuracy, which is never with lowest uncertainty amongst the tested methods. To further strengthen this empirical claim, we now explore the performance of MaxTrust with a real–world dataset.

3.3.3 Experiments on Real Data

In this second experiment, we focus on the application of cell tower mapping from crowdsourced cell detections that we introduced in Chapter 1. As already discussed, this is a key application for the mobile phone industry that involves all the major phone manufacturers, including Apple, Google and Microsoft-Nokia. Specifically, the objective is to build cell tower maps to improve the positioning system of their mobile phones (Ahern et al., 2006). In fact, by having a map of the cell towers located in the phone's local area, triangulation can rapidly give an accurate phone position with a lower battery drain compared to the standard GPS–based localisation. Moreover, cell tower positioning systems also allow phones to localise themselves in indoor environments.

However, the task of mapping the cell towers cannot be easily achieved manually due to frequent updates to the topology of cellular networks and the fact that the network operators do not always make the map of their installed radio masts available. 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

[2]This convergence is only showed empirically.

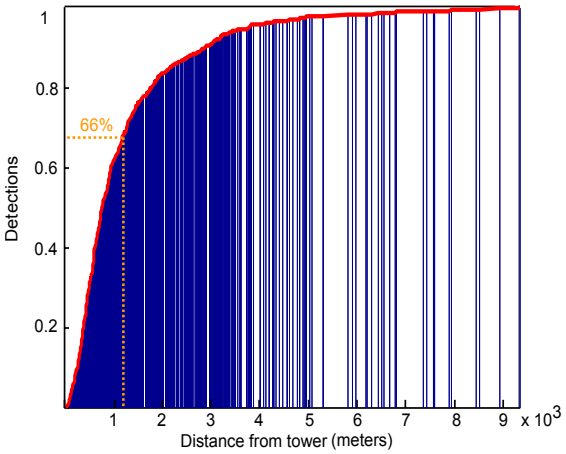


Figure 3.7: Cumulative distribution of cell detections over to the phone–tower distances for the OpenSignal–Cell tower dataset.

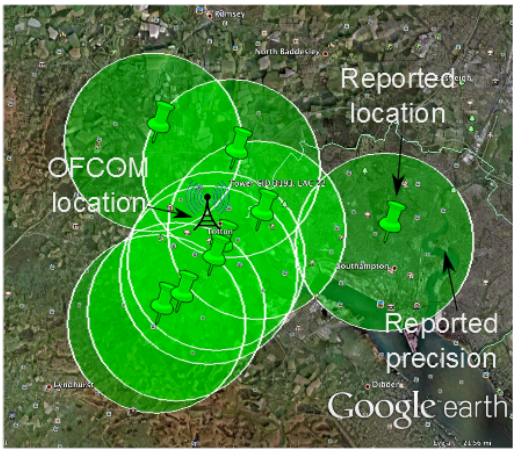


Figure 3.8: Example of the reports for the cell tower (CID 3139, LAC 22) from the OpenSignalMap dataset.

the cellular network to collect data about the cell tower locations.[3] Specifically, GPS–equipped phones can provide the list of the cells scanned in their surrounding area together with the phone's current GPS position. Then, the cell tower location can be determined through merging multiple cell detections reported by the phones from different positions.

However, in so doing, an important issue which relates to the trustworthiness of the reported cell detections is the fact that several inaccuracies may affect the reporting devices. Firstly, the GPS readings are often inaccurate, due the limited update frequency of the device that often returns out–of–date fixes. Secondly, the signal strength readings do not always accurately indicate the current phone–mast distance as the signal may change dynamically across the cell due to obstacles and reflections. Since such inaccuracies are an issue to reliably localise the cell towers, we now show how MaxTrust can be applied to this problem to improve the accuracy of crowdsourced cell tower maps. In particular, we focus on the case of an omni–directional cellular network illustrated in Figure 3.9, namely where a cell tower is placed at the centre of each hexagonal cell. In this network topology, the mast radiates the signal approximately uniformly spherically across the cell. Thus, this type of network suits the assumption of normally distributed probability of the cell tower detection made by MaxTrust in Section 3.1.1

3.3.3.1 Dataset

In this experiment, we used the dataset provided by OpenSignal (`opensignal.com`). This includes 1563 records of anonymised phones that reported detections for a set of 130 omni-directional cell towers (max=46, min=6, avg=12 reports per cell tower) in

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

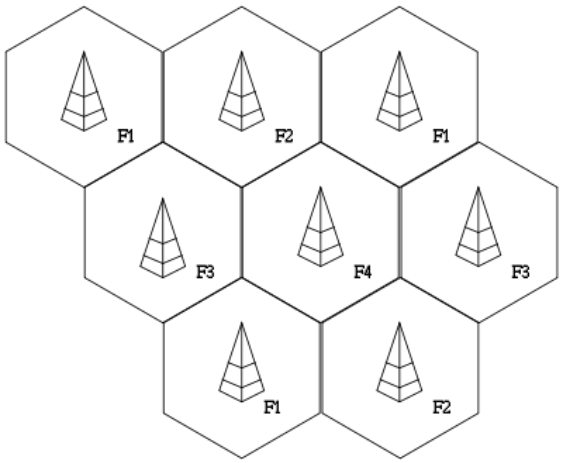


Figure 3.9: Topology of a cellular network with omni–directional cell towers.

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). For privacy reasons, the dataset did not provide any user identifier that could link between the single user and its multiple reports. Therefore, similarl to the previous experiment, we can only consider the single–reporting case in which each user reports only one cell detection. A complete description of this OpenSignal–Cell tower[4] dataset is provided in Appendix B. Furthermore, a second official dataset of cell tower locations is made available by the Authority of UK Communication (OFCOM, `ofcom.org.uk`) which we consider as a more reliable source and, as such, we use the OFCOM data as the ground truth cell tower location to evaluate the performance of our algorithms.

In order to apply MaxTrust to this dataset, we first convert each geographical position (in spherical degrees) to planar coordinates (in meters) applying the following standard equilateral projection:

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

$$x_1 = 111,229 \cdot \cos(Lat) \cdot (lon - lon_0) \tag{3.13}$$

$$x_2 = 111,229 \cdot (lat - lat_0) \tag{3.14}$$

where lat_0 and lon_0 are the coordinates of the point taken as the origin in the planar system (conventionally set to 50.84 N, 1.52 E). Specifically, given that at 50N one degree

[4]This name is adopted to distinguish this dataset from the second OpenSignal–WiFi dataset that will be used in Chapter 4.

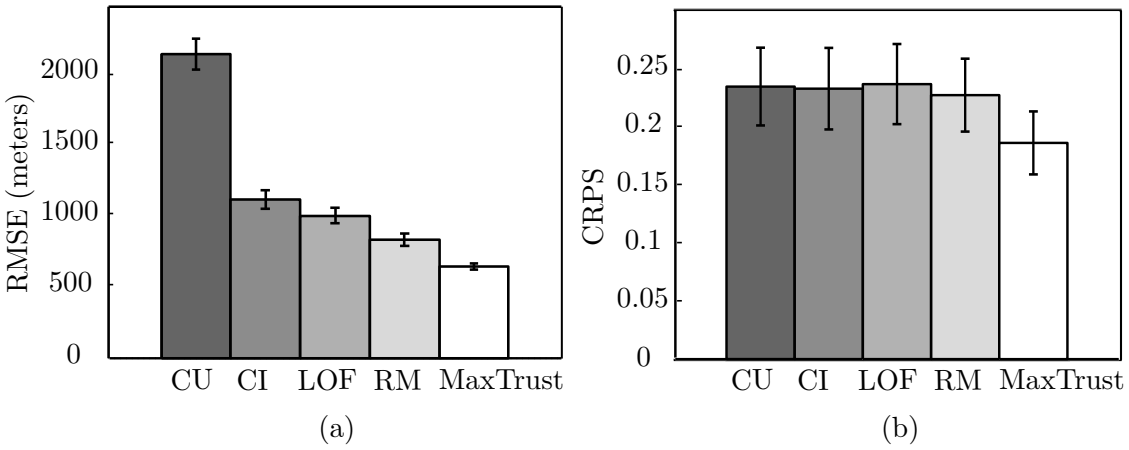


Figure 3.10: Bar plots of the RMSE (a) and CRPS (b) for the five algorithm in estimating the positions of the 130 cell towers.

of latitude corresponds to 111,229 meters, this projection compute the approximate line distance between the origin point and the given location. Such an approximation provides a good level of accuracy in small areas, which is the case of the area of Southampton that we consider. It is also faster to compute compared to the more complex Haversine formula that provides the appropriate trigonometric treatment for spherical distances between two locations but is constrained for numerical computation (Kells et al., 1951).

To set the values of θ_k, i.e. the precision of each reported detection, we notice that 66% of the reports were located within 1200 meters from the true cell tower position. This is also showed by the cumulative distribution curve of the phone-cell tower distance in Figure 3.7. Hence, given that $\sigma_0 = 1200$, we set θ_k as follows:

$$\theta_k = (\sigma\_{\text{GPS}}_k^2 + \sigma_0^2)^{-1} \tag{3.15}$$

where $\sigma\_{\text{GPS}}_k^2$ is the inverse of the GPS precision reported by user k. Summing up, the estimate associated with each cell detection reported by the devices is represented as $\boldsymbol{e}_k = \langle x_{k,1}, x_{k,2}, \theta_k \rangle$, where $\langle x_{k,1}, x_{k,2} \rangle$ is the GPS position of the device and θ_k is the precision of the GPS fix, respectively. In more detail, Figure 3.8 shows the reports collected for the cell (CID 3139, LAC 22) with the circles showing the phone's GPS location and the $3/\theta_k$ range (i.e. the 99% confidence interval) of each report. In this setting, we evaluate the accuracy of the fusion in each cell produced by our algorithms.

3.3.3.2 Results

Figure 3.10 (a) shows the RMSE of the algorithms based on the mean error of their fusions for the set of 130 cell towers. In particular, the error bars are the standard deviation of their mean error. As it can be seen, MaxTrust outperforms all the other methods with a RMSE which is 42% lower than the best non-trust method, CI, and 22%

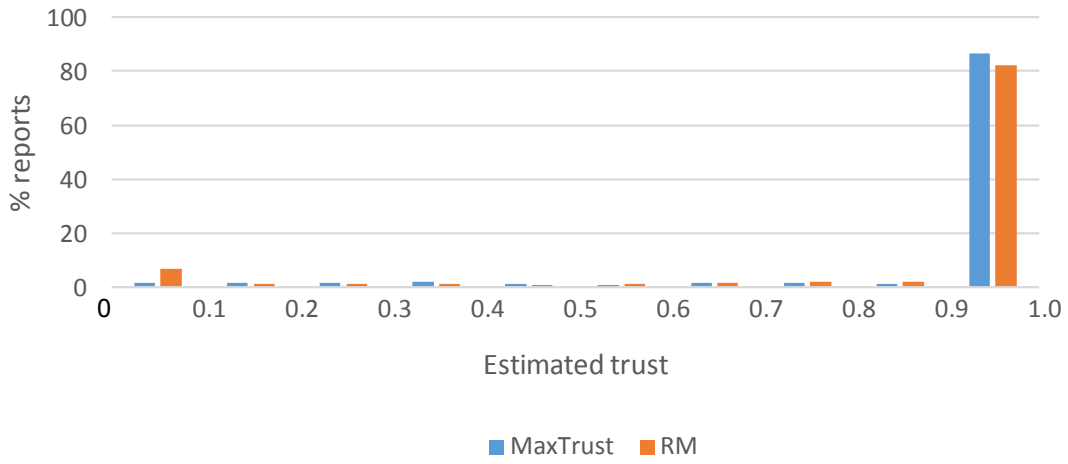


Figure 3.11: The histogram of the trust values estimated by RM and MaxTrust for the reports of the OpenSignal–Cell tower dataset.

lower than the second best trust-based method, RM. Importantly, such a lower error is equivalent to an accuracy gain of 147 meters (average error: RM = 804m vs. MaxTrust = 658m) in localising the cell tower. In addition, Table 3.2 reports the errors (in meters) measured as the line distance between the predictive mean computed by each algorithm and the true cell tower location for a subset of 15 cell towers (the results are similar for other subsets, as is apparent from the result of Figure 3.10 (a)). On such a subset, MaxTrust has the lowest error in 9 out of 15 cells and it has the lowest average error of 147 meters. Thus, this result shows that the parameter–free trust learning method adopted by MaxTrust is able to model the different distributions of reports in each cell more efficiently. Furthermore, Figure 3.10 (b) shows the CRPS of the predictions of the algorithms on the same dataset. From this we can see that MaxTrust has the lowest CRPS, which is 21% lower than the second best score. This means that, similarly to the conclusions of our previous experiment, MaxTrust provides the most informative fusions by having the lowest uncertainty in its predictions. Globally, our results show that the trust–based methods (MaxTrust, LOF and RM) outperform the non–trust ones (CI and CU) and that MaxTrust is the most accurate fusion algorithm having both the lowest RMSE and CRPS on the OpenSignal–Cell tower dataset.

Further insights can be gained from the analysing the trust values estimated by RM and MaxTrust that are showed in the histogram of Figure 3.11. The histogram shows that RM estimates that 6.8% reports have very low trustworthiness, within $[0, 0.1]$. In contrast, MaxTrust estimates that only 1.7% reports have such a range of trustworthiness. This means that RM is more aggressive in selecting untrustworthy reports which is due to the static threshold that it uses to distinguish the trustworthy reports from the untrustworthy ones (see Section 2.4.2). Consequently, RM estimates a lower percentages (82%) of reports with high trustworthiness, i.e., within $[0.9, 1]$, while MaxTrust

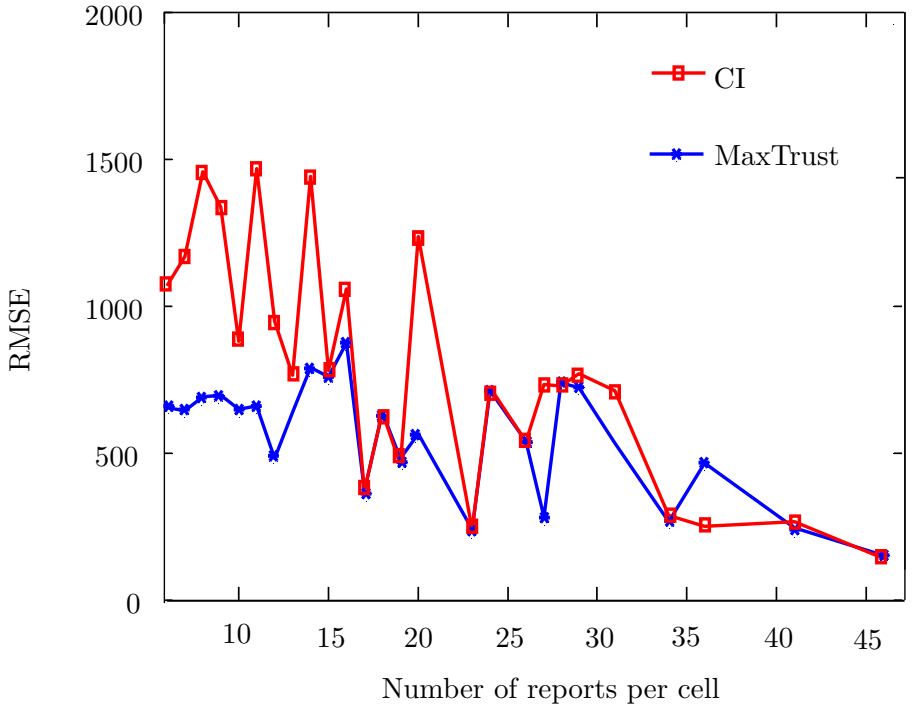


Figure 3.12: Error (y–axis) of CI and MaxTrust over the number of reports (x–axis) in each cell tower.

estimates 87% of reports for the same bin. The percentages for the other bins are more comparable between the two methods. Generally speaking, MaxTrust, which does not require any thresholds for learning trust values, is able to adapt more flexibly to the given dataset and it is less aggressive, but still very effective, than RM in selecting untrustworthy reports.

Another interesting result is the analysis of the RMSE of MaxTrust and CI, i.e. the two best methods among the trust–based and the non–trust based fusion classes, projected on the number of reports available in each cell (Figure 3.12) This analysis shows that MaxTrust minimises the error when the size of the report set is small (i.e. < 15 reports), while its error is comparable to CI for a medium (i.e. between 20 and 35 reports) and a large report set (i.e. > 35 reports). This is explained by the fact that having sufficiently many reports makes it more likely to have a majority of trustworthy reports that mitigate the error of the untrustworthy ones. However, an important finding is that MaxTrust provides better accuracy in the more challenging fusion tasks in which only a few reports are available.

3.4 Summary

In this chapter, we presented our first trust-based fusion method, MaxTrust, which addresses our requirement for merging crowdsourced estimates of stationary items (Req.

1). To address this requirement, we designed a trust model that applies uncertainty scaling techniques to represent the uncertainty of the user's reliability in the reported estimate (thus, also addressing Req. 2). Then, by combining this trust model with our new fusion method, that extends the standard CI by weighting the estimates according to the individual trustworthiness of the users, we are able to learn the user's trustworthiness from the crowd reports through maximum likelihood inference. Finally, we showed that MaxTrust outperforms five of the state–of–the–art methods in producing more accurate and highly informative fusions through experimental evaluation. In particular, we showed that MaxTrust achieves comparable accuracy with higher presence of untrustworthy users in our simulated experiments. Moreover, it provides 42% more accurate (i.e., localisation accuracy is improved by 147 meters) and 22% more informative fusions on the real data given by the OpenSignal cell detection reports .

However, there are several features of the current model that we can possibly leverage for further improvements, Firstly, this model is designed for a single–item setting (i.e., the crowd observes only one item) and can only be applied to a multi–item setting by assuming that the reports of such items are all independent. However, the limitation of doing so is that the model ignores the possible correlation in the user's trustworthiness emerging from its observations of multiple item. These correlations can be exploited to bootstrap trust knowledge from some items to be transferred to new items. Secondly, MaxTrust can only perform batch inference over the observed reports. This does not allow us to consider prior beliefs over the user's trustworthiness of the item's value in the inference or to update the learning outputs when a new report is added to the set. To address these limitations, we will analyse a new configuration of our trust model embedded in a hierarchical Bayesian learning framework for the multi–item setting. In particular, this new Bayesian trust model will be able to maintain probabilistic beliefs over the random variables and effectively transfer them across the users and the items being analysed through Bayesian learning, thus removing the assumption of independent items. This new model will be presented in our next chapter.

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	1527m	1098m	987m	805m	**658m**

Table 3.2: Error (in meters) between the predictive mean produced by the algorithms from the ground truth location (reported in brackets) for 15 cell towers randomly selected from the OpenSignal–Cell tower dataset. The best prediction of each cell is highlighted in bold.

Chapter 4

A Bayesian Trust Model for Fusing Crowdsourced Estimates of Stationary Continuous Quantities

In the previous chapter, we addressed the problem related to inferring reliable knowledge from crowdsourced estimates of a stationary item submitted by possibly untrustworthy users (Req. 1). Specifically, we introduced MaxTrust, an algorithm that simultaneously estimates the item's value and the trustworthiness of each user. We then showed that MaxTrust improves the accuracy of the fused output as a result of its underpinning trust–based fusion approach that de–emphasises the presence of reports coming from untrustworthy users (Req. 2). In MaxTrust, however, we assumed that a user's trustworthiness can be inferred on the basis of the observations that a user reports for a single item (see Section 3.1.1). However, this assumption limits the applicability of MaxTrust to crowdsourcing isolated items (e.g., only one cell tower) or to sets of uncorrelated items (e.g., a set comprising a cell tower, a WiFi hotspot and a balloon position), where it is reasonable to assume that there are no correlations between the reliability of a user's observations of different items[1]. In more detail, the only way to apply MaxTrust to multi–items crowdsourcing setting is to assume completely independent items. However, there are situations where such correlations cannot easily be ignored as they are an important indicator of the user' reliability. For example, we consider the two cases where a user's untrustworthiness is due to a consistent misunderstanding (e.g., the limited accuracy of its phone) or a sporadic error (e.g., due to a particular item that is more difficult to observe). While MaxTrust would treat these two cases in the same manner, a more appropriate trust–based fusion algorithm should be able to distinguish between

[1]This setting was also used in the experiment of MaxTrust applied to the OpenSignal–Cell tower dataset (see Section 3.3.3)

these different types of users' trust behaviour in order to achieve a better handling of their reports.

In this chapter, we improve upon this situation by extending the previous trust model to enable our trust–based fusion framework to account for correlations in the user's reports of multiple items. Specifically, we introduce a new model, that we call BACE (Bayesian Aggregation of Crowdsourced Estimates), that enhances the performance of MaxTrust in multi–item crowdsourcing settings. This makes our framework more suitable for the real–world applications, by introducing the following two refinements. Firstly, in contrast to MaxTrust, BACE assumes that the trust parameters are shared across the items. This allows it to capture a user's trustworthiness over the entire set of items based on patterns of reliability for the reports that emerge across the various items. Notice this is a non trivial extension of MaxTrust because of the fact of having shared trust parameters modifies the data likelihood that we described in Section 3.1.3. Consequently, a new analysis of the BACE's likelihood is required for the purpose of inferring the quantities of interest. Secondly, in BACE, the trust assessment follows directly from Bayesian theory. This has the advantage of providing a full estimation of the uncertainty over the trust parameters[2], which enriches the set of learning outputs and improves the accuracy of the fusion due to a better handling of uncertainty in the inference process given by the Bayesian probabilistic framework. Thus, while addressing the same requirements as MaxTrust in terms of aggregating crowdsourced estimates of a set of stationary continuous items (Req. 1) combined with learning the user's trustworthiness (Req. 2), BACE adds three key advantages.

- Using trust parameters shared across the items, BACE achieves *transfer learning*, whereby the trustworthiness of a user learned from the previous items automatically forms an evidence of its reliability for observing new items. This is a key feature that makes BACE more robust against sparse datasets, i.e., data with (possibly highly) unbalanced numbers of reports per item. In this situation, BACE is able to transfer user's trust knowledge from the items with more reports to other items with less reports, thus providing a more efficient inference of the aggregated estimate.

- BACE is able to account for the uncertainty over the trust parameters more efficiently by adopting a Bayesian learning framework that is optimal under uncertainty (Bishop, 2006). By doing so, BACE provides probabilistic estimates of these parameters as well as handling such an uncertainty more efficiently in the computation of the aggregated values.

- BACE naturally adapts to both batch and online learning settings. That is, it can process the crowd reports either as a single batch or sequentially.

[2]Recall that, in MaxTrust, these two quantities were estimated as a single value rather than as probability densities.

Thus, we make the following contributions:

- We present BACE, a new trust–based Bayesian fusion algorithm for combining sets of crowdsourced estimates in multi–item settings through an efficient transfer learning mechanism of user's trustworthiness realised in a principled Bayesian framework.
- We provide a sampling algorithm to compute the BACE estimates of the true value of each item and the trustworthiness of each user, along with the uncertainty around these quantities, from reported estimates of a set of stationary items.
- We demonstrate that BACE is more efficient than MaxTrust and other existing fusion algorithms in a multi–items setting being able to improve accuracy by 45% in a real experiment with crowdsourced WiFi hotspot location data, and by 48% on synthetic data. Furthermore, it achieves comparable accuracy to existing methods even with 15% more untrustworthy users.

In the remainder of the chapter, we first describe the theory of BACE in Section 4.1 and detail its Bayesian inference in Section 4.2. We then describe a sampling algorithm that enables approximate but tractable inference in our proposed model in Section 4.3. Subsequently, we present our empirical results analysing both real data and synthetic data in Section 4.4. Finally, we summarise our conclusions against the requirements of this thesis in Section 4.5.

4.1 Model Description

In this section, we formally describe our BACE model. First, we summarise the notation introduced in Chapter 3 (see Section 3.1) that provides a common basis for both MaxTrust and BACE. Specifically, suppose that there is a set of M multivariate items (such as cell-tower positions) that we wish to estimate given reported observations from a crowd of K users. For each item i, we define the vector $\boldsymbol{\mu}_i \in \mathbb{R}^d$ to be the unobserved item's true value, for which we receive a set of $p_{k,i}$ observations from each user k. In each case, the jth observation from k about i is a pair, $\langle \boldsymbol{x}_{k,i,j}, \theta_{k,i,j} \rangle$, where $\boldsymbol{x}_{k,i,j}$ is an estimate of $\boldsymbol{\mu}_i$ with reported precision $\theta_{k,i,j}$. To deal with uncertainty about the user's trustworthiness, each user is assigned a latent trust value $t_k \in [0, 1]$, which models the accuracy of k in providing observations. In particular, values for t_k close to 0 mean that k is generally unreliable, and should be largely ignored; while values close to 1 mean that k is trustworthy, and in particular, precisions reported by k accurately reflect the reliability of its estimates. To capture this intuition, t_k is used to scale the precisions reported by k, such that the true precision of any given estimate, $\boldsymbol{x}_{k,i,j}$, is assumed to be $t_k \cdot \theta_{k,i,j}$. In particular, assuming Gaussian noise on each reported estimate, we obtain

the same likelihood of a *single* report described for MaxTrust in Equation 3.8 (with the only difference that the observations are now also indexed by i). That is:

$$p(\boldsymbol{x}_{k,i,j}|\boldsymbol{\mu}_i, \theta_{k,i,j}, t_k) = \mathcal{N}(\boldsymbol{x}_{k,i,j}|\boldsymbol{\mu}_{k,i,j}, (t_i\theta_{k,i,j}^{-1}\mathrm{I}_d)) \tag{4.1}$$

where I_d is the d-dimensional identity matrix. As per MaxTrust, this expression means that users are assumed to observe items with uncorrelated (diagonal) noise proportional to their reported precision scaled by t_k. However, since t_k is unknown, uncertainty about its value must be dealt with in some way. In MaxTrust, this is achieved by assigning a single maximum likelihood estimate to t_k (see Section 3.1.3) However, as discussed earlier, this does not properly account for the amount of uncertainty in t_k, and due to the nature of MaxTrust, must be estimated separately for each item.

To address these issues, we define $\boldsymbol{\Theta}_{k,i,j} = \theta_{k,i,j}\mathrm{I}_d$ as the multivariate precision reported by user k for the j–th estimate for item i, and let $\boldsymbol{x}$, $\boldsymbol{\theta}$, $\boldsymbol{t}$, $\boldsymbol{\mu}$ be vectors comprising all reported estimates, precisions, trust values and true item values respectively. Let us use $\boldsymbol{x}'$ to indicate the transpose vector of $\boldsymbol{x}$ and so forth for all the other variables. According to the Gaussian noise model in Equation 4.1, the joint likelihood of the reported estimates of *all* the items, $\boldsymbol{x}$, is thus the product of the Gaussian densities associated with each report [3], which can be written as follows:

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\theta}, \boldsymbol{t}) = \prod_{i=1}^{M}\prod_{k=1}^{K}\Bigg\{ \exp\Bigg(-\frac{t_k}{2}\Bigg\{\Bigg[\sum_{j=1}^{p_{k,i}} \boldsymbol{x}'_{k,i,j}\boldsymbol{\Theta}_{k,i,j}\boldsymbol{x}_{k,i,j}\Bigg] - 2\boldsymbol{\mu}'_i\Bigg[\sum_{j=1}^{p_{k,i}}\boldsymbol{\Theta}_{k,i,j}\boldsymbol{x}_{k,i,j}\Bigg] \\ + \boldsymbol{\mu}'_i\Bigg[\sum_{j=1}^{p_{k,i}}\boldsymbol{\Theta}_{k,i,j}\Bigg]\boldsymbol{\mu}_i\Bigg\}\Bigg) \times \prod_{j=1}^{p_{k,i}}\sqrt{\frac{|t_k\boldsymbol{\Theta}_{k,i,j}|}{(2\pi)^d}}\Bigg\} \end{aligned}$$

To simplify the notation, let $\boldsymbol{W}_{k,i} = \sum_{j=1}^{p_{k,i}}\boldsymbol{\Theta}_{k,i,j}$ and $\hat{\boldsymbol{x}}_{k,i} = \boldsymbol{W}_{k,i}^{-1}\sum_{j=1}^{p_{k,i}}\boldsymbol{\Theta}_{k,i,j}\boldsymbol{x}_{k,i,j}$ where $\boldsymbol{W}_{k,i}$ is an invertible matrix given by the sum of the reported positive definite precision matrices. In particular, it is interesting to notice that $\boldsymbol{W}_{k,i}$ and $\hat{\boldsymbol{x}}_{k,i}$ are equivalent to the covariance intersection (CI)'s precision matrix and mean vector obtained by the CI fusion of the reports of user k for a single item (see Section 2.3.2.1), even though we derived them through different steps. Thus, the likelihood can be expressed as:

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\theta}, \boldsymbol{t}) = \Bigg(\prod_{i=1}^{M}\prod_{k=1}^{K}\prod_{j=1}^{p_{k,i}}\sqrt{\frac{|t_k\boldsymbol{\Theta}_{k,j}|}{(2\pi)^d}}\Bigg)\exp\Bigg[\sum_{k=i}^{K}\Bigg\{-\frac{t_k}{2} + \sum_{i=1}^{M}\Bigg(\Bigg[\sum_{j=1}^{p_{k,i}}\boldsymbol{x}'_{k,j}\boldsymbol{\Theta}_{k,j}\boldsymbol{x}_{k,j}\Bigg] \\ - 2\boldsymbol{\mu}'\boldsymbol{W}_{k,i}\hat{\boldsymbol{x}}_{k,i} + \boldsymbol{\mu}_i\boldsymbol{W}_{k,i}\boldsymbol{\mu}'_i\Bigg)\Bigg\}\Bigg] \end{aligned}$$

[3] Recall that in our previous model the likelihood is taken as the expectation of a report over the fused estimate of a single item

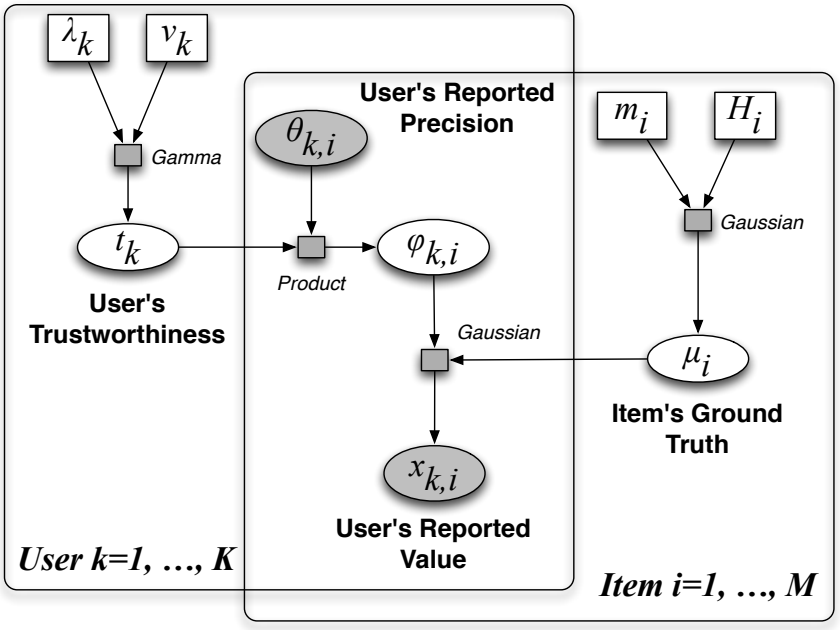


Figure 4.1: The factor graph of BACE, illustrating the probabilistic relationship between the variables of the model.

To derive the quadratic form of the likelihood, we complete the square of the exponential term[4] by adding and subtracting $\hat{\boldsymbol{x}}'_{k,i}\boldsymbol{W}_{k,i}\hat{\boldsymbol{x}}_{k,i}$:

$$p(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\theta},\boldsymbol{t}) = \left(\prod_{i=1}^{M}\prod_{k=1}^{K}\prod_{j=1}^{p_{k,i}}\sqrt{\frac{|t_k\boldsymbol{\Theta}_{k,i,j}|}{(2\pi)^d}}\right)\exp\left(\sum_{k=1}^{K}\left\{-\frac{t_k}{2}+\sum_{i=1}^{M}\left(\left[\sum_{j=1}^{p_{k,i}}\boldsymbol{x}'_{k,i,j}\boldsymbol{\Theta}_{k,i,j}\boldsymbol{x}_{k,i,j}\right]\right.\right.\right.$$
$$\left.\left.\left.-\hat{\boldsymbol{x}}'_{k,i}\boldsymbol{W}_{k,i}\hat{\boldsymbol{x}}_{k.i}+(\hat{\boldsymbol{x}}_{k,i}-\boldsymbol{\mu}_i)'\boldsymbol{W}_{k,i}(\hat{\boldsymbol{x}}_{k,i}-\boldsymbol{\mu}_i)\right)\right\}\right) \tag{4.2}$$

In this function, we have four terms inside the exponential whose sum models the fit of the latent variables $\boldsymbol{\mu}$ and $\boldsymbol{t}$ to the dataset. In particular, the first term is the negative semi–sum of the trust parameters, the second term is the linear sum of user' estimates multiplied by their precision, the third term is the CI linear fusion of the user's reports and the fourth term models the distance of the user's fused estimate from $\boldsymbol{\mu}_i$.

In more detail, the factor graph in Figure 4.1 illustrates the probabilistic relationships between the variables of the likelihood. In particular the observed variables $\boldsymbol{x}_{k,i}$ and $\theta_{k,i}$ appear as shaded nodes, while latent variables t_k and $\boldsymbol{\mu}_i$) are in unshaded nodes – the index j is omitted in this example for readability. In the graphical model, the variable $\varphi_{k,i}$ denotes the product of the reported precision scaled by trust. The rectangular boxes of the graph represent *plates* that are duplicated parts of the graphical model. Specifically, the graph has two plates that include the random variables associated with the various items and users. Specifically, the users' plate includes t_k, while the items'

[4]Additional details on Gaussian square completion can be found in Narasimhan (2008)

plate includes $\boldsymbol{\mu}_i$. Both $\theta_{k,i}$ and $\boldsymbol{x}_{k,i}$ are shared between the two plates as they define the association between the users and their observed items. The directed links denote the probabilistic factors connecting the variables. Specifically, t_k is connected to $\theta_{i,j}$ by a deterministic product factor, while $\boldsymbol{\mu}_i$ is connected to $\boldsymbol{x}_{k,i}$ and $t_k\theta_{k,i}$ by a Gaussian factor.

Having now described the data likelihood of BACE, the probabilistic inference of all the unknown variables is described in the following section.

4.2 A Monte Carlo Inference Process

We now describe the prerequisites to perform inference of the trust's parameters and the items' value using BACE. Like in MaxTrust, all the parameters $\boldsymbol{\mu}_i$ and t_k are unknown, therefore we must infer their likely values from the estimates reported by the users. To accomplish this though Bayesian inference, we assign conjugate prior distributions to each unknown parameter[5]. In particular, given the form of the likelihood, the conjugancy of our model is satisfied by setting the prior for each $\boldsymbol{\mu}_i$ to be Gaussian with multivariate mean $\boldsymbol{m}_i$ and precision matrix $\boldsymbol{H}_i$, and setting the prior for each t_k to be a Gamma distribution, with shape parameter λ_k and scale parameter ν_k. That is:

$$p(\boldsymbol{\mu}_i) = \mathcal{N}(\boldsymbol{\mu}_i|\boldsymbol{m}_i, \boldsymbol{H}_i) \qquad \forall i \tag{4.3}$$

$$p(t_k) = \text{Gamma}(t_k|\lambda_k, \nu_k) \qquad \forall k \tag{4.4}$$

In particular, the parameters of the priors (or *hyperparameters*) can be appropriately chosen to define some prior beliefs over these latent variables. For example, we may want to set a high prior trustworthiness of a particular user by selecting a high λ_k and a low ν_k. This feature is meaningful in many expert crowdsourcing settings where some users are known to be more reliable than others (Tran-Thanh et al., 2012).

With this in mind, we can apply Bayes theorem to derive the joint posterior distribution of $\boldsymbol{\mu}$ and $\boldsymbol{t}$ as proportional to the likelihood (Equation 4.2) multiplied by their priors (Equations 4.3 and 4.4). That is:

$$p(\boldsymbol{\mu}, \boldsymbol{t}|\boldsymbol{x}, \boldsymbol{\theta}) \propto p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\theta}, \boldsymbol{t}) \left[\prod_{i=1}^{M} p(\boldsymbol{\mu}_i)\right] \left[\prod_{k=1}^{K} p(t_k)\right] \tag{4.5}$$

This posterior is similar to a combination of independent Normal-Gamma models (DeGroot and Schervish, 2012) applied separately to each item. However, the introduction of trust parameters as defined in BACE means that the posterior no longer has a separable form. Although this coupling ultimately enables transfer learning, it also means that we

[5]In Bayesian analysis, conjugate priors are distributions specifically chosen so that the posterior distribution has the same form, which simplifies inference (Schlaifer and Raiffa, 1961)

cannot have an exact inference for $\boldsymbol{\mu}_i$ and t_k. To deal with this, we need to approximate the marginal posterior of each latent variable in some way. For this purpose, while a number of approximation techniques for Bayesian inference are available (Bishop, 2006), we choose a Monte Carlo method that provides an accurate and still tractable approach to approximate inference in Bayesian models (DeGroot and Schervish, 2012)[6]. In particular, we perform approximate inference in BACE using a particular Monte Carlo process called Gibbs sampling (Gelfand and Smith, 1990). This is a Markov chain Monte Carlo (MCMC) algorithm that computes the marginal distributions of the latent variables by obtaining samples from their conditional posterior distributions. After an initial burn-in phase, the chains converge to samples from the posterior distribution and the full marginal distributions of each variable can be estimated from the frequencies of the observed samples. To derive a Gibbs sampler for BACE, we require the conditional distributions of each latent variable which we derive by conditioning the joint posterior (Equation 4.5) on all the other variables. The next sub–section provides the detailed proofs of these derivations.

4.2.1 Conditional Distribution for the Items

Starting with analysing the probability density of the true value of each item i, $\boldsymbol{\mu}_i$, we condition the likelihood on some known value of $\boldsymbol{t}$. Then, dropping the constant terms that do not depend on $\boldsymbol{\mu}$, we have:

$$
\begin{aligned}
p(\boldsymbol{\mu}_i|\boldsymbol{t},\boldsymbol{x},\boldsymbol{\theta}) \propto\ & \exp\left(-\frac{1}{2}\left\{ \sum_{k=1}^{K} t_k(-2\boldsymbol{\mu}_i'\boldsymbol{W}_{k,i}\hat{\boldsymbol{y}}_{k,i} + \boldsymbol{\mu}_i'\boldsymbol{W}_{k,i}k\boldsymbol{\mu}_i) \right\} \right) \\
& \propto \exp\left(-\frac{1}{2}\left\{ -2\boldsymbol{\mu}_i' \left[\sum_{k=1}^{K} t_k \boldsymbol{W}_{k,i}\hat{\boldsymbol{y}}_{k,i} \right] + \boldsymbol{\mu}_i' \left[\sum_{k=1}^{K} t_k \boldsymbol{W}_{k,i} \right] \boldsymbol{\mu}_i \right\} \right)
\end{aligned}
$$

Let $\boldsymbol{S}_i = \sum_{k=1}^{K} t_k \boldsymbol{W}_{k,i}$ and $\bar{\boldsymbol{x}}_i = \boldsymbol{S}_i^{-1} \sum_{k=1}^{K} t_k \boldsymbol{W}_{k,i}\hat{\boldsymbol{x}}_{k,i}$ be weighted averages of the users' reported precision and estimates, respectively. Hence, if we complete the square of the exponential term, we have:

$$
\begin{aligned}
p(\boldsymbol{\mu}_i|\boldsymbol{t},\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{m}_i,\boldsymbol{H}_i) &\propto p(\boldsymbol{x}|\boldsymbol{\mu}_i,\boldsymbol{\theta},\boldsymbol{t}) p(\boldsymbol{\mu}_i|\boldsymbol{m}_i,\boldsymbol{H}_i) \\
&\propto \exp\left(-\frac{1}{2}\left[(\bar{\boldsymbol{y}}_i - \boldsymbol{\mu}_i)'\boldsymbol{S}_i(\bar{\boldsymbol{y}}_i - \boldsymbol{\mu}_i) + (\boldsymbol{\mu}_i - \boldsymbol{m}_i)'\boldsymbol{H}_i(\boldsymbol{\mu}_i - \boldsymbol{m}_i) \right] \right) \\
&\propto \exp\left(-\frac{1}{2}\left[\boldsymbol{\mu}_i'(\boldsymbol{S}_i + \boldsymbol{H}_i)\boldsymbol{\mu}_i - 2\boldsymbol{\mu}_i'(\boldsymbol{S}_i\bar{\boldsymbol{y}}_i + \boldsymbol{H}_i\boldsymbol{m}_i) \right] \right)
\end{aligned}
$$

[6] Alternative techniques that may offer performance advantages in the inference of BACE can be found in Bishop (2006)

From this, after completing the square of the exponential term, we find that the normalised conditional posterior of $\boldsymbol{\mu}_i$ is Gaussian distributed with probability density function (p.d.f.):

$$p(\boldsymbol{\mu}_i|\boldsymbol{t},\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{m}_i,\boldsymbol{H}_i) = \mathcal{N}(\boldsymbol{\mu}_i|\boldsymbol{b}_i,\boldsymbol{A}_i) \qquad \forall i \tag{4.6}$$

where $\boldsymbol{A}_i = \boldsymbol{S}_i + \boldsymbol{H}_i$ and mean $\boldsymbol{b}_i = \boldsymbol{A}_i^{-1}(\boldsymbol{S}_i\bar{\boldsymbol{x}}_i + \boldsymbol{H}_i\boldsymbol{m}_i)$. Notice that this function does not depend on any $\boldsymbol{\mu}_j$ for $j \neq i$, which implies that all $\boldsymbol{\mu}_j$ are mutually independent given $\boldsymbol{t}$. Moreover, since the posterior for $\boldsymbol{\mu}_i$ is based on the trust–based weighted average of user reports in both its Gaussian mean and precision, this means that the model encodes the same idea of our previous model (MaxTrust) based on the majority assumption, which we now derived through Bayesian analysis.

4.2.2 Conditional Distribution for the Trust Parameters

By a similar process, we now derive the conditional posterior of t_k given some known value of $\boldsymbol{\mu}$ and $\boldsymbol{t}_{-k}$, i.e., all the trust parameters but t_k. Hence, absorbing the terms of the likelihood that do not depend on t_k into a constant, we have:

$$\begin{aligned} p(t_k|\boldsymbol{\mu},\boldsymbol{t}_{-k},\boldsymbol{x},\boldsymbol{\theta}) &\propto \prod_{i=1}^{M}\prod_{j=1}^{p_{k,i}} \left\{ \sqrt{t_k} \exp\left(-\frac{1}{2}(\boldsymbol{x}_{k,i,j} - \boldsymbol{\mu}_i)' t_k \boldsymbol{\Theta}_{k,i,j} (\boldsymbol{x}_{k,i,j} - \boldsymbol{\mu}_i) \right) \right\} \\ &\propto t_k^{\frac{1}{2}\sum_{i=1}^{M} p_{k,i}} \exp\left(-\frac{1}{2} t_k \sum_{i=1}^{M}\sum_{j=1}^{p_k} \left\{ (\boldsymbol{x}_{k,i,j} - \boldsymbol{\mu}_i)' \boldsymbol{\Theta}_{k,i,j} (\boldsymbol{x}_{k,i,j} - \boldsymbol{\mu}_i) \right\} \right) \end{aligned}$$

Define $\alpha_k = \frac{1}{2}(\sum_{i=1}^{M} p_{k,i})$ (recall $p_{k,i}$ is the number of reports of user k for item i) and $\beta_k = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{p_{k,i}} (\boldsymbol{x}_{k,i,j} - \boldsymbol{\mu}_i)' \boldsymbol{\Theta}_{k,i,j} (\boldsymbol{x}_{k,i,j} - \boldsymbol{\mu}_i)$. Now we have:

$$\begin{aligned} p(t_k|\boldsymbol{\mu},\boldsymbol{t}_{-k},\boldsymbol{x},\boldsymbol{\theta},\lambda_k,\nu_k) &\propto p(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\theta},\boldsymbol{t}) p(t_k|\lambda_k,\nu_k) \\ &\propto t_k^{\left(\frac{1}{2}(\alpha_k+\lambda_k-1)\right)} \exp\left(-t_k(\beta_k + \nu_k) \right) \end{aligned}$$

From this we find that the conditional posterior of t_k is Gamma distributed with p.d.f:

$$p(t_k|\boldsymbol{\mu},\boldsymbol{x},\boldsymbol{\theta}) = \text{Gamma}(t_k|\alpha_k + \lambda_k, \beta_k + \nu_k) \quad \forall k \tag{4.7}$$

That is, the shape parameter $(\alpha_k + \lambda_k)$ of the Gamma posterior is proportional to the number of user reports, while the scale parameter $(\beta_k + \nu_k)$ is proportional to the distance of the user's reports from $\boldsymbol{\mu}_i$.

At this point, we have derived the key equations required for Bayesian inference with BACE. Now, since we wish to estimate the approximate marginal distributions of the items and the trust parameters using Gibbs sampling, we must iteratively sample from the conditional posteriors of these variables until we obtain a sufficient number of independent samples. This procedure is detailed in the next section.

Algorithm 4.1 BACE

Inputs:

Reports $\langle \boldsymbol{x}, \boldsymbol{\theta} \rangle$, hyperparameters $\lambda_k, \nu_k, \boldsymbol{m}, \boldsymbol{H}$;

Outputs:

Item's value samples $\boldsymbol{\mu}$, trust value samples $\boldsymbol{t}$

Algorithm *BACE*

randomly initialise $\boldsymbol{\mu}_i(0)$ and $t_k(0)$
for $s = 1$ **to** max_samples **do**
  for all item i **do**
    $\boldsymbol{W}_{k,i} = \sum_{j=1}^{p_{k,i}} \boldsymbol{\Theta}_{k,i,j} \qquad \forall k$
    $\hat{\boldsymbol{x}}_{k,i} = \boldsymbol{W}_{k,i}^{-1} \sum_{j=1}^{p_{k,i}} \boldsymbol{\Theta}_{k,i,j} \boldsymbol{x}_{k,i,j} \qquad \forall k$
    $\boldsymbol{S}_i = \sum_{k=1}^{K} t(s-1)_k \boldsymbol{W}_{k,i}$
    $\bar{\boldsymbol{x}}_i = \boldsymbol{S}_i^{-1} \sum_{k=1}^{K} t_k(s-1) \boldsymbol{W}_{k,i} \hat{\boldsymbol{x}}_{k,i}$
    $\boldsymbol{A}_i = \boldsymbol{S}_i + \boldsymbol{H}$
    $\boldsymbol{b}_i = \boldsymbol{A}_i^{-1}(\boldsymbol{S}_i \bar{\boldsymbol{y}}_i + \boldsymbol{H}\boldsymbol{m})$
    $\boldsymbol{\mu}_i(s) \sim \mathcal{N}(\boldsymbol{b}_i, \boldsymbol{A}_i)$
  end for
  for all users k **do**
    $\alpha_k = \frac{1}{2}(\sum_{i=1}^{M} p_{k,i})$
    $\beta_k = \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{p_{k,i}} (\boldsymbol{x}_{k,i,j} - \boldsymbol{\mu}_i(s))' \boldsymbol{\Theta}_{k,i,j} (\boldsymbol{x}_{q,i,j} - \boldsymbol{\mu}_i(s))$
    $t_k(s) \sim \text{Gamma}(\alpha_k + \lambda_k, \beta_k + \nu_k)$
  end for
end for
return $\boldsymbol{\mu}, \boldsymbol{t}$

4.3 The BACE Training Algorithm

In this section, we describe the algorithm for training BACE over a set of crowdsourced estimates and so infer the aggregated values of the observed items as well as the trustworthiness of the users. In detail, using the conditional densities of the latent variables previously derived in Equations 4.6 and 4.7, the Gibbs sampler for BACE can be described as follows (see Algorithm 4.1). Given the set of reported estimates and the initial hyperparameters of the prior distributions, first we define an starting value of $\boldsymbol{t}$ and $\boldsymbol{\mu}$, which can be set randomly. Then, the algorithm executes the item's sampling loop (steps 3-11) where, for each item i, it computes the fusion of the estimates using the current values of t_k (steps 7). After combing the fused estimate with the prior of $\boldsymbol{\mu}_i$ as by Equation 4.6 (steps 8-9) it draws a random sample of $\boldsymbol{\mu}_i$ from its Gaussian posterior p.d.f. with the updated parameters (step 10). Subsequently, the algorithm executes the user's sampling loop (steps 12-16). Specifically, for each user k, it draws a new random sample of t_k (step 15) from its Gamma posterior with parameters that are updated using the latest samples of μ_i. The algorithm iterates over these steps until the required number of samples is produced. The output is then the chain of samples of $\boldsymbol{\mu}_i$ and $t_k, \forall i, k$ that empirically describe the marginal distributions of each variable. In practice, this algorithm can generate tens of thousand of samples within minutes on a standard PC, that can approximate the marginal densities sufficiently accurately. Our experiments presented in the next section will provide more insights on a real application of this algorithm.

4.4 Experimental Evaluation

In this section, we present the results of the evaluation of BACE. Following the same methodology used in the evaluation of MaxTrust (see Section 3.3), we conduct a first experiment with synthetic data to test the correctness of the trust learning and the robustness of BACE against various levels untrustworthy crowds (Section 4.4.1). Subsequently, our second experiment will assess the efficacy of BACE in the real–world application of WiFi hotspots localisation from crowdsourced reports (Section 4.4.2), which is an equivalent application to cell–tower localisation presented in the previous chapter (see Section 3.3.3).

In our experiments, we compare the performance of BACE to MaxTrust and several other methods that were described in the previous chapters (see Section 3.3 for more details). Specifically, we consider the following four benchmarks: {CI, MaxTrust, RM, Optimal Fusion} that include representative methods from both the classes of trust–based fusion (MaxTrust and RM) and non–trust based fusion (CI) algorithms. In this comparison, we do not consider CU and LOF that were already showed by our previous experiments to be less efficient methods in our crowdsourcing setting (see Section 3.3.3).

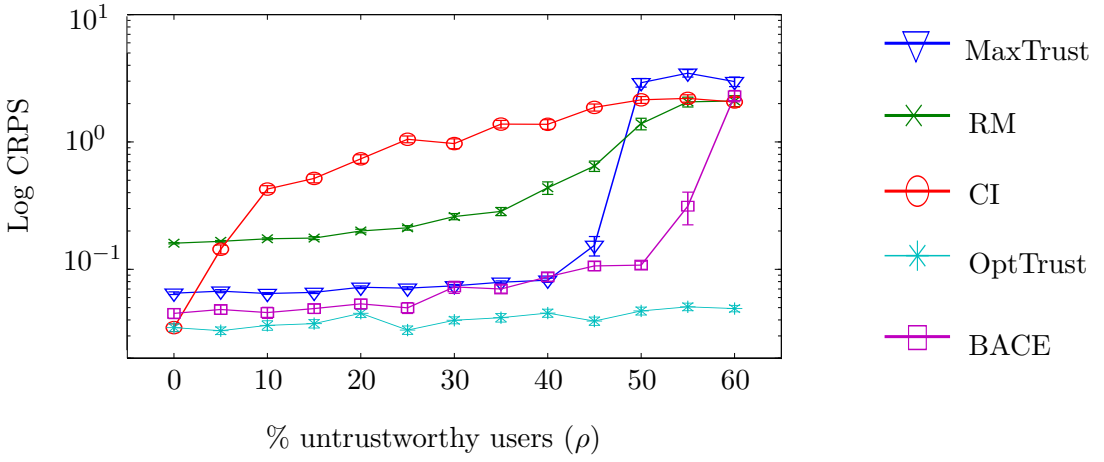
Specifically, we ran the BACE training algorithm with 1000 samples and burn–in phase of 100 samples. This number of samples was sufficient to guarantee the convergence of the MCMC chains all the random variables in our settings – the plots of these MCMC samples are provided in Section 4.4.2. The priors for both $\boldsymbol{\mu}$ and $\boldsymbol{t}$ were uninformative with hyperparameters $\boldsymbol{m}_i = 0, \boldsymbol{H}_i = diag(100)$ and $\lambda_k = 0, \nu_k = 0$. In particular, the use of uninformative priors for all the items' and trust variables is the most plausible setting for our case in which no available prior information about the users and the items. Nevertheless, these priors can be differently chosen to account for initial beliefs over users and items in more general settings. Moreover, to guarantee the interpretability of the t_k values, the Gamma posterior of t_k, which has support in $\mathbb{R}^+$, is truncated into the range $[0, 1]$ by applying rejection sampling within the training algorithm, i.e., rejecting samples of t_k that fall outside this range.

Furthermore, the accuracy of each method is measured by the two metrics described in the previous chapter (Section 3.3.1.2): the RMSE and the CRPS. In particular, for the BACE predictions, which are given in the form of sampled distributions, the CRPS is computed directly from the samples of the posterior of $\boldsymbol{\mu}$ via a discretised integral approximation (see the Appendix A for details on this CRPS approximation).

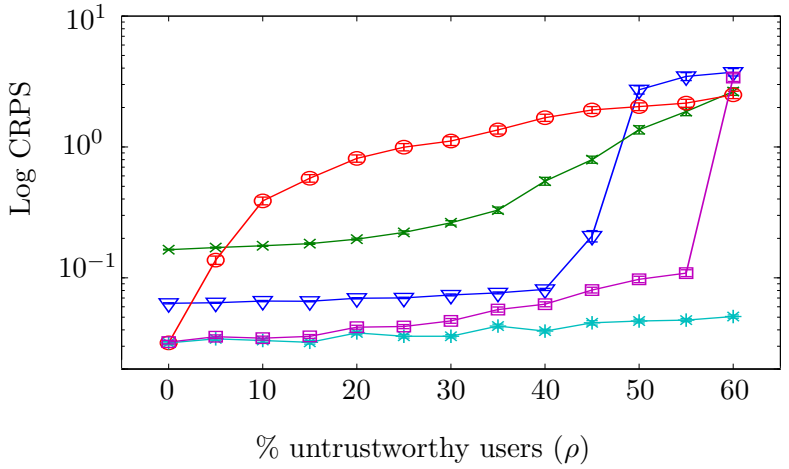
4.4.1 Experiments on Synthetic Data

In this experiment, we evaluate BACE on fusion tasks with synthetic univariate data. Same as before, we simulate a crowd of $K = 20$ users composed by $\rho\%$ untrustworthy users and $(1 - \rho)\%$ trustworthy users. For a given number of items M, we generate their true value as $\mu_i \sim U[0, 100]$, Each user reports $p_{k,i} \sim U[3, 10]$ estimates. Specifically, each estimate is randomly generated with values $\theta_{k,i,j} \sim U[0.2, 1.5]$ and $x_{k,i,j} \sim \mathcal{N}(\mu_i, \theta_{k,i,j})$. To simulate untrustworthy reports, we randomly pick $\rho\%$ of the users and add a random bias to their reports, i.e., $\hat{x}_{k,i,j} = x_{k,i,j} + w_{k,i,j}$ with $w_{k,i,j} \sim \pm U[2, 8]$ (keeping the sign of w fixed in a single run to avoid balancing effects between biases).

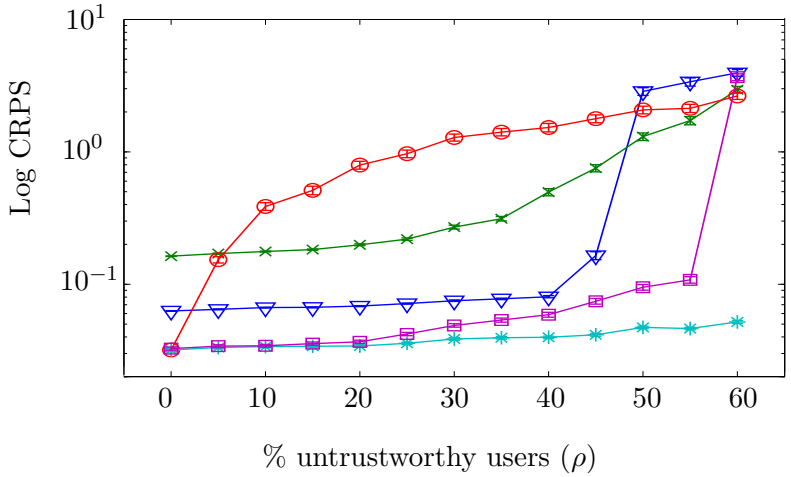
Figure 4.2 shows the plot of the CRPS of the algorithms averaged over 100 runs for $\rho = \{0, 10, 20, 30, 40, 50, 60\}$ with a small ($M = 1$, Figure 4.2(a)), moderate ($M = 1$, Figure 4.2(b)), and large ($M = 10$, Figure 4.2(c)) item set. In particular, the CRPS of CI, RM and MaxTrust are consistent with the results of our previous experiments with these three methods that rank as MaxTrust < RM < CI (the lower, the better) for $0\% < \rho < 40\%$. Importantly, the results also show that BACE improves on the accuracy of all these methods with gains that are up to 30% for one item, up to 45% for five items, and up 48% for ten items. For one item, this accuracy gain shows that Bayesian learning performed by BACE that account for the full uncertainty over the random variables in the fusion process, is more effective than the other algorithm. For five and ten items, the further improvement on accuracy gains are due to the efficacy of



(a) CRPS one item ($M = 1$)

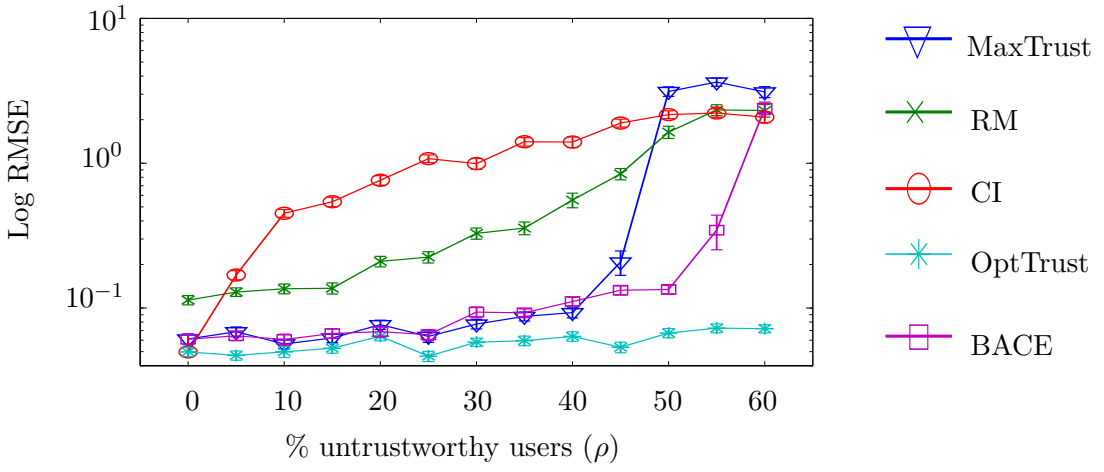


(b) CRPS five items ($M = 5$)

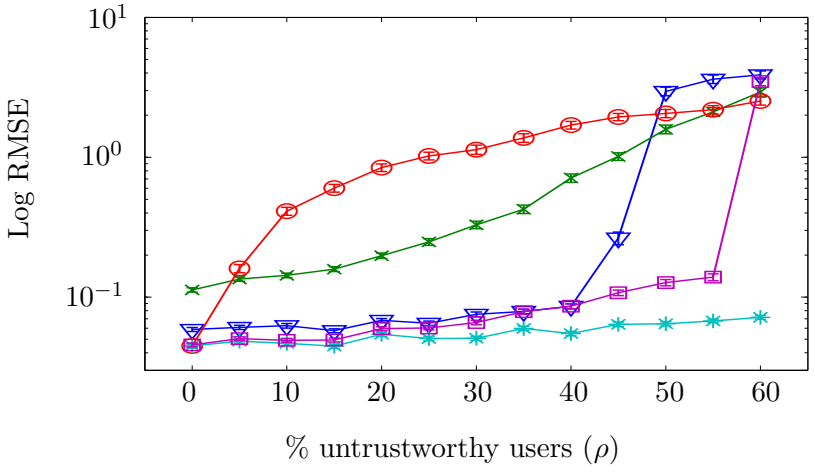


(c) CRPS ten items ($M = 10$)

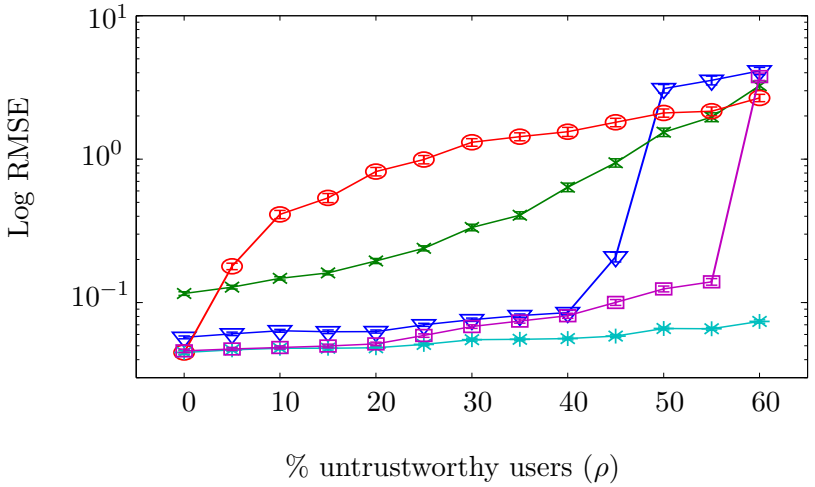
Figure 4.2: The CRPS of the five methods with increasing percentages of untrustworthy users and different numbers of items.



(a) RMSE one item ($M = 1$)



(b) RMSE five items ($M = 5$)



(c) RMSE ten items ($M = 10$)

Figure 4.3: The RMSE of the five methods with increasing percentages of untrustworthy users and different numbers of items.

the trust's transfer learning that allows BACE to improve accuracy with a larger set of items by leveraging correlations in the reliabilities of the reports. These improvements are more evident in the graph in the comparison between one and five items and it is marginally visible between five and ten items in our setting. In contrast, the accuracy of the other algorithms is invariant to the number of items as a consequence of treating items independently by these method. It is also interesting to compare the levels of ρ at which the error of each method becomes significantly higher than the optimum, i.e., $\log CRPS > 1$, which is due to the high presence of untrustworthy reports. In particular, we can see that BACE start to have a high error only at $\rho = 60\%$ while MaxTrust, the second best method, can tolerate untrustworthiness only up to $\rho = 45\%$. This means that BACE achieves comparable accuracy with 15% more presence of untrustworthy users, thus it is generally more robust to untrustworthy crowds.

Furthermore, Figure 4.3 shows the RMSE of the algorithms computed from the same set of runs. While generally showing a similar trend, the RMSE shows lower but significant improvements in the performance of BACE against an increasing dimension of the item set. In particular, the progression of its accuracy gains in terms of RMSE is up to 6%, ($M = 1$, Figure 4.3(a)), for one item, up to 22% for five items, ($M = 5$, Figure 4.3(b)), and up to 25% for ten items, ($M = 10$, Figure 4.3(c)). This means that BACE substantially reduces the uncertainty in the predictions and also marginally improves the absolute error of the fused estimates in our simulated crowdsourcing setting.

4.4.2 Experiments on Real Data

In this second experiment, we test the performance of BACE with a real–world application of crowdsourcing WiFi hotspots maps from crowdsourced WiFi location reports provided from mobile devices. This application is relevant to the domain of mobile WiFi network where several crowdsourcing approaches are currently being explored by a number of projects such as OpenSignal (`www.opensignal.com`), Fon (`www.fon.com`) and Devicescape (`www.devicescape.com`) to maintain up–to–date WiFi maps available for the users. Thus, we now show how BACE can be employed to improve the accuracy of these maps with its trust–based Bayesian fusion of the reports.

4.4.2.1 Dataset

In our test, we use a dataset of crowdsourced WiFi hotspot location data collected from Android phones by OpenSignal. This dataset included 3608 reports from 46 Android devices for 149 WiFi hotspots (avg. reports per device = 50.63). Specifically, each report provides (i) the SSID and BSSID of the detected WiFi hotspot (ii) the location (latitude and longitude) of the phone, (iii) the precision of the location fix (in meters) and (iiii) the OpenSignal app version installed by the phone. In total, the reports were

Device–as–user (CRPS)				
WiFi	CI	RM	MaxTrust	BACE
1	0.198	0.270	0.177	**0.063**
2	**0.138**	0.171	0.205	0.208
3	0.340	0.185	0.698	**0.063**
4	0.135	0.120	0.227	**0.061**
5	0.195	0.744	0.719	**0.052**
6	0.162	N/A	0.214	**0.046**
7	0.654	**0.416**	0.679	0.575
8	0.095	0.206	0.185	**0.044**
9	0.153	0.061	0.757	**0.035**
10	0.185	0.160	0.199	**0.104**
11	0.128	0.188	0.176	**0.037**
12	0.142	0.218	0.209	**0.079**
13	0.187	N/A	0.193	**0.105**

Table 4.1: The CRPS of the four methods (columns) in predicting the location of the 13 WiFi hotspots (rows) of the OpenSignal–WiFi dataset in the *device–as–user* setting. The best run of each row is highlighted in bold.

Device–as–user (RMSE)				
WiFi	CI	RM	MaxTrust	BACE
1	25.94	17.29	20.59	**14.73**
2	**622.5**	603.25	574.7	626.3
3	**18.18**	32.73	302.7	31.65
4	**4.915**	34.21	36.16	11.87
5	**92.76**	102.3	501.6	123.2
6	**42.41**	N/A	621.6	45.26
7	**25.97**	28.53	26.35	34.40
8	5.898	114.3	95.52	**5.660**
9	12.44	5.816	16.31	**1.593**
10	56.42	48.61	**42.97**	54.05
11	16.37	38.03	31.74	**7.505**
12	**24.09**	22.37	26.89	27.60
13	26.38	N/A	111.15	**25.85**

Table 4.2: The RMSE (in meters) of the four methods (columns) in predicting the location of the 13 WiFi hotspots (rows) of the OpenSignal dataset in the *device–as–user* setting. The best run of each row is highlighted in bold.

sent from devices with 35 unique app versions (avg. reports per app version = 104.55). Furthermore, we acquired the true location for 13 of the WiFi hotspots from the British Telecom (BT) WiFi network database (`www.btWiFi.com`).

With this data, we focus on the task of recovering the true location of the WiFi hotspot by merging the signal detections taken by users at different locations. As discussed for the case of the OpenSignal–Cell tower dataset (see Section 3.3.3), the reports may have several sources of inaccuracy due, for instance, to the noise of the GPS readings or to the limited accuracy of the devices. However, a key difference between the cell–tower dataset and the WiFi dataset is represented by the density of the reports. In fact, the detection range of a WiFi hotspot is approximately 100 meters, which is two orders of magnitude lower than a cell tower range that can cover up to 10 km. Thus, in the WiFi dataset, we have more clustered set of reports which makes the user's trust assessment more challenging within a single item. Given this, we now apply BACE to estimate the position of the WiFi hotspots as well as the trustworthiness of the devices. In particular, in our setting, we map each device to a single user and each WiFi hotspot to a single item. For each report, we set $\boldsymbol{\mu}_{k,i,j}$ to be equal to the device's GPS location and $\theta_{k,i,j}$ to be equal to the reported GPS precision added by 100 meters, which is the default maximum WiFi range.

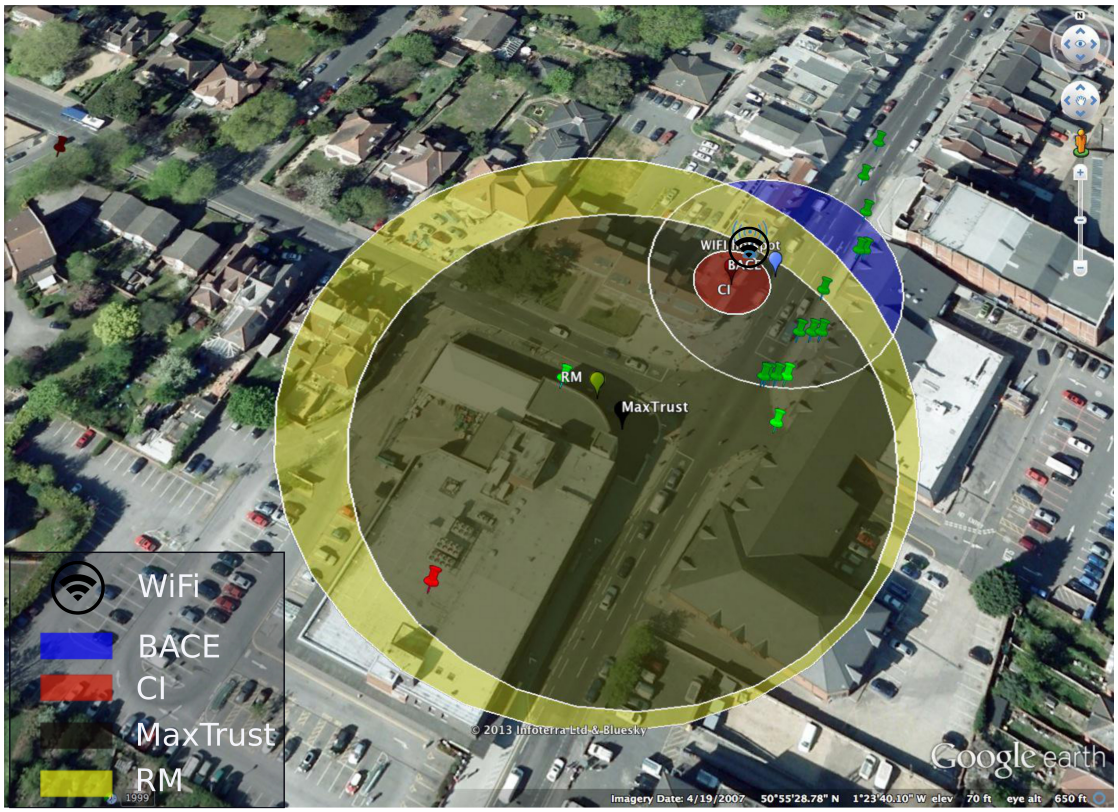


Figure 4.4: Example of the predicted WiFi hotspot location produced by the four methods for one of the 13 WiFi hotspot of the OpenSignal–WiFi dataset in the *device-as-user* setting.

4.4.2.2 Results

Table 4.2 (left table) reports the CRPS of the algorithms for each of the 13 WiFi hotspots with ground truth[7]. In particular, the results show that BACE is the best predictor in 11 out of 13 test cases and, on average, it outperforms the other methods by 45% (0.113 vs. 0.207) with a lower CRPS. In addition, Table 4.1 (right table) reports the RMSE of the algorithms measured in the same set of runs. The RMSE shows a comparable error between BACE and CI. Overall, this means the main advantage of BACE is a more accurate prediction of uncertainty in the fused estimate compared to the other methods. In general, these results confirm the finding of our previous experiment by showing that BACE is able to predict the uncertainty of its fusion more accurately, as a result of using its transfer learning framework. To illustrate this, Figure 4.4 shows the fused estimates computed by each algorithm for one sample WiFi hotspot of the OpenSignal–WiFi dataset (corresponding to the 8-th entry in Table 4.2 and Table 4.1). In particular, by looking at the individual predictions of the WiFi hotspot location, which are depicted as a circular area with a radius of 3 predictive standard deviations around the mean, we

[7]The two missing values of RM are due to the fact that the algorithm did not converge in those test cases with our parameters setting.

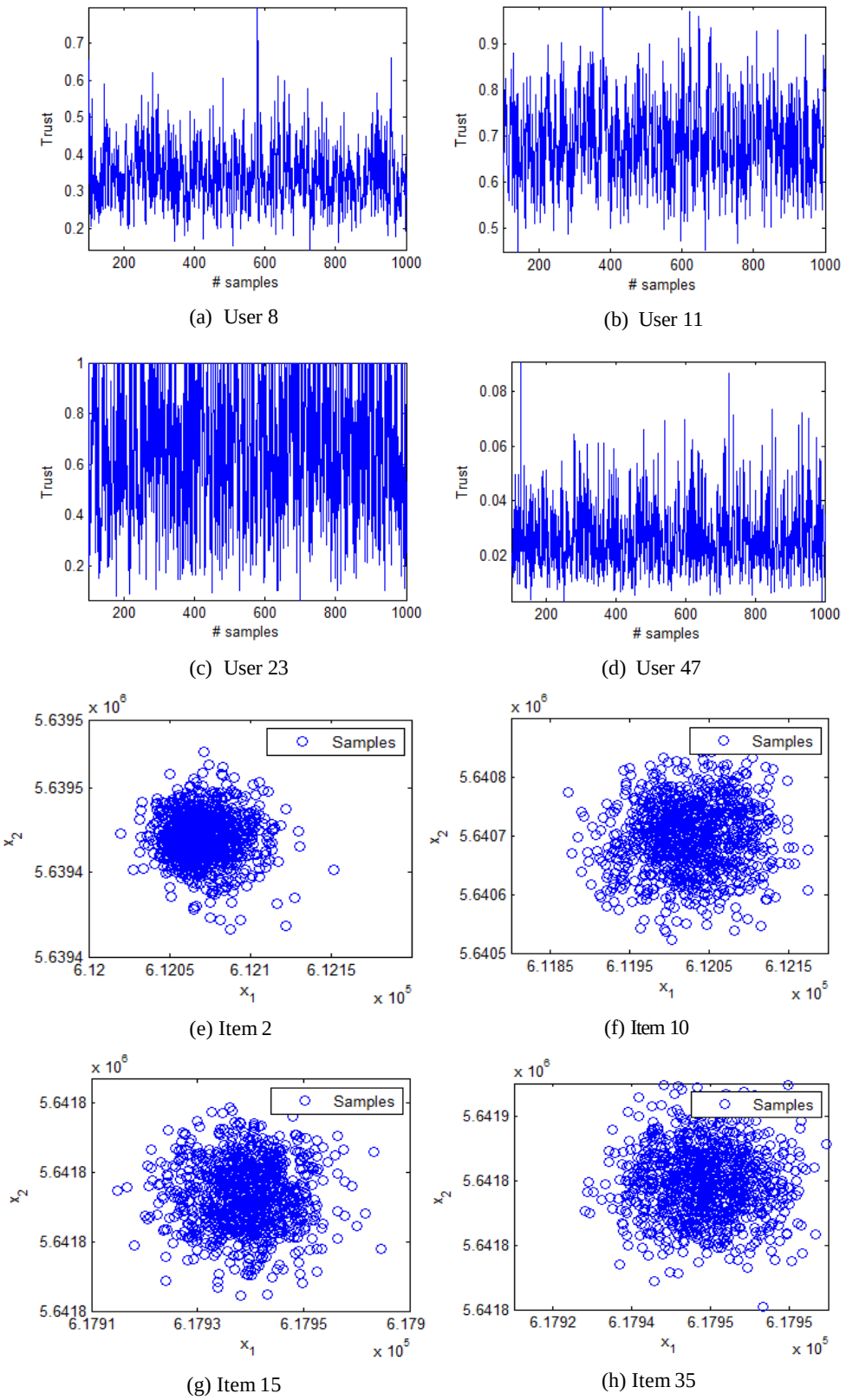
can see that BACE's estimate is the most consistent with the ground truth's location of the WiFi hotspot. In fact, CI predicts the mean value close to the true location (hence the lower RMSE) but it estimates the uncertainty overconfidently (hence the higher CRPS), thus providing an inconsistent estimate.

To monitor the convergence of the sampling–based approximate Bayesian inference of BACE, Figure 4.5 shows the MCMC chains of samples generated from the BACE's training algorithm for a sub–set of four users (Figure 4.5(a), (b), (c), (b)) and four items (Figure 4.5 (e), (f), (g), (h)). For the user's trustworthiness, the MCMC chain reaches convergence fast, already after the first 100 burn–in samples. For the items, it can be seen that the samples are distributed around a Gaussian cluster that corresponds to the posterior p.d.f.

Another interesting result is the trust values of individual devices estimated from BACE. These are shown in Figure 4.6(a). Analysing these values, although one might expect that the cheaper devices would have lower trustworthiness, we did not find a significant correlation between the device's estimated trustworthiness and its market price. In contrast, we found a significant correlation between these trust values and the average errors of the devices. In fact, using a standard statistical test of the Spearman rank correlation (Myers et al., 2010), we found that the rank correlation coefficient between the trust values and the devices' errors is $\rho = -0.527$ ($p = 0.0047$). This indicates a significant inverse correlation between the two ranks. That is, the devices with estimated low (high) trustworthiness are mostly far from (close to) the item's location, which is indeed the hypothesis of a correct trust leaning formulated by BACE.

In a second test, we repeated the same experiment taking app versions as users, i.e., treating all the users with the same app version as having the same trust value. By doing so, we are able to test the impact of a particular app release on the trustworthiness of the reports and the quality of the fusion. This shows the usefulness of BACE for debugging a particular app release by being able to reveal whether it produces more untrustworthy reports. In particular, Table 4.3 (left table) reports the CRPS of our experiments in this *app version–as–user* setting. The results show a generally lower accuracy for all the methods compared to the previous *devise-as-user* setting. However, also in this setting, BACE is the best predictor in 9 out of 13 (69%) the test cases. On average, it outperforms by 26% (0.148 vs. 0.201) the accuracy of the other methods. In contrast, the RMSE shows that CI is the best predictor in 8 out of 13 test cases in this setting (Table 4.4, right table).

Also, the trust values estimated by BACE for each app version (Figure 4.6) show that there is a majority of 62% app versions that are identified as more trustworthy with respect to the others. Furthermore, applying the same rank correlation test between the estimated app version's trust values and the app version's average errors of their reports, we obtained a Spearman's rank correlation coefficient $\rho = -0.314$ ($p = 0.1648$)



(a) User 8 (b) User 11

(c) User 23 (d) User 47

(e) Item 2 (f) Item 10

(g) Item 15 (h) Item 35

Figure 4.5: The chains of MCMC samples for the trust values (a, b, c, d) and the item values (e, f, g, h) generated from the BACE training algorithm in the *device-as-user* setting.

App version–as–user (CRPS)				
WiFi	CI	RM	MaxTrust	BACE
1	0.138	0.345	0.295	**0.092**
2	**0.159**	0.172	0.179	0.179
3	0.359	0.261	0.617	**0.152**
4	0.127	0.164	0.165	**0.095**
5	**0.154**	0.200	0.212	0.702
6	0.145	N/A	0.213	**0.067**
7	0.642	0.304	0.683	**0.202**
8	0.092	0.214	0.187	**0.039**
9	0.1627	0.684	0.716	**0.050**
10	0.201	**0.141**	0.195	0.166
11	0.150	0.046	0.175	**0.042**
12	0.168	**0.051**	0.127	0.061
13	0.122	0.191	0.179	**0.082**

Table 4.3: The CRPS of the four methods (columns) in predicting the location of the 13 WiFi hotspots (rows) of the OpenSignal–WiFi dataset in the *app version–as–user* setting. The best run of each row is highlighted in bold.

App version–as–user (RMSE)				
WiFi	CI	RM	MaxTrust	BACE
1	25.94	**19.34**	21.18	19.38
2	**622.5**	721.5	714.94	637.9
3	**18.18**	30.66	133.76	36.82
4	**4.915**	27.22	25.89	14.06
5	**92.76**	965.05	953.6	435.9
6	42.41	N/A	66.82	**38.48**
7	25.97	28.56	27.87	**25.69**
8	5.898	114.3	99.72	**5.210**
9	12.44	72.28	62.05	**11.79**
10	**56.42**	**56.42**	**56.42**	57.05
11	**16.37**	**16.37**	**16.37**	16.48
12	**24.09**	24.09	26.89	24.89
13	**26.38**	135.3	121.9	28.02

Table 4.4: The RMSE (in meters) of the four methods (columns) in predicting the location of the 13 WiFi hotspots (rows) of the OpenSignal dataset in the *app version–as–user* setting. The best run of each row is highlighted in bold.

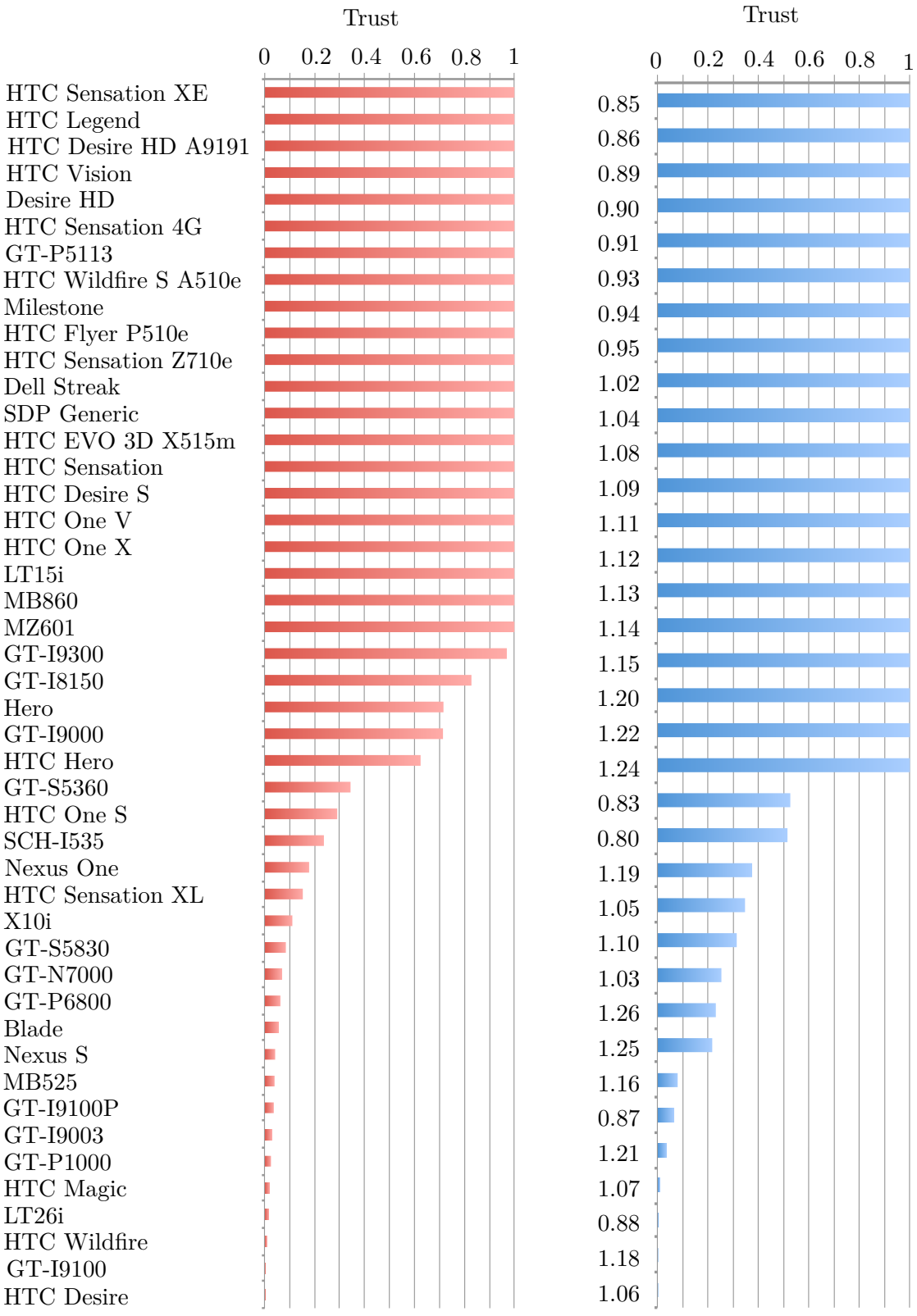
which confirms the same, although less significant, inverse correlation between the two ranks. Therefore, our results show that BACE improves the quality of the fused estimate, mostly in terms a lower predictive uncertainty, and produces a correct learning of trust values in both the evaluated settings.

4.5 Summary

In this chapter, we improved our solution to the problem related to the inference of reliable fused outputs from untrustworthy estimates of continuous quantities in crowdsourcing applications. This problem requires the joint learning of the individual trustworthiness of users and the computation of consistent fusion of the data in settings where crowd observations are pairs of values and precisions and items' values are defined in continuous spaces. Building upon our previous model (MaxTrust), we presented a new Bayesian model that improves the qualities of MaxTrust in a multi–item crowdsourcing setting by modelling correlations between the user's reports for different items. In fact, the key innovation of BACE is a fully Bayesian treatment of a probabilistic model in which the trust parameters are shared across items integrated in multivariate Gaussian framework. Then, using a Monte Carlo sampling process, we are able to learn approximate posterior probabilities of (i) the trust value of each user and (ii) the true value of each item. In particular, the use of Bayesian hierarchical modelling allows BACE to achieve the key feature of transfer learning, whereby user trust knowledge learned from observations of previous items is used as evidence for new items. By doing so, BACE is

able to improve the accuracy of the fusion by a more efficient handling of uncertainty in the inference of the items' true value. We ran several experiments on WIFI hotspots localisation data crowdsourced from Android devices and compared the performance of BACE against MaxTrust and other state-of-the-art fusion methods. We showed that BACE is 45% more accurate in estimating the WIFI hotspots locations and it correctly learns user trust values that are correlated to the true user's errors. We also showed that BACE achieves the same accuracy compared to the benchmarks with 15% more untrustworthy users through experiments on synthetic data.

From this we conclude that MaxTrust is our algorithm of choice for crowdsourcing a single item, as it allows for a more robust trust–based data fusion. Alternatively, BACE is more indicated for multi–items settings where its Bayesian transfer learning mechanism is able to exploit correlations between items to provide a more efficient estimation of uncertainty in the fused estimate. With these two algorithms, we provided a strong set of solutions which address our first two requirements related to fusing crowdsourced continuous estimates of stationary quantities. In the next two chapters, we will focus on our second set of requirements related to the fusion of spatial data for non–stationary quantities. In detail, we will discuss different instantiations of our trust–based fusion approach in the context of crowdsourcing spatial fields in Chapter 5 and crowdsourcing spatial point processes in Chapter 6.



(a) Devices (b) App versions

Figure 4.6: Estimated mean trust levels for (a) each device and (b) each app version learned from BACE on the OpenSignal–WiFi dataset.

Chapter 5

A Trust-based Heteroskedastic Gaussian Process Model for Fusing Crowdsourced Estimates of Spatial Functions

In the previous two chapters, we developed a set of accurate and efficient algorithms for fusing crowdsourced estimates for a single or multiple stationary items. By so doing, we provided an important set of solutions that fulfil our first set of requirements (Req. 1 and Req. 2) related to the design of a reliable crowd–based information system (see Section 1.2). However, the algorithms developed so far strongly rely on the assumption that all the reported estimates are referring to a fixed continuous quantity, which in our application examples were referring to the true position of a cell tower (Chapter 3) of a WiFi hotspot (Chapter 4). In fact, the data aggregation approaches of our algorithm consist of comparing the observations of a user to the crowd consensus given by the majority of the trustworthy users. As a result, they are not suitable to be applied in settings where the crowd reported data relate to non–stationary quantities. In such settings, their approach fails to consider the fact that each observation is not directly comparable to one another due to their dependency on the user's location. For this reason, they do not satisfy our second set of key requirements that relate to fusing estimates of non–stationary items (Req. 3 and Req. 4).

In this and the next chapter, we turn to the challenge of fusing crowdsourced estimates of non–stationary quantities. As discussed in Chapter 1, this challenge is of interest to the class of participatory sensing applications that rely on crowdsourcing spatial and spatial–temporal information[1], such as tracking contagious diseases (Sadilek et al.,

[1]Recall that GP regression over spatial and spatial–temporal data can be addressed by choosing specific types of separable space-time covariance functions within the standard GP framework (Osborne, 2010).

2012), monitoring traffic flows (Horvitz et al., 2012) and monitoring nuclear radioactivity in disaster response (Gertz and Di Justo, 2012). However, similar issues of data trustworthiness that we discussed in the context for the stationary data (see Section 1.2) represent one of the main obstacles to make the best use of such information in spatial crowdsourcing as well. In fact, the range of users' reliability that affects the data trustworthiness makes the task of aggregating crowdsourced spatial data into a single function difficult to achieve in practice. As a result, the computation of reliable aggregations of possibly untrustworthy spatial estimates is a key challenge in these crowdsourcing domains (Gao et al., 2011).

In this chapter, we address the first sub–problem related to the fusion of crowdsourced spatial data. Specifically, we focus on the fusion of continuous estimates of a spatial function (Req. 3) from the observations reported by possibly untrustworthy users (Req. 2)[2]. In particular, this setting is relevant to several application areas of crowdsourced environmental monitoring including sensing weather events, traffic monitoring and nuclear radiation monitoring. The key feature of this problem is that observations reported by the crowd may not only be untrustworthy but they may also be spatially correlated; a feature that is not handled by our previous models. In fact, in spatial crowdsourcing settings, the reports are typically pairs of values including a location and the measurements taken at that location, often including the precision of such a measurement. All together, this information represents a set of location–based observations of a spatial phenomenon and the goal of the task requestor is to estimate the entire function over the monitored area. To address this problem with our trust–based fusion approach, we need to extend our framework to be able to handle spatial correlation within the dataset. In doing so, we must be aware of the fact that typically such spatial correlations make the inference of the aggregated function more challenging. In fact, the inference space has higher complexity and the aggregated output has to be estimated as a continuous function, Moreover, the crowd observations need to be considered as related to a specific location, which makes them not directly comparable to one another. As a result, care must be taken in considering the individual trustworthiness of the reports as related to their level of the agreement with the underlying function. As discussed in Section 2.5, a natural way to deal with this problem is by means of regression models. However, the current spatial regression techniques, which are not designed for a crowdsourcing context, can only deal with data with constant–variance (i.e., homoskedastic) noise. That is, they work in a setting where the observations are affected by a general underlying noise, which reflects the possible perturbations of the reporting process as a whole. However, this idea is far from the concept of heterogeneous data reliabilities of a crowdsourcing context in which the data are corrupted by different noise sources associated with the varying trustworthiness of the users.

[2]The second class of models for spatial crowdsourcing applications will be discussed in Chapter 6.

To address these limitations, we propose a new method that extends our trust–based fusion approach to spatial regression. Specifically, our method is based on the integration of the user trust model that we defined in Section 3.1.1 within the heteroskedastic Gaussian process (HGP) framework. As we discussed in Section 2.5.3, the HGP is our model of choice for a spatial regression model as it provides a powerful non–parametric framework for Bayesian spatial regression. In particular, it is able to model non–linear spatial phenomena with a tractable Bayesian framework without necessarily requiring a specific knowledge of the physical model of the system. These qualities make such a model attractive to be employed for merging spatial data in crowdsourcing settings. Applying our user trust model, our new trust–based HGP (which we name TrustHGP) tackles the problem of dealing with heterogeneous data reliabilities in spatial regression by using a set of trust parameters for the users to scale the data noise rates of the HGP. In this way, the model has the ability to flexibly increase the noise around subsets of reports associated with untrustworthy users. Then, by training the model with the spatial estimates gathered from the crowd, we are able to estimate the underlying function at any location of interest and also learn the individual user's trustworthiness. We show that our method is more accurate than other standard GP and HGP approaches with an extensive experimental evaluation on both synthetic and real-word data. In particular, we show the robustness of our model against various levels of untrustworthy crowds using synthetic data. Then, we use real–world radiation data collected during the 2011 Fukushima earthquake to show the efficacy of the TrustHGP in an important disaster response application of crowdsourced radiation monitoring.

Thus, the contributions of this chapter are as follows:

- We propose a new trust-based HGP model that extends the HGP through a probabilistic user trust model to be able to aggregate spatial observations while learning the trustworthiness of the users from crowdsourced spatial estimates.
- We show that our method significantly improves the quality of the predictions of other GP and HGP methods in an application of crowdsourced radiation monitoring using real-world data from the 2011 Fukushima nuclear disaster. In particular, we show that our method outperforms the state of the art by up to 23% in providing more accurate radioactivity predictions. We also provide an in-depth analysis of the performance of our method using synthetic data. In particular, we show that our method provides comparable results with up to 30% more untrustworthy users.

The remainder of this chapter is structured as follows. Section 5.1 describes the TrustHGP model. Then, section 5.2 details the algorithm to compute the probabilistic predictions of the spatial function and the trustworthiness of each user from spatial observations. Section 5.3 presents an empirical evaluation of the TrustHGP using both synthetic data

and real-world data. Finally, Section 5.4 summarises our results in the context of the requirements of this thesis.

5.1 Model Description

In this section, we introduce our TrustHGP model by first describing its user trust model for spatial crowd reporting (Section 5.1.1). Then, we detail our HGP model to incorporate such a trust model in Bayesian spatial regression (Section 5.1.2).

5.1.1 A User Trust Model for Crowdsourced Spatial Estimates

In our model, we assume that there is a crowd of K users observing an environmental phenomenon represented by the function $f : \mathbf{R}^d \rightarrow \mathbf{R}$. Without loss of generality, let us assume a two-dimensional case (i.e., $d = 2$) to conform with the setting of spatial crowdsourcing with location data as inputs. Recalling our running examples given in Chapter 1, we assume that f may represent the crowdsourced radiation levels of a nuclear cloud, as in the scenario of in the aftermath of the 2011 Japan earthquake, or the incident levels of the waterborne disease that spread across the Haiti population as a consequence of a ground water contamination after the 2010 earthquake (Farmer, 2012). In these examples, the domain of f is the range of locations describing the land area where the phenomenon is monitored, while the codomain of f is the range of values that such a phenomenon can assume. Given this, we typically have $f : \mathbf{R}^2 \rightarrow \mathbf{R}_+$ meaning that f takes only two–dimensional locations as inputs and that negative values are excluded from the function outputs that refer to nuclear radiation levels or the water contamination levels.

Thus, we assume that each user k reports p_k estimates of f at different locations[3]. Each reported estimate $\boldsymbol{e}_{k,j}$ provides (i) a location $\boldsymbol{x}_{k,j} \in \mathbf{R}^2$, i.e., the position of the user (assumed to be also the position of the measurement), (ii) the output $y_{k,j} \in \mathbf{R}$, i.e., the value measured at $\boldsymbol{x}_{k,j}$ and (iii) the precision $\theta_{k,j} \in \mathbf{R}_+$ i.e., the uncertainty around $y_{k,j}$. In particular, as we detailed in Section 3.1.1, $\theta_{k,j}$ may be referring, for example, to the precision of a sensor (which is automatically provided by the GPS for location data), or the user's confidence level, or the variance of some repeated measurements. Also, in cases where a user is unable to report its precision, it is still possible to set $\theta_{k,j}$ to a non–negative default value while still preserving the properties of our model. Summing up, we have a report set $\boldsymbol{R} = \{\langle \boldsymbol{x}_{k,j}, y_{k,j}, \theta_{k,j} \rangle | k = 1 \dots K, j = 1 \dots p_k\}$ that consists of $q = \sum_{k=1}^{K} p_k$ estimates; $\boldsymbol{x} = \{\boldsymbol{x}_{k,j} | k = 1, \dots K, j = 1, \dots p_k\}$ is the vector of the inputs, $\boldsymbol{y} = \{y_{k,j} | k = 1, \dots K, j = 1, \dots p_k\}$ is the vector of the outputs and $\boldsymbol{\theta} = \{\theta_{k,j} | k = 1, \dots K, j = 1, \dots p_k\}$ is the vector of the precisions.

[3]Notice that the index i, which was used for indexing items in our previous models, is now omitted from the variables since the estimates no longer refer to a specific item.

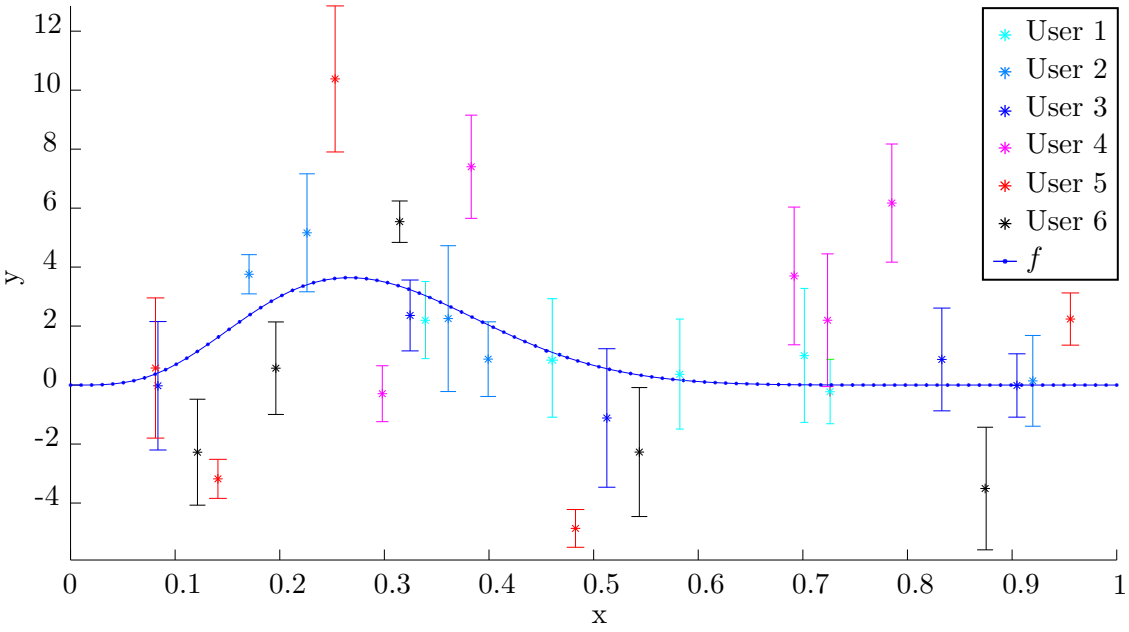


Figure 5.1: Example of trustworthy (users $1, 2, 3$) and untrustworthy (users $4, 5, 6$) reporting behaviour in our spatial regression model.

As per our previous models (MaxTrust and BACE), we capture the uncertainty related to the user's reported precisions and the unknown user's trustworthiness using the same modelling approach introduced in Section 3.1.1. Specifically, we relate the noise of an estimate to the reported precision by assuming that $\theta_{k,j}$ is the precision of the Gaussian noise corrupting $y_{k,j}$. Furthermore, we define a set of trust parameters $t_k \in (0, 1]$ to denote the reliability of a user (1 for a fully trustworthy user and approximately 0 for an untrustworthy user); $\boldsymbol{t} = (t_1, \ldots, t_K)$ is the vector of such parameters. Each t_k is used as a scaling parameter of the precisions reported by user k. By doing so, in our spatial setting, the user's trustworthiness is defined by the level of how the user's estimates are consistent with the true values of f. That is, a trustworthy user is expected to report observations that are sampled from f with a random noise. In contrast, untrustworthy users typically reports observations that are uncorrelated with f, as we discuss with an example below.

With this in mind, our trust–based model of spatial estimates can be formally described as follows. Let $\tilde{y}_{k,j}$ be actual value of f at $\boldsymbol{x}_{k,j}$, i.e. $\tilde{y}_{k,j} = f(\boldsymbol{x}_{k,j})$. Then, we consider that $y_{k,j}$ is a noisy measurement of $\tilde{y}_{k,j}$ with an additive zero-mean, Gaussian noise $\epsilon_{k,j}$ with precision $t_k \cdot \theta_{k,j}$. That is:

$$y_{k,j} = \tilde{y}_{k,j} + \epsilon_{k,j}, \qquad \tilde{y}_{k,j} = f(\boldsymbol{x}_{k,j}), \qquad \epsilon_{k,j} \sim \mathcal{N}(0, (t_k \theta_{k,j})^{-1}) \tag{5.1}$$

In this way, the model reproduces the same noise scaling effect of an untrustworthy estimate described in MaxTrust (see Section 3.1.1) which we have now extended to spatial observations. As a result of this formulation, the information of an untrustworthy estimate is downgraded by increasing its uncertainty proportionally to t_k.

In more detail, Figure 5.1 shows an example of six users with different levels of trustworthiness reporting observations of a one-dimensional function represented by Beta distribution with parameters $\alpha = 6, \beta = 18$ (blue-dotted line). Specifically, in this example, each user reports 5 estimates which are placed along x. Each estimate is plotted as its mean value $y_{k,j}$ (starred point) and the 95% reported confidence interval given by $\pm 2/\theta_{i,j}$ plotted as error bars. The user's trustworthiness can be inferred from the characteristics of their estimates relating to the consistency with the function. Specifically, we can see that user 1 and user 3 are highly trustworthy since all their estimates are consistent with the true value of the function. Furthermore, user 2 is mostly trustworthy since it has only one (the left-most) estimate that is inconsistent with $f(\boldsymbol{x}_{k,j})$. In contrast, users 4 and 6 are highly untrustworthy since all of their estimate are significantly far (i.e., more than 2 standard deviations away) from $f(\boldsymbol{x}_{k,j})$. Finally, user 6 is mostly untrustworthy since it has only one of its five estimates that is consistent with $f(\boldsymbol{x}_{k,j})$. This example provides a high–level idea of how we can capture the level of trustworthiness of each user, i.e., t_k, based on the level of consistency of its estimates with the true function. However, the challenge here is how to find the values of t_k that best explains the users' trustworthiness without observing the actual values of f. Thus, we detail how we address this problem through the design of a heteroskedastic Gaussian process model in the following section.

5.1.2 A Trust-Based Heteroskedastic Gaussian Process Model

In our model, we wish to perform probabilistic inference over f and $\boldsymbol{t}$ using the Gaussian process approach described in Section 2.5.3. To do so, we place a zero-mean GP prior over f, i.e. $m(\boldsymbol{x}) = 0$ with a kernel $K(\boldsymbol{x}, \boldsymbol{x}')$:

$$f(x) \sim \mathcal{GP}(0, K(\boldsymbol{x}, \boldsymbol{x}')) \tag{5.2}$$

Although any GP kernel can be used depending on the specific applications, here we use the squared–exponential covariance function which is commonly used as a kernel for modelling smoothly varying quantities:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f \exp\Big(- \frac{d(\boldsymbol{x}, \boldsymbol{x}')^2}{2l^2} \Big) \tag{5.3}$$

where σ_f is the signal variance, l is the length scale and $d(\cdot, \cdot)$ is the distance between the two inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$. To compute such a distance, since we primarily deal with location data as inputs, we use the standard equilateral projection:

$$d(\boldsymbol{x}, \boldsymbol{x}') = R_0 \sqrt{x^2 + y^2} \tag{5.4}$$

$$x = (lon - lon') \cos((lat + lat')/2) \tag{5.5}$$

$$y = lat - lat' \tag{5.6}$$

where $R_0 = 6,371km$ is the mean Earth's radius. In fact, this projection is computationally more efficient than the exact Haversine distance (Sinnott, 1984) (which provides the proper trigonometric treatment for spherical distances) although it introduces an error by approximating great–circle distances as triangular distances. However, its error is significant only for large distances, therefore it only marginally affects the correlations between the variables. In particular, for the case of squared–exponential covariance functions, which have non–zero values only for small distances, its error becomes even more negligible in spite of its computational advantages.

Next, as we discussed in Section 2.5.3, we need to assume mutual independence between the noise terms, i.e. $\epsilon_{k,j} \perp \epsilon_{k',j'}$ in order to have a tractable likelihood (Goldberg et al., 1997). This assumption implies that the $\theta_{k,j}$ and t_k parameters are also independent, i.e. $\theta_{k,j} \perp \theta_{k',j'}$ and $t_k \perp t_{k'}$, which is equivalent to assuming uncorrelated precisions between individual measurements and that users are independently trustworthy. Thus, under the HGP model, the likelihood of $\boldsymbol{y}$ is a normal p.d.f. expressed as follows:

$$p(\boldsymbol{y}|f) = \mathcal{N}(\boldsymbol{y}|f, \epsilon_{k,j}) \tag{5.7}$$

Now, let $\boldsymbol{x}_*$ be a 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 p.d.f. that can be written in a matrix form as follows:

$$\begin{bmatrix} \boldsymbol{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\boldsymbol{x},\boldsymbol{x}) + \Sigma_x & K(\boldsymbol{x},\boldsymbol{x}_*) \\ K(\boldsymbol{x}_*,\boldsymbol{x}) & K(\boldsymbol{x}_*,\boldsymbol{x}_*) \end{bmatrix}\right) \tag{5.8}$$

where

$$\Sigma_x = diag((t_k\theta_{k,j})^{-1}) \tag{5.9}$$

Specifically, Σ_x is the diagonal matrix of the noise terms that defined the variability of each data point and, in our model, it is given by $\theta_{k,j} \cdot t_k$. Notice that, if such noise terms are constantly set to σ_q, then Equation 5.3 is the same likelihood as the standard GP with $\Sigma_x = \sigma_q I_q$.

Under such a model, predictions of f can be made by conditioning $\boldsymbol{x}_*$ to the set of reported observations $\boldsymbol{x}$ and $\boldsymbol{y}$, given the trust parameters $\boldsymbol{t}$. Then, using the marginalisation properties of the Gaussian distributions, the predictive distribution of $f(\boldsymbol{x}_*)$ at the test location, is derived as follows:

$$p(\boldsymbol{y_*}|\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}_*, \boldsymbol{t}) = \mathcal{N}(E[\boldsymbol{y}_*], \sigma^2(\boldsymbol{y}_*)) \tag{5.10}$$

where

$$E[\boldsymbol{y}_*] = K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x]^{-1}\boldsymbol{y} \tag{5.11}$$

$$\sigma^2(\boldsymbol{y}_*) = K(\boldsymbol{x}_*, \boldsymbol{x}_*) - K(\boldsymbol{x}_*, \boldsymbol{x})[K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x]^{-1}K(\boldsymbol{x}, \boldsymbol{x}_*) \tag{5.12}$$

Specifically, the equations above are respectively the predictive mean and variance of f at the location $\boldsymbol{x}_*$ with our TrustHGP. Recall that these equations are all conditioned on the values of the set of hyperparameters $\boldsymbol{\Theta} = \{\sigma_f, l, t_1, \ldots, t_k\}$. Since these hyperparameters are typically unknown, their value needs to be estimated in some way as part of the model selection. In particular, a standard technique for estimating hyperparameters in GP models is marginal likelihood optimisation, which sets their values by maximising the evidence of the observations according to the marginal likelihood of the model (Rasmussen and Williams, 2006). This method is particularly convenient for our model since it is possible to derive the expression of the marginal likelihood in closed form by marginalising out f from Equation 5.7 over the GP prior of Equation 5.2 as follows:

$$\begin{aligned}\mathcal{L} &= \ln\Big(\int p(\boldsymbol{y}|f, \boldsymbol{x})p(f|\boldsymbol{x})df\Big) \\ &= -\frac{1}{2}\boldsymbol{y}^T C^{-1}\boldsymbol{y} - \frac{1}{2}\ln|C| - \frac{q}{2}\ln(2\pi)\end{aligned}$$

where $C = K(\boldsymbol{x}, \boldsymbol{x}) + \Sigma_x$. The partial derivatives of the marginal likelihood over Θ are:

$$\frac{\partial\mathcal{L}}{\partial\Theta} = \frac{1}{2}\boldsymbol{y}^T C^{-1}\frac{\partial C}{\partial\Theta}C^{-1}\boldsymbol{y} + \frac{1}{2}tr\left(C^{-1}\frac{\partial C}{\partial\Theta}\right)$$

and factoring in the expression of the squared exponential function (Equation 5.3), we find that:

$$\frac{\partial C}{\partial\sigma_f} = 2\sigma_f \exp\Big(-\frac{d^2}{2l^2}\Big) \tag{5.13}$$

$$\frac{\partial C}{\partial l} = -\frac{\sigma_f^2 d^2}{l^3}\exp\Big(-\frac{d^2}{2l^2}\Big) \tag{5.14}$$

$$\frac{\partial C}{\partial t_i} = -\frac{1}{t_i^2}\mathrm{diag}(0, \ldots, 0, \theta_{i,1}, \ldots \theta_{i,p_i}, 0, \ldots, 0)^{-1} \tag{5.15}$$

Then, we set the values of the hyperparameters as:

$$\boldsymbol{\Theta}_{\mathrm{ML}} = \{\sigma_{f,\mathrm{ML}}, l_{\mathrm{ML}}, \boldsymbol{t}_{\mathrm{ML}}\} = \arg\max_{\sigma_f, l, \boldsymbol{t}}(\ln p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t}, \sigma_f, l) \tag{5.16}$$

At this point, we derived the key equations of the TrustHGP to predict the mean (Equations 5.11) and the variance (Equations 5.12) of f from the data at any input location. Furthermore, the analytical expression of the marginal likelihood is suitable for optimising the trust parameters: To complete this step, we now describe an efficient algorithm

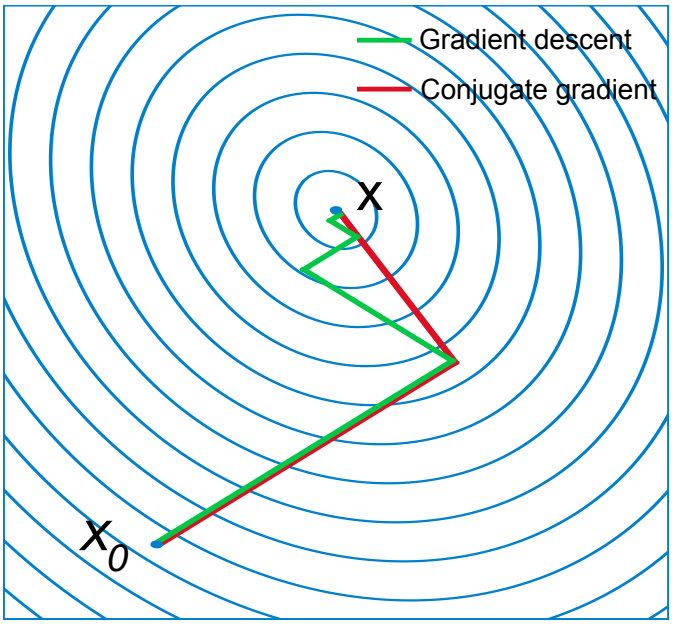


Figure 5.2: A comparison of the convergence of gradient descent (green) to conjugate gradient (red) in minimising a quadratic function.

needed for the training of the TrustHGP and learning both the kernel hyperparameters and the trust parameters.

5.2 The TrustHGP Training Algorithm

In this section, we describe the algorithm for computing the trustworthiness parameters and, on such a basis, predict the values of f. In particular, we estimate such parameters using the standard maximum marginal likelihood criterion described by Equation 5.16. However, since such a likelihood is a non-linear function, its maximisation is not tractable analytically and it must be carried out numerically. We do this by using an efficient gradient–based method for such a function optimisation that is able to leverage the analytical gradients of the hyperparameters that we derived in Equations 5.13, 5.14 and 5.15 . Within the family of gradient–based optimisation algorithms, we use the standard non-linear conjugate gradient method that is commonly used for iteratively minimising quadratic functions (that in our case it corresponds to minimising the negative marginal log–likelihood function) following the steepest conjugate gradient direction (Saad, 1996). In particular, as it is also illustrated in Figure 5.2, such a method typically converges to a (local) minimum faster than the standard gradient descent that follows perpendicular (zig-zag) directions.

Our TrustHGP training algorithm is described in Algorithm 5.1. Step 1 and 2 initialise $\boldsymbol{t}, l$ and σ_f to a random value. Then, the conjugate gradient loop (steps 5-12) computes the gradient with respect to the hyperparameters of the previous iteration and

Algorithm 5.1 TrustHGP (Non-linear conjugate gradient)

Variables :

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

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

1: $\boldsymbol{t}^{(0)} :=$ random initialisation
2: $\boldsymbol{\Theta}^{(0)} := \langle \sigma_f^{(0)}, l^{(0)}, \boldsymbol{t}^{(0)} \rangle$
3: $\gamma^{(0)} := -\frac{\partial}{\partial \Theta} \left(\ln p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Theta}^{(0)}) \right)$
4: $s := 0$
5: **while** ($|\boldsymbol{\Theta}^{(s-1)} - \boldsymbol{\Theta}^{(s)}| < err$ **and** $s < s^{\max}$) **do**
6: $\quad s := s + 1$
7: $\quad \Delta\boldsymbol{\Theta}^{(s)} := -\frac{\partial}{\partial \Theta} \left(\ln p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Theta}^{(s-1)} \right)$
8: $\quad \beta^{(s)} := \frac{(\Delta\boldsymbol{\Theta}^{(s)})^T (\Delta\boldsymbol{\Theta}^{(s)} - \Delta\boldsymbol{\Theta}^{(s-1)})}{(\Delta\boldsymbol{\Theta}^{(s-1)})^T \Delta\boldsymbol{\Theta}^{(s-1)}}$ (Polak-Ribière method)
9: $\quad \gamma^{(s)} := \Delta\boldsymbol{\Theta}^{(s-1)} + \beta^{(s)} \gamma^{(s-1)}$ (Wolfe line search)
10: $\quad \alpha^{(s)} := \arg\max_\alpha p(\boldsymbol{y}|\boldsymbol{x}, (\boldsymbol{\Theta}^{(s-1)} + \alpha\gamma^{(s-1)}))$
11: $\quad \boldsymbol{\Theta}^{(s)} := \boldsymbol{\Theta}^{(s-1)} + \alpha^{(s)} \gamma^{(s)}$
12: **end while**
13: $\boldsymbol{\Theta}^{(s)} := \langle \sigma_f^{(s)}, l^{(s)}, \boldsymbol{t}^{(s)} \rangle$
14: Compute $E[\boldsymbol{y}_*|\boldsymbol{x}_*]$ as by Equation 5.11.
15: Compute $\sigma^2(\boldsymbol{y}_*|\boldsymbol{x}_*)$ as by Equation 5.12.
16: **return** $(\boldsymbol{t}^{(s)}, E[\boldsymbol{y}_*], \sigma^2(\boldsymbol{y}_*))$

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, 1996). Since the purpose of evaluating different versions of these algorithms is out of the scope of this thesis, we use the standard Polak-Ribiere method (step 8) that provided by the GPML Matlab toolbox[4]. Therefore, 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 to update of the step length, the method guarantees 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 was found to be reached in 20-40 iterations[5], the algorithm returns the values of $\boldsymbol{t}$ and the other hyperparameters computed in the last iteration, together with the mean and variance predictions of the function at the test inputs $\boldsymbol{x}_*$. Analysing its complexity, the algorithm requires $O(q^3)$ time to compute the output due to the inversion of the covariance matrix. This is a lower-bound complexity of inference in GP methods (Rasmussen and Williams, 2006). However, after the inversion of the the covariance matrix, prediction only takes $O(q)$ time for the predictive mean and $O(q^2)$ for the predictive variance. In practice, we were able to train our model on up to $2,500$ data points in approximately 5 minutes on a 4 Core i5 3.6 GHz CPU, 8GB RAM architecture.

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

5.3 Experimental Evaluation

In our evaluation, we compare the performance of the TrustHGP to other non–trust GP and HGP methods described in Section 5.3.1 through two set of experiments. Using the same methodology as in the previous evaluation, we devise two set of experiments considering both synthetic data and real data. Specifically, in the first experiment, we run simulations of crowd users with different levels of trustworthiness in order to test the robustness of the methods in a controlled crowdsourcing setting (Section 5.3.2). Then, in the second experiments, we look at the key disaster response application of crowdsourced radiation monitoring evaluating the methods in making spatial predictions on a dataset of crowdsourced radiation data from the 2011 Fukushima earthquake in Japan (Section 5.3.3).

[4]GPML Matlab toolbox website: `www.gaussianprocess.org/gpml/code/matlab/doc/`
[5]This results refers to our test on simulated data presented in Section 5.3.2

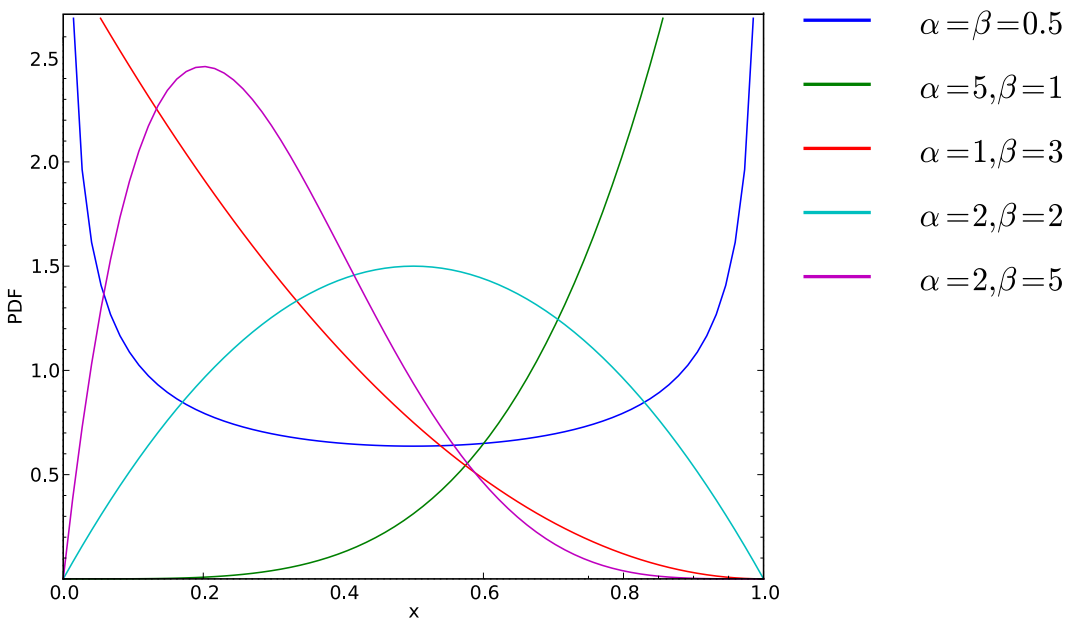


Figure 5.3: Beta function for different values of shape parameters.

5.3.1 Benchmarks

To evaluate our TrustHGP against other non-trust GP and HGP methods, we consider the following benchmarks:

- **GP**: This algorithm is the standard GP that assumes homoskedastic data noise as described Section 2.5.2.
- **HGP**: This algorithm is the HGP (Section 2.5.1) that assumes individual Gaussian noise terms for each input point with noise precision defined by $\theta_{k,j}$. In particular, this method is equivalent to our TrustHGP with all the trust parameters statically set to one, i.e. all the reports are equally trusted.
- **Optimal HGP**: This is the *hypothetical* optimal HGP regression which is obtained by providing the TrustHGP with the correct values of the trustworthiness of each user. That is, trustworthy users are set with $t_k = 1$ and untrustworthy users are set with $t_k = 0$. Notice we can only run this method for the case of synthetic data, since we do not have the ground truth of the values of the user's trustworthiness for the Fukushima experiment.

In summary, we evaluated four different GP models: {HP, HGP, Optimal HGP, TrustHGP} in performing regression with crowdsourced spatial data. Our experiments are presented in the following sections.

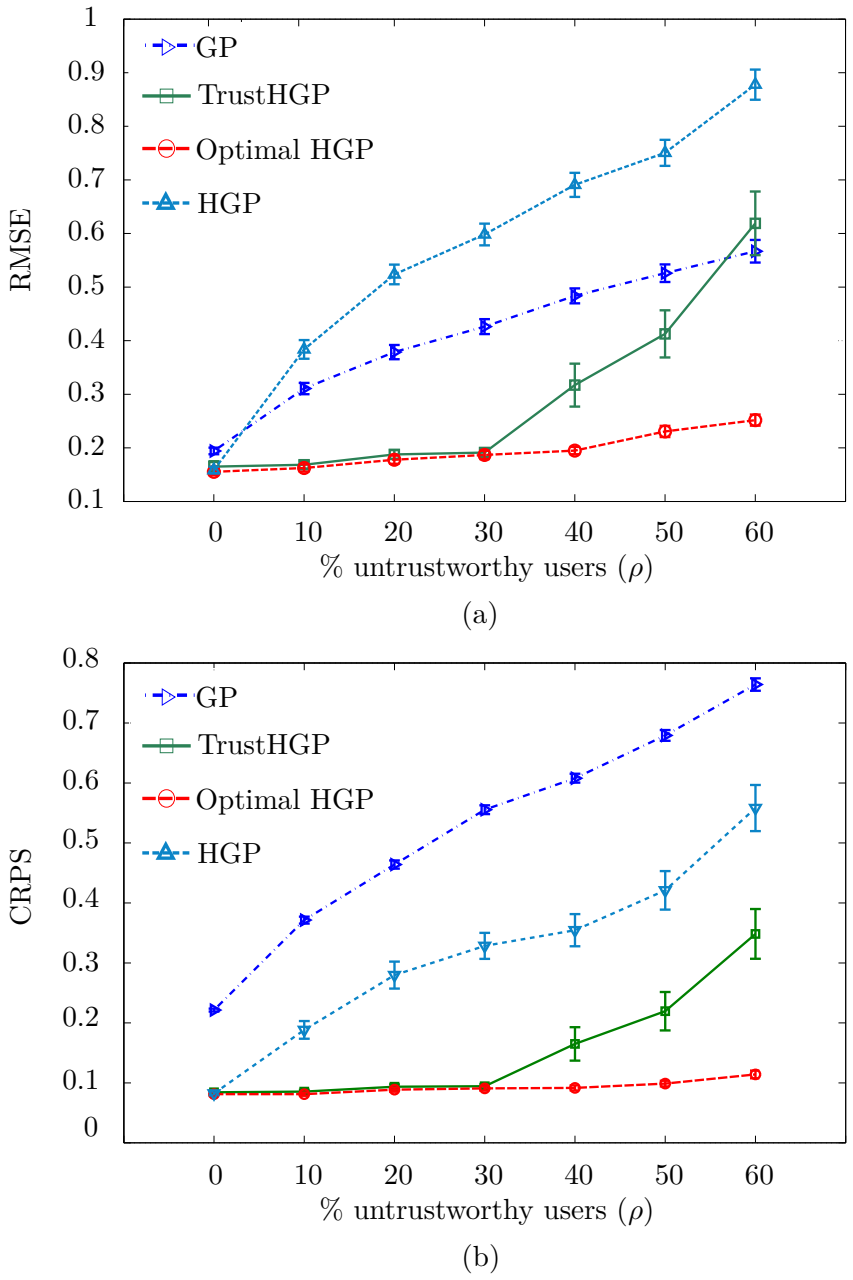


Figure 5.4: Performance of the four methods measured by the root mean square error (RMSE) (a) and the continuous ranked probability score (CRPS) (b).

5.3.2 Experiment on Synthetic Data

In this experiment, we evaluate TrustHGP in estimating a one-dimensional function from synthetic data with simulated trustworthiness features. Specifically, the experiment is set up as follows. We simulate f as a Beta function, $Beta(\alpha, \beta)$ with the two shape parameters α and β randomly sampled as $\{\alpha, \beta\} \sim U[1, 20]$. In particular, the choice of a Beta function with uniformly random shape parameters guarantees a sufficient variability of the shape of f in the simulation, as it also illustrated by Figure 5.3. Then, we simulate a set of observations of f reported by a crowd of 20 users. Each user k

reports p_k estimates, with $p_k \sim [3, 20]$ and each estimate is a vector as described in our model: (i) $x_{k,j}$, i.e. a point randomly selected in [0,1] (i.e. the domain of f) for the j-th report of user k, (ii) $y_{k,j}$, i.e. the observation of $f(x_{k,j})$ and (iii) $\theta_{k,j}$, i.e. the reported precision of $y_{k,j}$. To simulate noise in the observations, the parameters of each estimate are randomly generated as follows:

$$\theta_{k,j} \sim U[0.5, 20] \qquad x_{k,j} \sim U[0, 1]$$
$$y_{k,j} = f(x_{k,j}) + \epsilon_{k,j} \qquad \epsilon_{k,j} \sim \mathcal{N}(0, \theta_{k,j}^{-1})$$

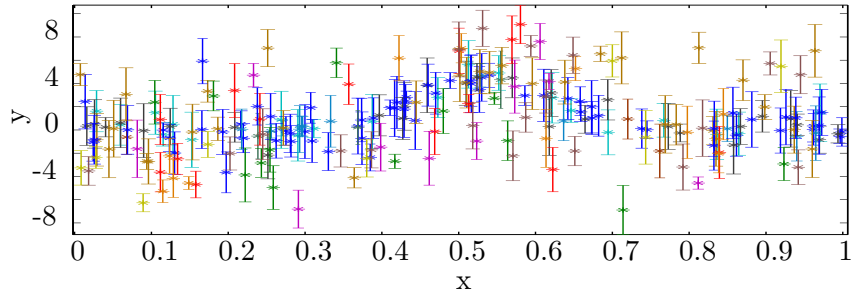
Furthermore, we simulate a percentage ρ of untrustworthy users within the crowd by adding an extra random noise w to some randomly selected users:

$$y_{k,j} = f(x_{k,j}) + \epsilon_{k,j} + w_{k,j} \qquad \epsilon_{k,j} \sim \mathcal{N}(0, \theta_{k,j}^{-1}) \qquad w_{k,j} \sim \pm U[1, 5]$$

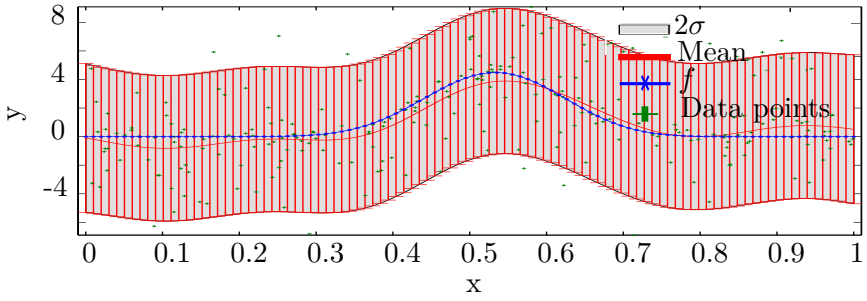
Finally, we measure the accuracy of the set of predictions of the GP methods using the same metrics (RMSE and CRPS) described in Section 3.3.1.2.

The results of 200 simulations varying the value of ρ as follows: $\rho = \{0,\ 10,\ 20,\ 30,\ 40,\ 50,\ 60\}$. are given in Figure 5.4. From this, we can see that, as expected, the RMSE (Figure 5.4(a)) of all the algorithms grows progressively with ρ, i.e. a large presence of untrustworthy users reduces the accuracy of the predictions. However, a key result is that 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, $\alpha = 0.01, p = 3.4 \cdot 10^{-33}$). In particular, its error is very close to the optimum up to $\rho < 30\%$ and is generally the lowest amongst the tested methods up to $\rho < 50\%$. This means that the trust learning adopted by the TrustHGP makes it more robust against the presence of untrustworthy users compared to the other methods. Another interesting result is the CRPS of the four methods showed in Figure 5.4(b). From this, we can see that the CRPS of the TrustHGP is significantly lower than the other methods for any ρ value. In particular, the TrustHGP outperforms the standard GP by 80% when $\rho = 30\%$ (statistical significance tested by a paired t-test, $\alpha = 0.01, p = 3.38 \cdot 10^{-124}$). This means that our algorithm computes the most accurate (i.e lowest RMSE) and also very informative (i.e. lowest CRPS) aggregated predictions in all the settings where the majority of the reports is trustworthy ($\rho < 50\%$). Also, we find that the standard GP ranks below the the HGP in terms of CRPS, even though the former is typically more accurate in terms of RMSE.

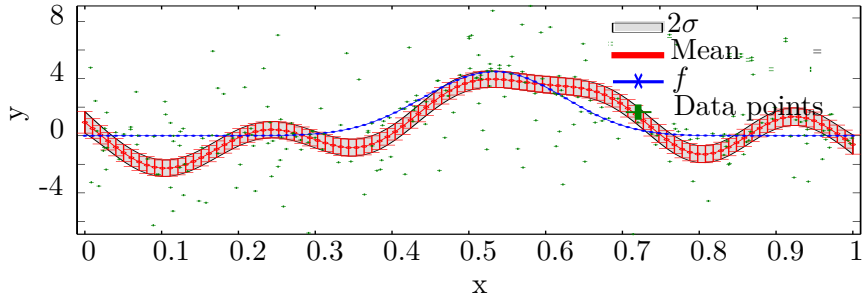
In more detail, Figure 5.5 shows the typical prediction results produced by the four methods. Given the dataset illustrated in Figure 5.5(a) consisting of 241 estimates reported by 20 users, and $\rho = 30$, the standard GP prediction is showed in Figure 5.5(b). In particular, such a prediction is very noisy due to the effect of having a single noise parameter, σ_q (see Section 2.5.2), which is increased by the GP training up to include all the estimates. Interestingly, this way of fitting the noise parameter to the



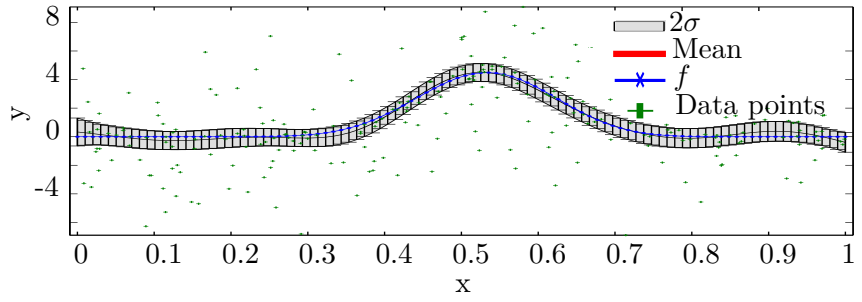
(a) Dataset



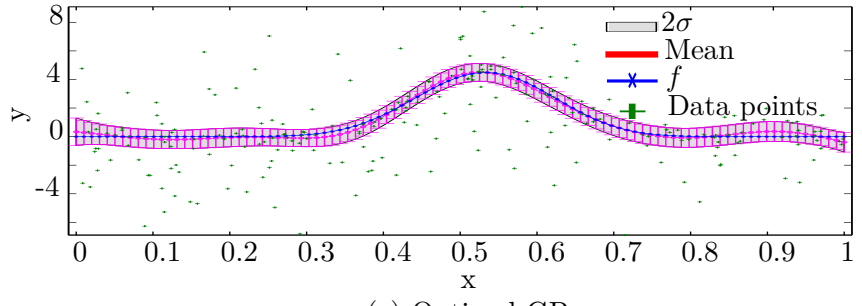
(b) GP



(c) HGP

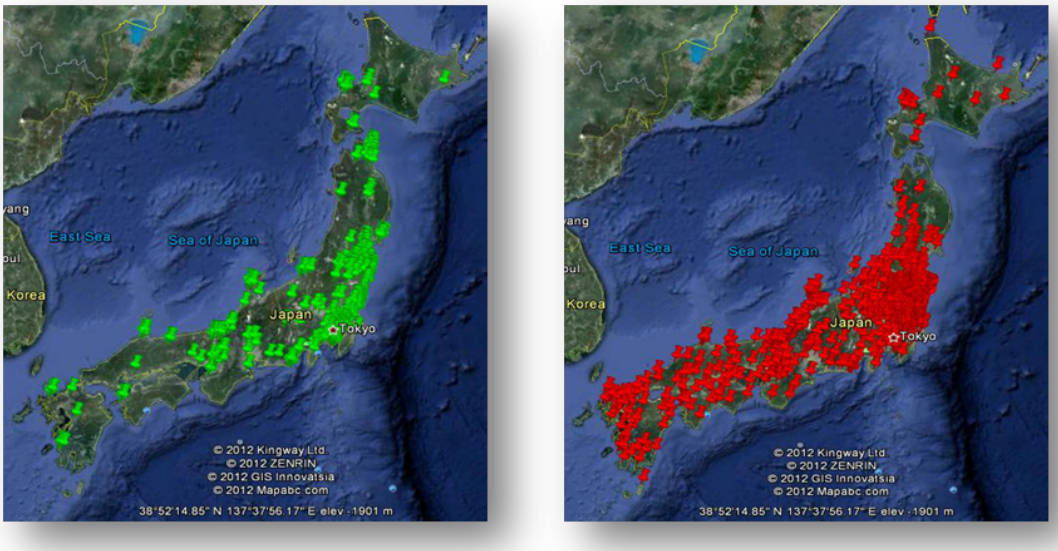


(d) Trust HGP



(e) Optimal GP

Figure 5.5: Example of regression of the four GP methods on a sample synthetic dataset of 20 users, 241 data points and $\rho = 30\%$ untrustworthy users.



(a) Xively crowdsourced sensors (b) SPEEDI official sensors

Figure 5.6: Maps of the 557 radiation sensors of the Xively network (a) and the 2122 radiation sensors of the SPEEDI network (b) located in Japan.

data is comparable to the behaviour of covariance union (CU) that we discussed in our previous experiment with stationary items (see Section 3.3.2). Furthermore, analysing the HGP prediction showed in Figure 5.5(c)) it typically has lower uncertainty but its mean prediction is less accurate due to the effect of following every noisy estimate considered by the model as equally trustworthy. In contrast to all the other methods, the TrustHGP prediction showed in Figure 5.5(d) achieves the highest accuracy and lowest uncertainty. This advantage is due to a correct estimation of trustworthiness parameters that allows the model to exclude most of the untrustworthy estimates. In particular, in this example, its performance is very close to the Optimal HGP showed Figure 5.5(e). Globally, these results shows that our method outperforms the benchmarks in both accuracy and informativeness in predicting the function from synthetic data. To reinforce our claims, we now provide another evaluation of the TrustHGP with real data.

5.3.3 Experiment on Real Data

In this second experiment, we consider the real–world application of crowdsourced radiation monitoring in disaster response introduced in Chapter 1. In particular, we refer to the scenario of the aftermath of the 2011 Fukushima earthquake where an unprecedented effort of local communities contributed a significant volume of crowdsourced radiation data. On 3 March 2011, a tsunami caused by a 9 magnitude earthquake hit the east coast of Japan severely damaging the nuclear power plant of Fukushima-Daichii. The subsequent nuclear accident led to radioactivity increases of up to 1,000 times the normal levels in the area of Fukushima and provoked the second-largest nuclear emergency

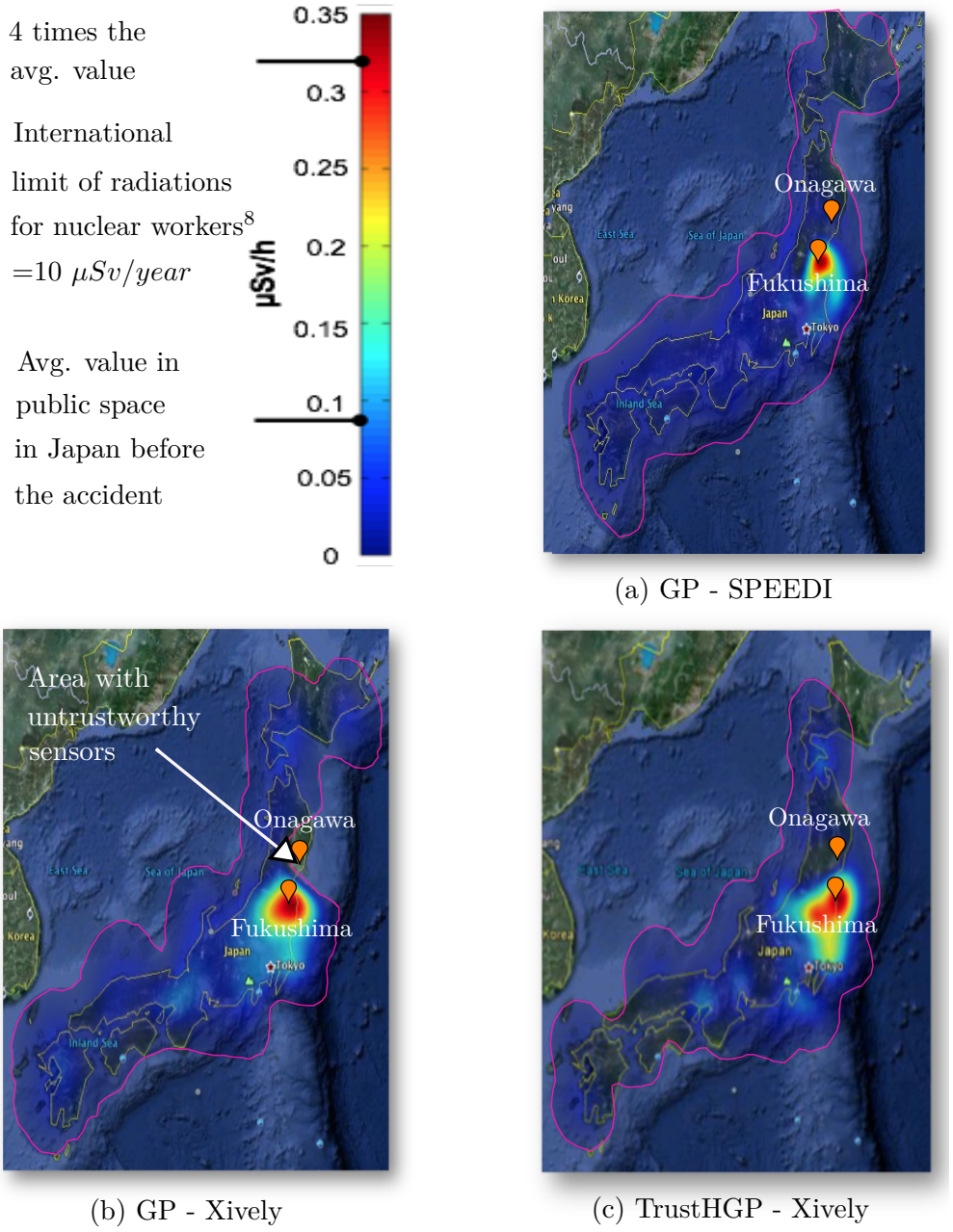
since Chernobyl, 1985. In response, private individuals deployed 557 Geiger counters across the country (many of them based on open-hardware boards such as Arduino or Goldmine) that were able to automatically report radiation data through the the Xively web platform (`xively.com`). This entirely crowdsourced sensor network, which is shown in Figure 5.6 (a), came to life in less than two weeks after the disaster and became a key resource for the public to gather live radiation data from the disaster scene. However, an unknown number of sensors were reporting verifiably wrong measurements (Slater et al., 2012). As a result, the rescue teams faced the key challenge of managing the data streamed by the sensors into a comprehensive spatial radioactivity prediction, while being aware of the untrustworthiness of some sensors. In this context, we now detail how our the TrustHGP can be applied to help address this challenge.

5.3.3.1 Dataset

We collected the readings reported by the Xively sensors over one day, 1 March 2012, one year after the main quake, through the Xively API[6]. In particular, this date was conveniently chosen to allow an overlap with the data provided by the SPEEDI sensors (see later for details) that we used as test data in our evaluation. However, similar results were observed on different dates and the same experiment running on a daily basis is also available at `jncm.ecs.soton.ac.uk`). To define a suitable setting for our TrustHGP, we computed the estimates of the radioactivity at each sensor's location based on the reported readings as follows. We estimate the mean value $y_{k,j}$ and the precision $\theta_{k,j}$ of the readings of each sensor k by taking the average and the inverse variance of the series of its measurements (assuming that only one estimate is reported by each sensor, i.e., $j = 1$). In total, the resulting Xively dataset includes 557 estimates, one from each sensor. The sensors were reporting readings in the unit of microsieverts per hour ($\mu Sv/h$) at an average frequency of 2 readings per hour. The complete description of the Xively dataset in provided in the Appendix C. To build a ground truth for this experiment, we used test data provided by the SPEEDI network: the official radiation monitoring network maintained by the Nuclear Division of the Ministry of Science of Japan (MEXT) (`bousai.ne.jp`)[7]. This network includes 2122 sensors reporting readings at a frequency of 6 readings per hour in the same unit. The map of the SPEEDI network is showed in Figure 5.6(b). Specifically, we use the SPEEDI data to obtain estimates of the radiation levels that are comparable to the ones of the Xively sensors over the same time window. Then, making the reasonable assumption that the SPEEDI sensors are more reliable due to their official source, we use the prediction of the standard GP on the SPEEDI dataset as the spatial radiation levels that we use to evaluate the prediction of our method. In particular, the radiation levels predicted by the GP on the SPEEDI dataset are shown

[6]This dataset and the Java code to query the Xively sensors are available at `eprints.soton.ac.uk/354861`.

[7]At present, the SPEEDI network offers digitalised data only starting from April 2012.



(a) GP - SPEEDI

(b) GP - Xively

(c) TrustHGP - Xively

Figure 5.7: Radiation heat maps showing the following predictions: the standard GP on the SPEEDI dataset (a), the standard GP on the Xively dataset (b) and the TrustHGP on the Xively dataset (c).

[8]Data source: United States Environmental Protection Agency (EPA, 2001)

	RMSE	CRPS
Standard GP	30.80 ± 0.30	64.34 ± 0.04
HGP	64.13 ± 0.99	9.31 ± 0.12
Trust HGP	26.74 ± 0.27	7.14 ± 0.08

Table 5.1: Errors of the three GP methods tested on the Xively dataset.

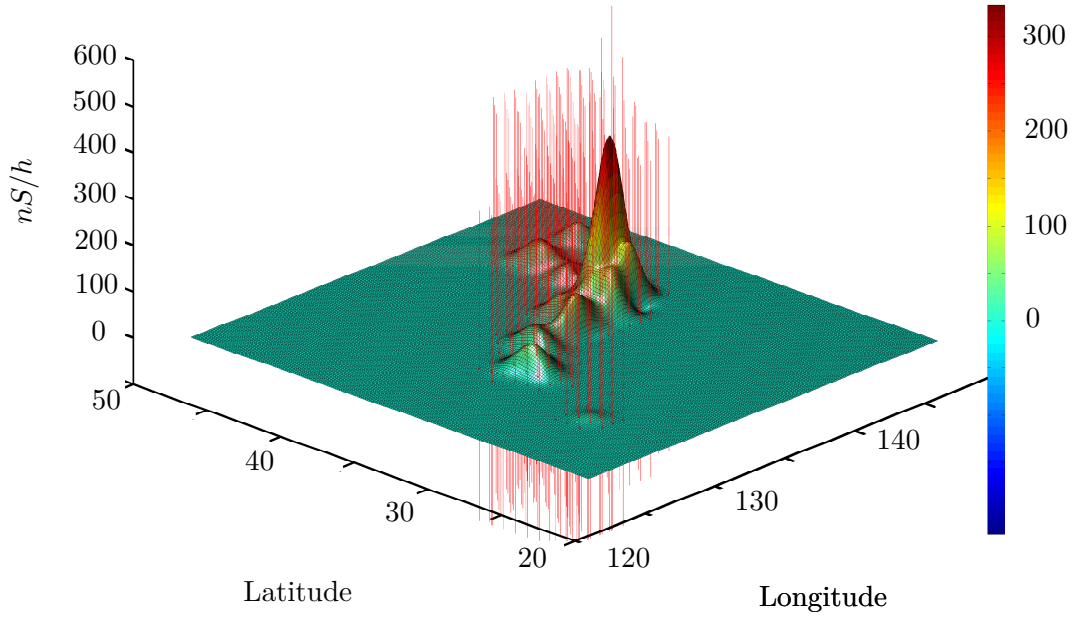
in Figure 5.7(a) as a heat map with a colormap in the scale of 0 - 0.35 $\mu Sv/h$. Given this setting, we present the results of our experiment in the next sub–section.

5.3.3.2 Results

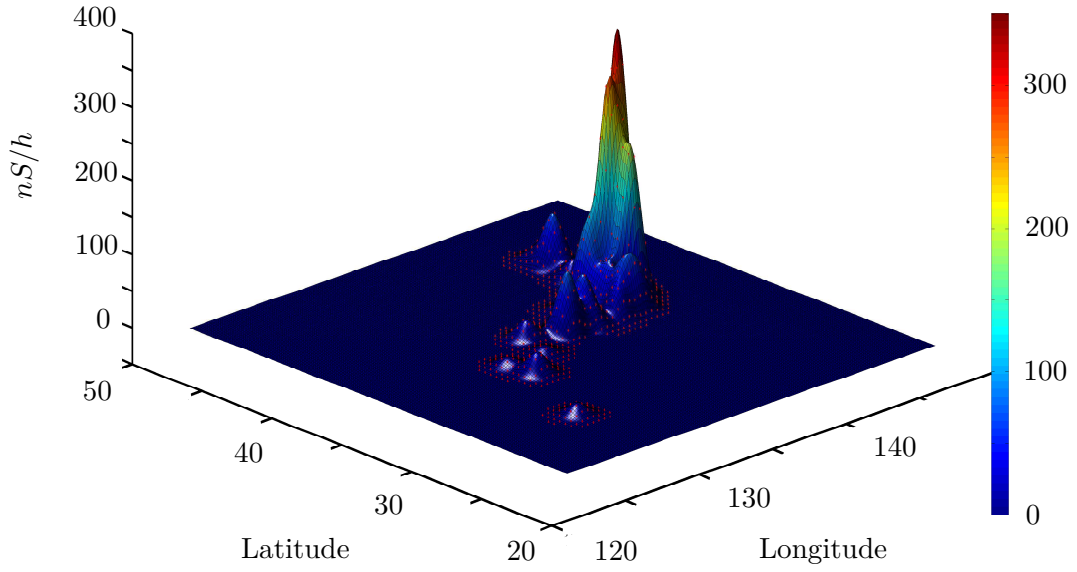
Table 5.1 reports the scores of the predictions of the three methods in 100 trials. In each run, we randomly sample 80% of the sensors in order to evaluate the performance of the tested methods over different portions of the Xively dataset. The results show that the TrustHGP outperforms the best benchmark by 13% with respect to the RMSE (statistical significance tested by a paired t-test, $\alpha = 0.01, p = 2.31 \cdot 10^{-167}$) and by 23% with respect to the CRPS (statistical significance tested by a paired t-test, $\alpha = 0.01, p < 10^{-4}$). In more detail, while the HGP improves the CRPS compared to the standard GP, the RMSE of the former is significantly worse. In contrast, our method achieves the best performance in both the RMSE and CRPS as a result of its correct learning of the sensor's trust values. In particular, its lower CRPS shows that its prediction is considerably more informative than a normal GP. This is even more evident by the 3D visualisation of the two predictions shown in Figure 5.8 where the red bars show the 2σ predictive standard deviations at each location. From this, we can see that the TrustHGP has very narrow (not visible) bars compared to the high bars of the standard GP.

Furthermore, Figure 5.7(b) and Figure 5.7(c) show the predictions of the two methods (GP and Trust HGP) on the Xively dataset depicted as heat maps. These two methods are similar in predicting the peak of radioactivity of approximately 0.33 $\mu Sv/h$ near to the location of the Fukushima power plant, which is approximately four times the average radiation level of 0.09 μSv measured in Japan before the earthquake.[9]. However, their predictions are substantially different in several locations. For example, it can be noticed that the standard GP does not provide valid radiation values near the location of Onagawa (Miyagi prefecture, 38.45 N, 141.44 E). In fact, we manually discovered that some of the sensors located in that area sporadically reported invalid measurements that caused the GP to predict invalid radiation values. In contrast, the Trust HGP makes more plausible predictions and overcomes this issue by correctly learning to place a low degree of trustworthiness on such sensors. In particular, it estimated that 17% of the Xively sensors have trustworthiness values lower than 0.5. The same analysis on the SPEEDI sensors revealed that only few of these (less than 1%) were untrustworthy

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



(a) GP - Xively



(b) TrustHGP - Xively

Figure 5.8: 3D visualisation of the GP prediction (a) and the TrustHGP prediction (b) on the Xively data. The red error bars show the two standard deviations of the radiation levels predicted at each location.

which confirmed our assumption about the SPEEDI network being more reliable. Thus, this result shows that our method provides an effective aggregation of radiation data from real–world crowdsourcing application.

5.4 Summary

In this chapter, we extended our trust-based approach to address the problem of estimating non-stationary quantities from crowdsourced estimates. Specifically, we presented an algorithm that addresses our third requirements related to building reliable aggregations of spatial information in crowdsourcing settings, thus complementing our previous set of algorithms for stationary quantities.

In more detail, we introduced a trust-based heteroskedastic Gaussian process for spatial regression to model a dataset of crowdsourced spatial estimates reported by untrustworthy users. The salient feature of such a model is to integrate user trust learning in spatial regression within a principled heteroskedastic Gaussian process framework. Based on this model, we provided an algorithm to estimate the spatial function describing the phenomenon observed by the crowd and also learn the trustworthiness of the users. Evaluating our algorithm on a set of synthetic data, we showed that it outperforms the standard, non-trust GPs being 34% more accurate and 80% more informative. Furthermore, a real application of our algorithm to the problem of crowdsourced radiation monitoring in Japan showed that our method estimates the radiation levels function with 13% lower error and 23% lower predictive uncertainty compared to the standard GP. With this, our solution is the first to address a spatial regression problem with crowdsourced information that is an important step in making use of intelligent machine learning technologies in trustworthy participatory sensing. Other significant extensions of our model related to space-time based trust estimation, modelling uncertainty over trust parameters and trust–based active learning that could possibly broaden its applicability will be discussed in Chapter 7.

Chapter 6

A Trust-Based Log Gaussian Cox Process Model for Fusing Crowdsourced Spatial Point Data

In this chapter, we address the challenge of modelling spatial patterns in crowdsourced point data, which relates to our last set of requirements (Req. 4). As discussed in Chapter 1, this is a key challenge in the disaster response domain where people on the ground provide important information to emergency responders in the form of geo–tagged data such as tweets and text messages. These reports typically relate to emergency events happening in the disaster area, such as trapped people, requests for food, water and shelter that are categorised and mapped through software platforms such as Ushahidi (`www.ushahidi.com`), Google crisis response tools (`www.google.org/crisisresponse`) or OpenStreetMap (`www.openstreetmaps.org`). For example, Figure 6.1 shows the map of the reports collected by Ushahidi during the 2010 Haiti disaster which were grouped by seven emergency categories that include water contamination, food shortage and power outage[1]. In Chapter 1, we discussed that an important goal for emergency responders who need to undertake relief activities on the ground is to extract information about the areas where supplies are most needed from such a large amount of reports (Gao et al., 2011). In fact, several studies support the use of crowd reports to recover the map of real emergency events by highlighting the fact that the spatial distribution of such crowd reports is highly correlated to the location of actual disaster events (Corbane et al., 2012; Goodchild and Glennon, 2010a). In more detail, a disaster event is likely to generate a number of reports in its surrounding area with an intensity that is correlated to its level of damage. This finding suggests that we can plausibly recover information about the locations of disaster events through the statistical analysis of the spatial patterns of the crowd reports. However, the accuracy and trustworthiness of these reports might,

[1]In total, Ushahidi collected more than 60,000 reports in Haiti that were manually classified by a team of volunteers and subsequently mapped into 3,584 events (Morrow et al., 2011).

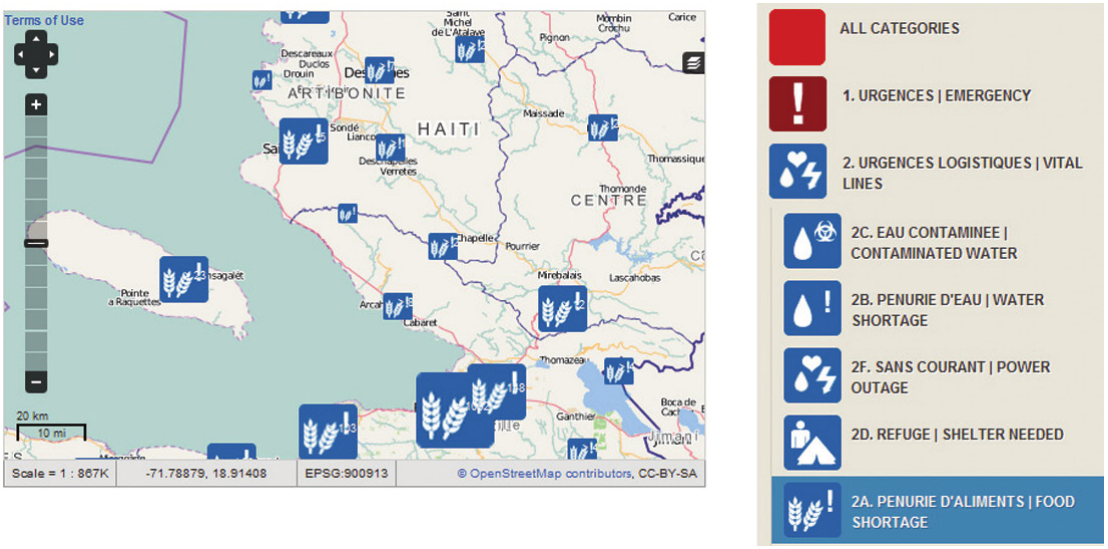


Figure 6.1: The Ushahidi-Haiti map of food shortage reports and other categories of emergencies that were reported by the crowd after the Haiti earthquake[2].

for example, be affected by people who exaggerate their real needs and priority of emergency. Thus, this is a crucial aspect that needs to be taken into account to correctly learn spatial patterns from such data. In this context, we consider the problem of fusing untrustworthy point data (i.e., Req. 4) that relates to the locations of crowdsourced emergency reports in the setting were the data has various categories of trustworthiness, (i.e., Req. 2). By so doing, we wish to provide an automatic learning tool that relies on our trust–based fusion approach to help first responders convert a sparse sets of crowd reports into an actionable map.

As discussed in Chapter 2, the problem of inferring spatial point patterns in crowdsourcing presents some unique challenges compared to the problems we studied so far. In particular, in contrast to spatial regression that we discussed in Chapter 5, where we discussed the problem of inferring a latent function based on its location–based noisy observations provided by the crowd, a key difference is that now we do not directly observe values that are drawn from the continuous function that we wish to estimate. For example, related to the radiation monitoring scenario discussed in Section 5.3.3, observations consisted of the readings of a spatial nuclear radiation field as provided by a number of crowdsourced sensors. Conversely, in spatial point pattern analysis, we only observe a finite number of points (i.e., the reports) placed in some random locations, e.g., we received $\phi(\boldsymbol{X})$ reports in a certain region $\boldsymbol{X}$. Then, our goal is to infer the function that predicts the intensity of reports over the entire space. In fact, the knowledge of such an intensity function is useful to learn information about the most endangered areas by localising places with high values of predicted crowd reporting rates. However, an important requirement of this learning task is to consider the uncertainty

[2]Data source: Gao et al. (2011)

about the reliability of some groups of reports that might deviate from the main spatial point pattern, thus appearing as outliers to the intensity function. For example, these untrustworthy points might be given by groups of reports that are more likely to be generated by isolated events or by reports with locations that are incorrectly reported by the devices. Given this, we require a model for learning spatial point patterns that is able to deal with inaccuracies of the data. In Chapter 2, we identified the Log Gaussian Cox Processes (LGPC) as a suitable basis for a solution to this problem. In fact, the LGCP is a powerful spatial point process model that provides approximate but still tractable probabilistic inference for non–linear intensity functions, which are likely to occur in crowdsourced data. Furthermore, an important advantage of this model is that it represents a natural extension of the Gaussian process to spatial point processes. This allows us to incorporate techniques developed for our TrustHGP model in solving this new problem. In particular, we discussed that, similarly to other GP–based models applied to crowdsourcing, the standard LGCP suffers from the inability to deal with data of varying trustworthiness.

Against this background, we introduce a new trust–based LGCP as the first model designed for a fusion task with untrustworthy spatial point data related to a crowdsourcing context. In particular, the key feature of our model is the ability to simultaneously learn the spatial intensities from a set of given points and the trustworthiness of categories of such points. In more detail, our model is inspired by the setting of Ushahidi where the reports are categorised by a taxonomy of emergency events. Thus, drawing from the theory of our previous trust–based GP model, we use trust parameters to represent the reliability of each category of reports. These reliabilities are then used to scale the correlations of points within categories in a LGCP–based fusion process. By so doing, we demonstrate the versatility of our trust–based fusion approach that can be adapted to different types of trust models. For example, the data trustworthiness can be related to the single report, as in our cell tower application (Chapter 3), to the user, as in our WiFi mapping (Chapter 4) and radiation monitoring application (Chapter 5) or to categories of reports. Specifically, we experimentally show that our model provides 10% more accurate intensity estimates against the state–of–the–art models tested on synthetic data. Furthermore, we apply our model to the Ushahidi–Haiti dataset and show that its spatial intensity prediction effectively overcomes the presence of untrustworthy reports and generally provides more informative intensity maps compared to the non–trust LGCP methods. In addition, we show that our model learns useful information about the trustworthiness of the reports' categories which can be used to define priorities of report verifications or rescue tasks.

Thus, we advance the state of the art as follows:

- We introduce the trust–based Log Gaussian Cox Process (TrustLGCP), the first model that simultaneously learns the spatial intensities of random point process and the trustworthiness of categories of crowdsourced point data.

- We show that our TrustLGCP improves the accuracy of the standard LGCP on synthetic data. We also show that our model can effectively learn intensity maps from crowdsourced emergency reports and provide the estimated trustworthiness of each emergency category with an application to the Ushahidi dataset collected during the 2010 Haiti earthquake.

In the remainder of the Chapter, we first describe our model formally in Section 6.1. We then provide an algorithm for training the model on crowdsourced categorised point data in Section 6.2. Subsequently, we evaluate the method on various datasets in Section 6.3. Finally, we summarise our conclusions in Section 6.4.

6.1 Model Description

In this section, we define a categorical trust model to represent data reliability in our model within the disaster response scenario (Section 6.1.1). Then, we detail the use of such a model in the definition of our trust–based LGCP model (Section 6.1.2).

6.1.1 A Trust Model for Categories of Crowdsourced Reports

In disaster response, we gather a set of N geo–located and categorised emergency reports $R := \{(x_o, c_o) : o = 1, \ldots, N\}$ submitted by the crowd. Each report includes (i) the location of the reported event $\boldsymbol{x}_i \in \mathbb{R}^2$ and (ii) the category of its emergency c_i, selected from a pre–defined list of F emergency categories, for example the Ushahidi categories used in the Haiti deployment (see Morrow et al. (2011)). In particular, we are interested in learning the spatial patterns of such a report set which is expressed by an unobserved *intensity function* $\lambda : \boldsymbol{X} \rightarrow \mathbb{R}$. This function defines the number of reports (i.e. points) that are expected to be observed in a certain region $\boldsymbol{X}$. As discussed in Section 1.2, the knowledge of λ is important to infer information about the location and the strength of the real disaster events, e.g. collapsed buildings, trapped persons and unavailable services. However, the premise for the accurate learning of λ is to be aware of the possible untrustworthiness of some reports that might not necessarily correlate to the true intensity rates. To deal with such uncertainty about the data reliability, we start by making the assumption about the categorical trustworthiness of the reports. That is, we assume that each category c has an individual and unknown trustworthiness level expressed by $t_c \in [0, 1]$ that represents the average reliability of its reports. Specifically, we relate such a categorical trustworthiness to the fitness of the intensities of reports contained in c with respect to the underlying λ function. Thus, similarly to the idea of our previous models (see Section 3.1.1 and Section 5.1.1), untrustworthy data is considered as the instances that do not follow that generative process described by the majority of the trustworthy points. Importantly, our choice of modelling trust over data

categories follows from the setting of Ushahidi in which reports are categorised by their emergency type. However, it is important to notice that alternative representations of data trustworthiness can be defined without significantly changing the structure of our inference model with different parametrisations of uncertainty. For example, trust can be defined at the single report level or at the user level in the same way as defined in our previous models.

Therefore, we want to infer the intensities of reports over a certain region while taking into account their categorical trustworthiness. To do so, we reasonably assume that the global trustworthiness of the intensities observed in $\boldsymbol{X}$ is the average intensity of each category weighted by t_c. More formally, let $\phi(\boldsymbol{X})$ be the number of reports observed in $\boldsymbol{X}$, then the value of $\phi(\boldsymbol{X}|\boldsymbol{t})$ given the vector of categorical trust values $\boldsymbol{t} = \{t_c : c = 1, \ldots, C\}$ is obtained as:

$$\phi(\boldsymbol{X}|\boldsymbol{t}) = \left\lfloor \frac{1}{T} \sum_{c=1}^{C} t_c \phi_c(\boldsymbol{X}) \right\rfloor \tag{6.1}$$

where $T = \sum_{c=1}^{C} t_c$ and $\phi_c(\boldsymbol{X})$ is the partial count of reports of category c located in $\boldsymbol{X}$. Notice that the floor operator is needed to constrain such counts to be integer value. In fact, the generative model of the Poisson process assumes that such intensities are generated from a Poisson distribution, which is defined over integer values. Then, we say that total trustworthiness $t_{\boldsymbol{X}}$ of the report count observed in $\boldsymbol{X}$ is taken as the average of the categorical trust parameters weighted by the report counts of each category. That is:

$$t_{\boldsymbol{X}} = \frac{1}{\phi(\boldsymbol{X})} \sum_{c=1}^{C} t_c \phi_c(\boldsymbol{X}) \tag{6.2}$$

For example, suppose that 15 reports are counted in $\boldsymbol{X}$, 10 of these belong to the category "natural hazards" that we trust with value 0.7 and 5 reports belong to the category "food requests" that we trust with value 0.3. Then, we will consider that an average of 8 reports is observed $\boldsymbol{X}$ and the trustworthiness of such an observation is 0.56. Thus, while a non–trust based aggregation method would simply count all reports as equally trustworthy, our method discounts the report counts of each category with the associated trustworthiness so that the effect of untrustworthy categories is downgraded in the total count. Notice that this trust–based averaging method differs from the noise scaling technique adopted by our previous trust models. This is due to the fact that, in the disaster response setting, the inference process is defined over the integer report counts rather than a continuous function's measurements. Therefore, it is less convenient to apply trust–based noise scaling in our current setting since the crowd reports do not provide the precisions for the counts. On the other hand, our choice of using a trust–based averaging method is a more intuitive way to relate the observed intensities to the reliability of their categories and, on such a basis, enable a two-way transfer learning

mechanism whereby the trustworthiness of the categories informs the one of the single point. In any case, we must be able to infer the trust parameters from the data as a prerequisite to making predictions of intensities with our trust model. To address this, we use a model–based machine learning approach that builds upon the LGCP model.

6.1.2 A Trust-Based Log Gaussian Cox Process Model

From the preliminaries introduced in Chapter 2, the LGCP is a non–homogeneous Cox process model with Poisson intensities that are generated by a latent function $z = \ln(\lambda)$. In particular, this point process model is characterised by the property that the number of reports that are counted in two disjoint and bounded regions $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are independent and Poisson distributed with intensity:

$$\lambda(\boldsymbol{X}_i) = \int_{\boldsymbol{X}_i} \lambda(x)dx \tag{6.3}$$

Then, the LGCP defines a Gaussian Process prior over z (Equation 2.25) in order to model both the uncertainty about z and the random number of points observed in each region (see Section 2.6.3). However, we also discussed that the premises to have a tractable inference in a LGCP model are that (i) the space where the points are observed must be discretised into disjoint bins $\boldsymbol{X} = \cup \boldsymbol{X}_i : \boldsymbol{X}_i = \boldsymbol{X}_{i-1} + \Delta_x$ and (ii) the posterior distribution of the log–intensities z_i associated with each bin i is assumed to be multivariate normal under the Laplace approximation. Following this, the approximate data likelihood of the model factorises over the Poisson distributions of each bin as given by Equation 2.27 and the predictive posterior distribution at new test points $\boldsymbol{x}_*$ is approximately multivariate normally distributed with p.d.f. as given by Equation 2.28. Under this model, we can compute predictions of the intensities at any location of interest after selecting an appropriate kernel K for the GP prior of z (assuming a zero mean GP prior). In particular, assuming a smoothly changing correlations in the spatial intensities, we can choose K to be a squared–exponential kernel (Equation 2.14) with a single–variance noise matrix $\sigma_N^2 \boldsymbol{I}_N$.

Now, we consider the feature of having different trustworthiness around each intensity z_i. In this setting, the standard LGCP would be prone to the error of assigning high intensities even to regions where the points that belong to untrustworthy categories are located. To rectify this, we design a new LGCP kernel that allows the model to flexibly increase the uncertainty around regions of untrustworthy points while still modelling correlations in the locality of such regions. Specifically, we define a trust–based LGCP kernel where the uncertainty of each region of the intensities is scaled by the trust

parameters as follows:

$$\Sigma_{\text{trust}} = diag(t_{\boldsymbol{X}_1}, \ldots, t_{\boldsymbol{X}_N})^{-1} \tag{6.4}$$

$$K_{\text{trust}} = K + \sigma_N^2 * \Sigma_{\text{trust}} \tag{6.5}$$

That is, Σ_{trust} is a diagonal matrix of the inverse trust parameters that regulates the variance of the process noise in each region. In this way, untrustworthy reports are downgraded by the effect of decreasing the correlation of points located in untrustworthy regions. Notice that the configuration of $\boldsymbol{t} = \{1, \ldots, 1\}$ reduces K_{trust} to the standard single–variance LGCP model.

Then, following the inference steps of Section 2.6.3, we derive the predictive distribution at the test points $z_* = z(\boldsymbol{x}_*)$ as:

$$p(z^*|\boldsymbol{R}, \boldsymbol{\Theta}) \approx \mathcal{N}(E[z_*], \sigma^2(z_*)) \tag{6.6}$$

where

$$E[z_*] = K_{\text{trust}}(\boldsymbol{x}_*, \boldsymbol{x}) K_{\text{trust}}(\boldsymbol{x}, \boldsymbol{x})^{-1} \hat{\boldsymbol{z}} \tag{6.7}$$

$$\sigma^2(z_*) = K_{\text{trust}}(\boldsymbol{x}_*, \boldsymbol{x}_*)(K_{\text{trust}}(\boldsymbol{x}, \boldsymbol{x}) + \hat{\boldsymbol{\Sigma}}^{-\mathbf{1}})^{-1} K_{\text{trust}}(\boldsymbol{x}_*, \boldsymbol{x}) \tag{6.8}$$

where $\hat{\boldsymbol{\Sigma}}^{-\mathbf{1}}$ is the negative Hessian matrix of the likelihood around its mode (see Equation 2.33). Thus, the equations above fully characterise our trust–based LGCP model, hereafter named TrustLGCP. In particular, they provide the predictive equations of the spatial intensities obtained by combining the empirical intensities observed in the reports with trustworthiness of their categories. Furthermore, we can derive the approximate marginal likelihood of the model by factoring K_{trust} in Equation 2.34 as follows:

$$\begin{aligned} \ln p(\boldsymbol{z}|\boldsymbol{R}, \boldsymbol{\Theta}) = &- \frac{1}{2} \hat{\boldsymbol{z}}^T K_{\text{trust}}(\boldsymbol{x}, \boldsymbol{x})^{-1} \hat{\boldsymbol{x}} + \ln p(\boldsymbol{R}|\hat{\boldsymbol{z}}) \\ &- \frac{1}{2} |I_N + \hat{\boldsymbol{\Sigma}}^{\frac{1}{2}} K_{\text{trust}}(\boldsymbol{x}, \boldsymbol{x}) \hat{\boldsymbol{\Sigma}}^{\frac{1}{2}}| \end{aligned} \tag{6.9}$$

In particular, such a marginal likelihood is important for training the model, i.e., finding the best kernel hyperparameters and trust parameters for our dataset. Using the same training approach described for the TrustHGP in Chapter 5, we can learn the values of the hyperparameters $\boldsymbol{\Theta} = \{\sigma_f, l, \sigma_N, \boldsymbol{t}\}$ (σ_f and l are the two hyperparameters of k) by optimising such a likelihood function. In more detail, an algorithm that implements such an optimisation for the TrustLGCP training is described next.

Algorithm 6.1 TrainLGCP (Non–linear Conjugate Gradient)

Variables :

$\boldsymbol{R}$ **:** Report set
Δ_x **:** Resolution of the spatial grid
$\boldsymbol{z}$ **:** Mean value of the predicted log-intensities
$\boldsymbol{\sigma^2}$ **:** Variance of the predicted log-intensities
$\boldsymbol{t}$ **:** Trustworthiness parameters

Algorithm *TrustLGCP*

1: Partition space into bins of size: $\Delta_x \times \Delta_x$
2: $\phi(\boldsymbol{X}_i) \leftarrow$ Number of points in each bin $\boldsymbol{X}_i$.
3: $\Theta^{(0)} \leftarrow$ Initialise hyperparameters
4: **repeat**
5: $\hat{\boldsymbol{z}} := \arg\max_{\boldsymbol{z}} p(\boldsymbol{z}|\lambda, \Theta^{(s-1)})$ (Compute MAP estimates)
6: $\hat{\Sigma} := - \bigtriangledown \bigtriangledown \ln p(\hat{\boldsymbol{z}}|\lambda, \Theta^{(s-1)})$
7: $\Delta\Theta^{(s-1)} := \frac{\partial}{\partial \theta^2} \ln p(z|\boldsymbol{R}, \Theta^{(s-1)}, \hat{\boldsymbol{z}}, \hat{\Sigma})$ (Compute likelihood gradient)
8: $\Theta^{(s)} := \Theta^{(s-1)} - \alpha\Delta\Theta^{(s-1)}$ (Update hyperparameters)
until convergence
9: $z, \sigma^2 \leftarrow$ Compute the predictive mean and variance of the intensities under $\Theta^{(s)}$ (Equation 6.7 and Equation 6.8)
10: $\boldsymbol{t} \leftarrow$ Trustworthiness parameters in $\theta^{(i)}$
11: **return** $(\boldsymbol{z}, \sigma^2, \boldsymbol{t})$

6.2 The TrustLGCP Training Algorithm

In this section, we describe a training algorithm our model based on marginal likelihood maximisation; a standard training method for GP models (Rasmussen and Williams, 2006). This method finds local estimates of the hyperparameters as given by $\boldsymbol{\Theta}_{\texttt{ML}} = \arg\max_{\Theta} p(\boldsymbol{R}|\Theta)$. To do so, we must consider the fact that this likelihood is typically a non–convex function that requires numerical techniques to perform such a function's maximisation. As per the TrustGP (Section 5.2), we consider gradient–based methods as they typically provide a faster convergence compared to the standard first–order optimisation algorithms. In particular, we focus on the conjugate gradient method as it was already showed to provide efficient performances with the TrustHGP described in Chapter 5. Typically, this method is suitable to optimise functions with a moderately large set of hyperparameters with polynomial computation (Saad, 1996). Therefore, we describe its steps in Algorithm 6.1. In step 1-2 the algorithm partitions the space into bins of size Δ_x and pre–computes the number of points in each bin. Steps 5-6 compute the MAP values required by the Laplace approximation and steps 7 updates the gradient of the approximate marginal likelihood. Step 8 updates the value of the hyperparameters based on the new gradient. These steps are then repeated for a finite number of epochs until the parameters reach the convergence. The complexity of this computation is dominated by the $O(N^3)$ time taken by the inversion of the covariance matrix involved in computing the likelihood gradient (step 7). In practice, we were able to run this algorithm on our largest set of 2774 points on a 4 Core i5 3.6GHz CPU, 8GB RAM in approximately 7 minutes

Having described our model of the reports, we now turn to its experimental evaluation.

6.3 Experimental Evaluation

In this section we provide empirical insights into the performance of our TrustLGCP. With the same methodology as all our previous evaluations, we first describe the set of benchmarks that we test our method against in Section 6.3.1. Then, we present an experiment on synthetic data to quantitatively evaluate the robustness of our method against different levels of trustworthiness in the data in Section 6.3.2. Subsequently, we run experiments on a real–world dataset consisting of the Ushahidi–Haiti emergency reports to show the efficacy of our method in practice in Section 6.3.3.

6.3.1 Benchmarks

To show that our method outperforms the state of the art, we compare its performance against two non–trust based existing benchmarks:

- **LGCP**: This algorithm is the non–trust version of the standard LGCP that does not explicitly model data trustworthiness (see Section 2.6.3). That is, this algorithm is equivalent to a TrustLGCP where the trust parameters are not learned but they are instead set all statically to one.

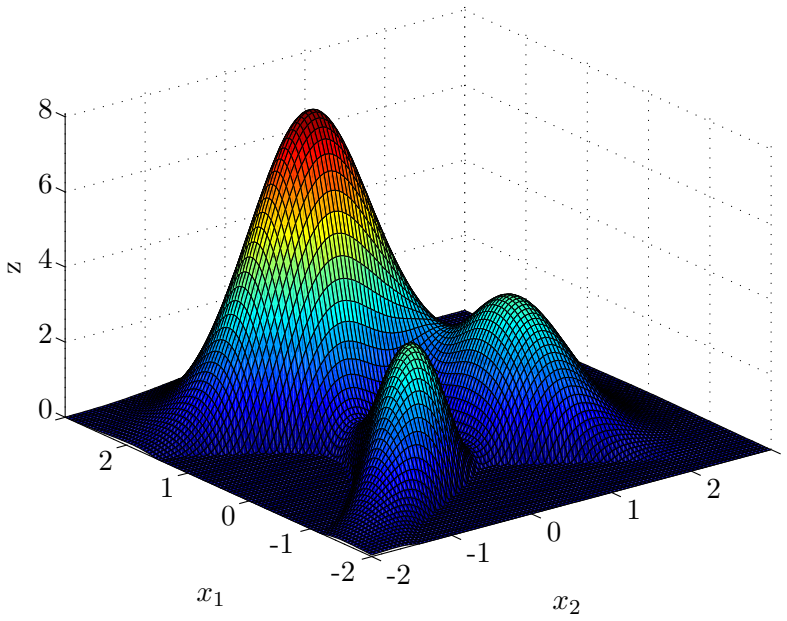
- **Optimal LGCP**: This method is equivalent to a TrustLGCP in which there is perfect knowledge of the trust parameters. That is, it is a TrustLGCP with the values of t_c that are set to their simulated values, thus providing the best possible filtering of untrustworthy data. Notice that we can only run this method for the case of synthetic data, since we do not have the ground truth of the categories' trustworthiness for the experiment with real data.

In particular, compared to the evaluation presented in Chapter 5, this set of benchmarks does not comprise the heteroskedastic version of our model with individual noise rates for each observation. This is due to the fact that our current setting does not consider points with reported precisions, which is a possible direction for future work. As a result, it is not trivial to define heteroskedastic noise terms in our current model using the Ushahidi–Haiti data. In summary, the set of three methods {LGCP, Optimal LGCP, TrustLGCP} were evaluated in various spatial point process learning tasks. As per the previous evaluations, the accuracy of each method is measured by the RMSE and the CRPS (see section 3.3.1.2).

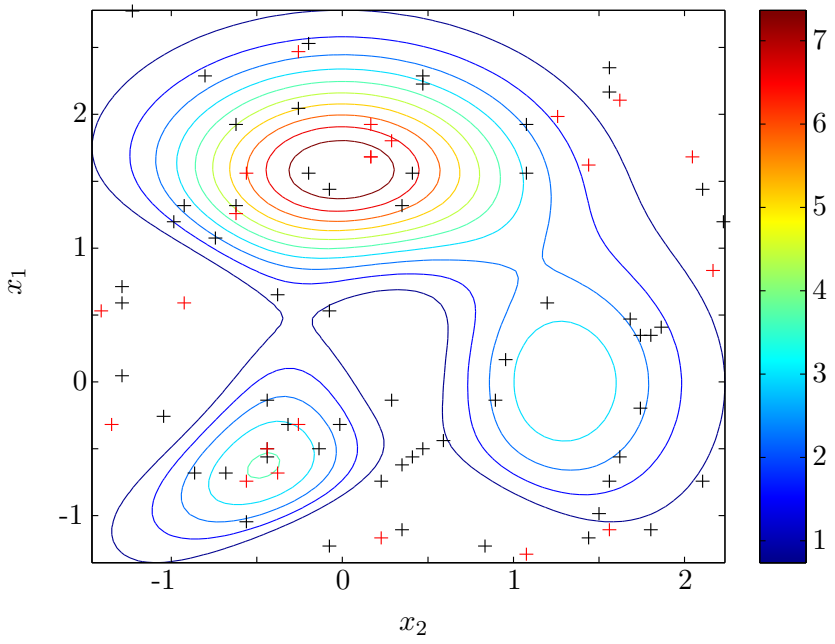
6.3.2 Experiment on Synthetic Data

In this first experiment, we test our method on the task of learning spatial intensities from a synthetic dataset comprising points that are generated from a number of trustworthiness categories. Specifically, we simulate two categories of trustworthiness as given by two disjoint groups of points sharing the same average trustworthiness. In particular, these two categories are randomly assigned with a trust value $t_1 \sim U[0.6, 1]$ to more trustworthy points, and $t_2 \sim U[0, 0.4]$, to less trustworthy ones. Notice it is trivial to extend this setting to multiple categories, which will mainly result in a more complex, but still feasible to optimise, search space for the parameters. However, this binary categorisation setting is useful for us to investigate the robustness of our learning method without requiring a high computational overhead. Then, we take the two–dimensional "peaks" function[3] to simulate the true intensity map as shown in Figure 6.2 (b). The input space of this function is bounded in $x_1 \in [-2, 3]$ and $x_2 \in [-2, 3]$. Then, we generate a set of observations of point intensities from the two categories as follows. Firstly, we partition the space into a regular grid with step $\Delta_x = 0.05$ over the x_1 and x_2 axis. Secondly, we randomly take a subset of 1000 cells X_i and for each of them we sample the

[3]The peaks function is a Matlab example function for two variables obtained by translating three Gaussian distributions (`www.mathworks.co.uk/help/matlab/ref/peaks.html`)

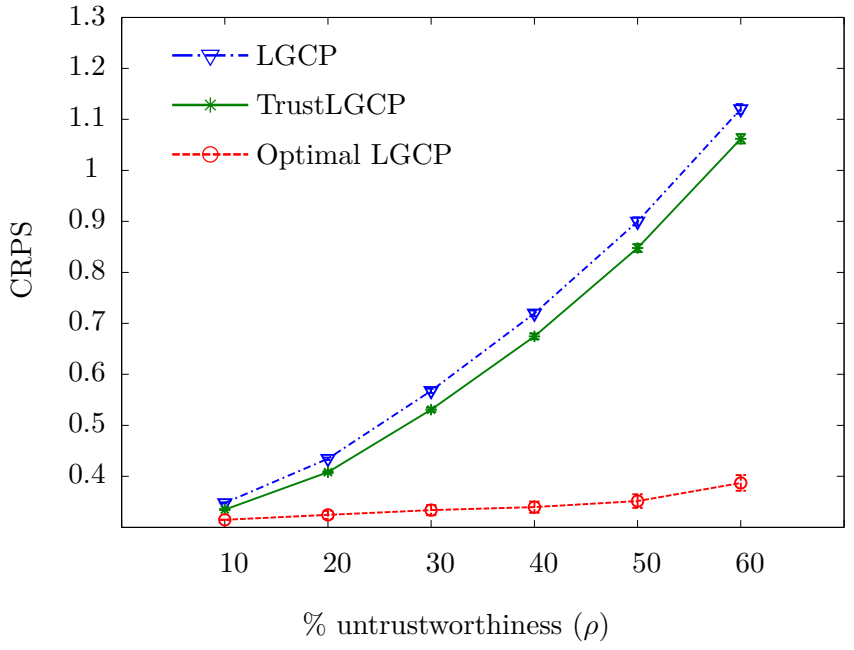


(a) Ground truth intensity function

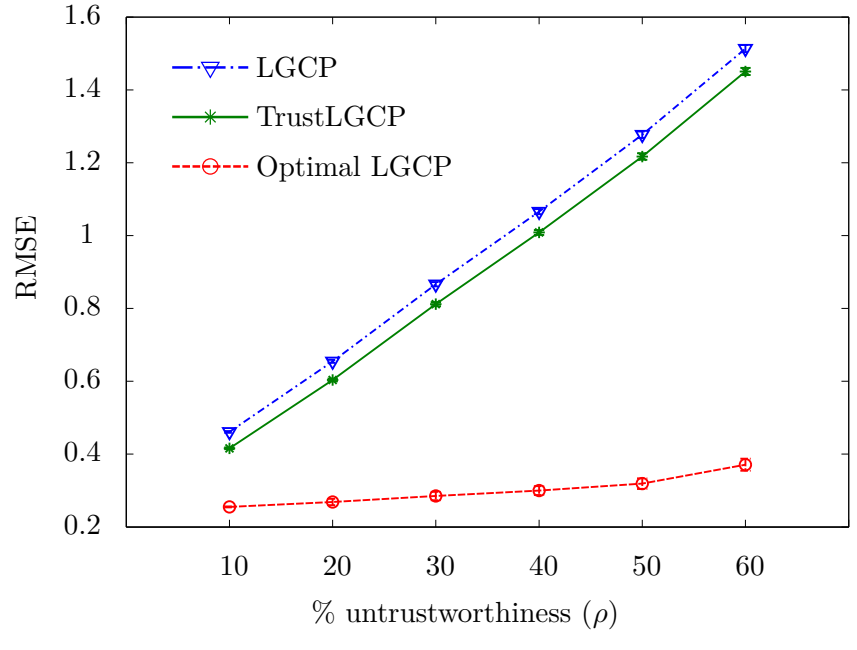


(b) Intensities contour and sampled points

Figure 6.2: The 3D plot (a) and the contour plot (b) of the"peak" function used as the ground truth in our first experiment. Figure (b) shows an example dataset of intensities sampled from two categories related to this function. The black + are intensities from the more trustworthy category and the red + are intensities from the less trustworthy category.

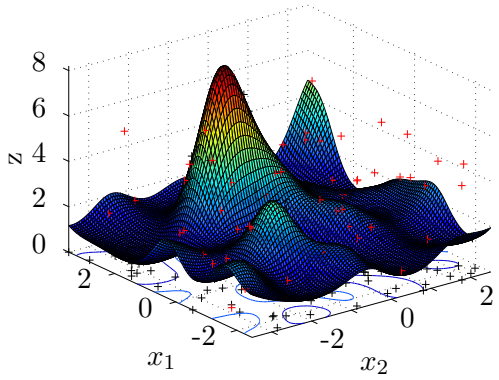


(a)

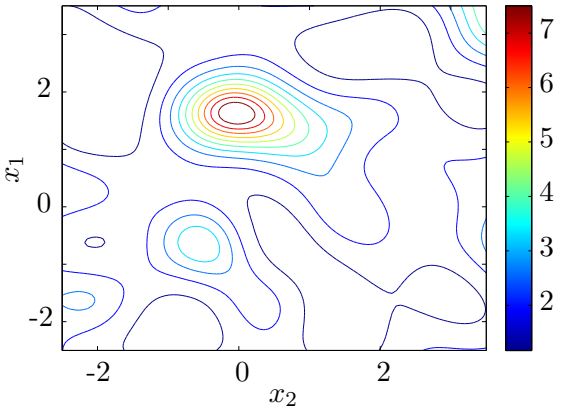


(b)

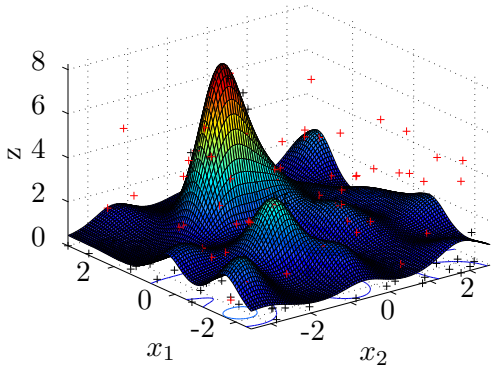
Figure 6.3: Performance of the tested methods on synthetic data as measured by the CRPS (a) and the RMSE (a).



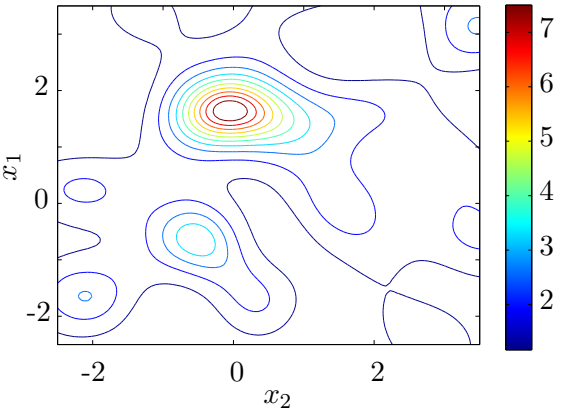
(a) LGCP 3D plot



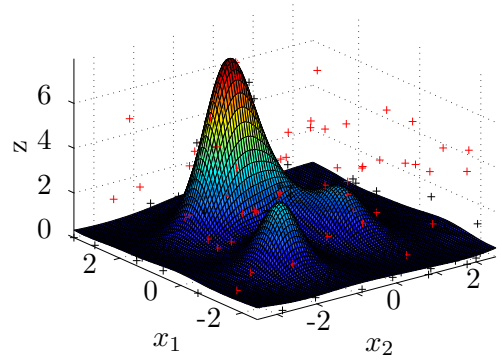
(b) LGCP contour



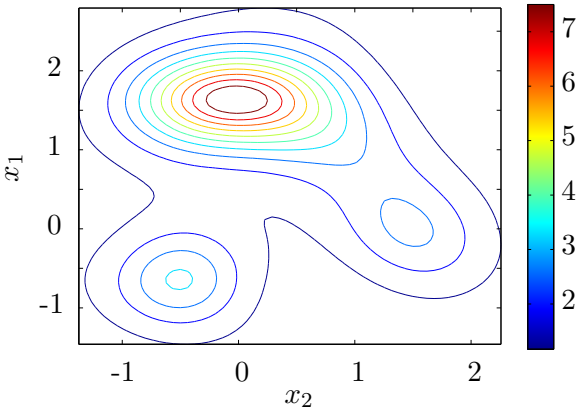
(c) TrustLGCP 3D plot



(d) TrustLGCP contour



(e) Optimal LGCP 3D plot



(f) Optimal LGCP contour

Figure 6.4: Plots of the spatial intensities estimated by the LGCP (a and b) the TrustLGCP (c, d) and the Optimal LGCP (e, f) from the synthetic dataset of Figure 6.2 (b).

	RMSE	CRPS
LGCP	1.11	0.77
TrustLGCP	0.89	0.61
Optimal LGCP	0.40	0.36

Table 6.1: Errors of the three LGCP methods in one test on the example synthetic dataset of Figure 6.2 (b).

points located in the cell, i.e., the intensity of X_i, as $\phi(X_i) \sim Poisson(\exp(\hat{z}_i))$ where $\hat{z}_i$ is the true value of the intensity given by the peaks function. Thirdly, similarly to our previous experiments, we want to simulate a percentage ρ of untrustworthy intensities to challenge the models to correctly learn the trust distribution of such intensities. To do so, we define a random assignment of $\rho\%$ of bins that contains points from the less trustworthy category 2 and we consider the remaining $(1-\rho)\%$ of intensities to be of category 1. Then, we introduce a random bias over $\phi(X_i)$ set proportionally to the categorical trust value of X_i. That is:

$$\gamma(X_i) = \phi(X_i) + (1 - t_i) * \exp(w) \qquad w \sim U[0, 5] \tag{6.10}$$

In this setting, we test the accuracy of our method in making predictions of the true intensity function with data generated under different values of ρ.

In more detail, Figure 6.3 (a) shows the CRPS of the three methods for 100 runs with $\rho = \{10, 20, 30, 40, 50, 60\}$. The graph shows that the TrustLGCP improves the performance of the standard LGCP in all configurations. In particular, its CRPS is 10% lower than LGCP with $\rho = 50\%$ and it is generally better than the LGCP for any ρ. This result shows that our method improves the predictions by correctly assigning low trustworthiness to less reliable intensities. Moreover, having the highest accuracy gain for high ρ means that the trust learning of the TrustLGCP is particularly helpful in the more challenging settings in which the data contain a higher number of untrustworthy intensities.

Furthermore, Figure 6.3 (b) shows the RMSE of the methods in the same experiment. In particular, it shows that the TrustLGCP also outperforms the LGCP in terms of absolute error. This means that the accuracy gain of our method is mostly given by a lower error in the predictive mean intensity rather than its predictive uncertainty. To illustrate this, Figure 6.4 shows the plots of the predictions of each method on the same dataset of Figure 6.2 (b), in which $\rho = 30\%$. From these plots, we notice that the predictions of the LGCP and the TrustLGCP are similar in their main structures, i.e., they both identify two peaks of the true function at $x_1 = -0.45, x_2 = -0.4$ and $x_1 = 1.7, x_2 = 0$. However, the improvement of the TrustLGCP can be seen, for example, in the prediction of $x_1 = 3, x_2 = 3$ where the LGCP incorrectly predicts an intensity of value 5, while the TrustLGCP predicts an intensity of value 2.2 against a true value of

Category	**# reports**	**Estimated t**
Emergency	332	1
Vital lines	1724	0.45
Public health	40	1
Security threats	66	0.45
Infrastructure damage	84	0.56
Natural hazards	5	1
Services available	360	0.78
Other	163	0.22

Table 6.2: The description of the Ushahidi–Haiti dataset with the eight emergency categories, the number of reports in each category, ant the estimated trustworthiness of each category as computed by the TrustLGCP.

approximately zero. In fact, the assignment of lower trustworthiness to the category of red points helps our model reduce their noise in the prediction.

Finally, the Optimal LGCP provides the best possible learning of the true function by completely filtering all the untrustworthy points. Furthermore, the RMSE and the CRPS of all the methods tested on this example dataset are reported in Table 6.1, which show that the TrustLGCP errors are significantly lower than the LGCP. Thus, our experiment on synthetic data demonstrates that our TrustLGCP provides intensity predictions that are closer to the optimum.

6.3.3 Experiment on Real Data

In this second experiment, we apply the TrustLGCP to the Ushahidi–Haiti dataset that was described in Section 6.3.3.1. Subsequently, we discuss the results of our evaluation in Section 6.3.3.2.

6.3.3.1 Dataset

The Ushahidi–Haiti dataset consists of a set of 2774 emergency reports that were submitted by Haitian people by emails, SMS and tweets during the crisis of the 2010 earthquake. The reports were collected between the 12 January 2010 (the same day of the main quake) and the 1 August 2010 and were classified among the eight emergency macro–categories of the Ushaidi–Haiti deployment that are reported in Table 6.2[4]. Specifically, each report provides the location, the timestamp, the category and the text content that describes the event as reported by the user. Figure 6.5 shows the map of all the reports that covers a geographical area of 78,517 km^2 (Boundaries: min. longitude =

[4]The categories reported in the example of Figure 6.1 are sub–groups of the eight Ushahidi macro–categories

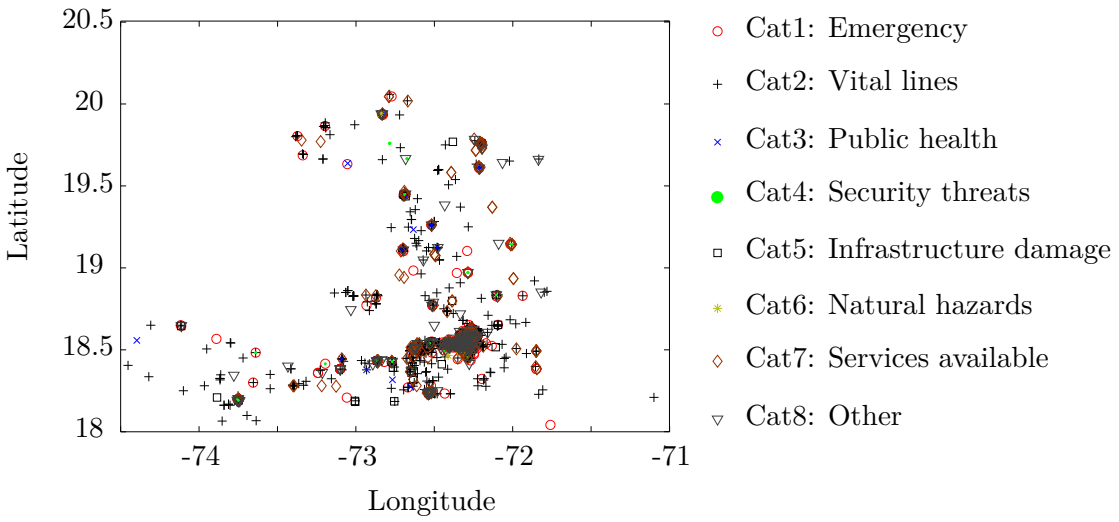
-74.453, max. longitude $= -71.099$, min. latitude $= 18.041$, max. latitude $= 20.058$). A complete description of this dataset can be found in Morrow et al. (2011).

We are interested in learning the intensity maps of these reports over the entire area using our model. To do so, we first project the points to the Cartesian coordinates (in meters) using the Universal Transverse Mercator (UTM) projection with a reference zone of 18Q (Snyder, 1987). Then, we partition the space into a regular grid of 200 bins along the two axis and compute the point counts of each cell. Therefore, each count provides the empirical intensity observed in the cell.

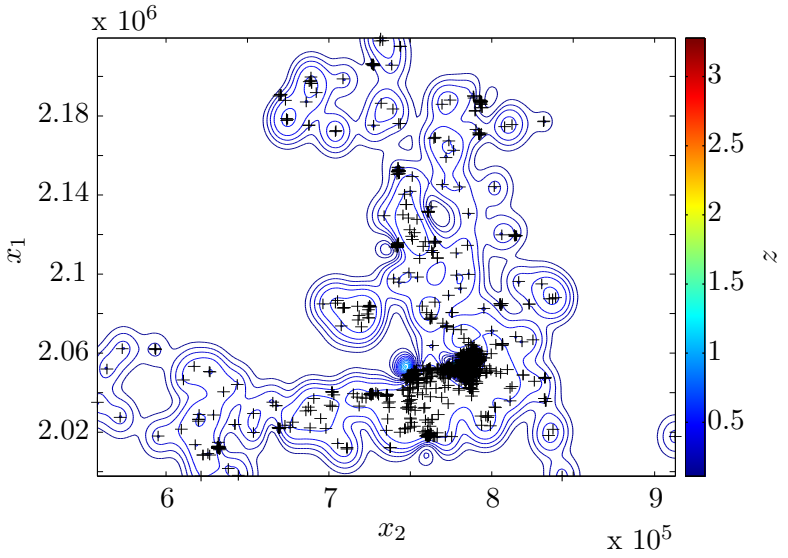
6.3.3.2 Results

Figure 6.5 shows the contour plots of the predictions of the two LGCP methods computed on the Ushahidi–Haiti dataset. In particular, it can be noticed that the LGCP concentrates its prediction mainly around one location, i.e., close the capital city of Port–au–Prince, where the peak of reports is observed. As a counter effect, it is readily apparent that the LGCP does not provide good predictions in other parts of the map where its estimated intensities are generally low. Importantly, using the same kernel hyperparameters as the LGCP, the TrustLGCP is able to provide much more granular predictions. In particular, it identifies several areas where higher intensities with respect to other areas are predicted, i.e. the red contours (or red areas). As a result, the TrustLGCP predictions appears to better highlight areas with high intensities from the rest of the map, which are more helpful in localising damaged areas. Also, we notice that the TrustLGCP avoids the issue of predicting a very high peak of intensities in only one point as in the case of the LGCP. This benefit comes from the fact that the learned trust values provide a better smoothing of the intensity function. This makes the model more robust to avoid suspicious peaks of intensities observed in the reports. In particular, the estimated trust values of each category, which are reported in Table 6.2, reveal that the vital lines category that has the largest number of reports (37%) is only 0.45 trustworthy. This means that the reports in this class tend to be less indicative of the intensities outlined by the reports of the other categories. In general, the TrustLGCP predicts an average intensity of $\lambda = 1.5$ in the red areas and, by integrating this intensity over the size of such an area, it is possible to find the number of reports which are expected to be submitted from that region.

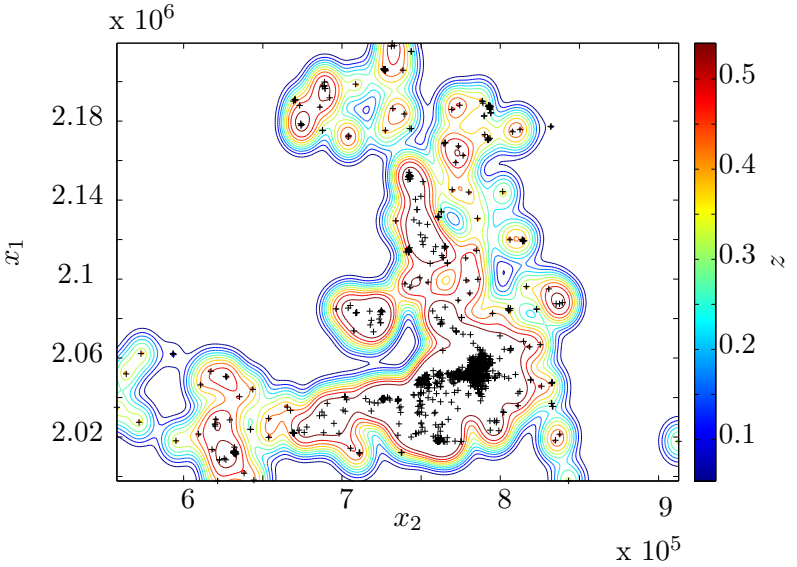
Validation with Ground Truth Data A good strategy to validate the learning outputs of our model would be to compare its predictions to the actual trustworthiness of the crowd reports. Unfortunately, assessing the trustworthiness of all the reports collected in Haiti turned out to be infeasible as it required the in–situ verification of each reported event (the Ushahidi team was able to verify only 1% of all the reports). Alternatively, a more feasible strategy would be to compare the predicted intensities to



(a) The Ushahidi–Haiti dataset



(b) The LGCP contour



(c) The TrustLGCP contour

Figure 6.5: Predictions of the LGCP (a) and the TrustLGCP (b) computed on the Ushahidi–Haiti dataset.

the ground truth of the damage levels in the area. This ground truth data about such damage levels can be collected by experts who are involved in assessing the actual damages in the disaster area. Despite the temporal discrepancy that such ground truth data might have compared the crowdsourced reports, which are available immediately after the disaster, one can measure the correlations between the spatial intensities predicted by the spatial point process models on the two data sources More specifically, assuming that ground truth data do not require any special trust–based treatments and can be handled by a normal LGCP, we can possibly compare their intensities to the predictive output of the TrustLGCP versus the standard LGCP computed on the crowdsourced reports. Unfortunately, we do not have such ground truth data to make this comparison. Therefore, we can only provide an intuitive validation based on the fact that our model seems to make more sensible decisions about avoiding peaks of intensities from the LGCP predictions, using the same kernel hyperparameters. Thus, despite the effort that we made to provide this new model for crowdsourced spatial point processes, these conclusions should be taken with caution given the lack of a full validation with ground truth data.

6.4 Summary

In this chapter, we addressed our fourth and last requirement concerning the fusion of untrustworthy spatial point data in crowdsourcing. Specifically, we considered the disaster response scenario where the crowd provides a number of geo–tagged emergency reports that are categorised by their emergency type. In particular, we focused on the task of estimating the intensity map that predicts the expected number of reports over the entire area, which provides a key insight about the disaster scene. To address this problem, we extended our trust–based fusion approach to the class of spatial point process models. In particular, we used a LGCP model as a basis to define our new trust–based LGCP that advances the original model by considering varying trustworthiness over categories of points. In this way, our TrustLGCP is able to simultaneously learn the spatial intensities of a random point process and also the trustworthiness of each data category. In particular, our model assumes that the majority of trustworthy points are representative samples of the true intensity function. Then, it learns truthfulness of the reports' categories on the basis of their fit to the underlying function and refines its intensity predictions accordingly. By testing our model on synthetic data, we showed that it improves accuracy by 10% against the non–trust LGCP models as a result of an effective trust learning that enables our model to filter the noise of untrustworthy categories of data. Furthermore, by applying our TrustLGCP to the Ushahidi–Haiti dataset, we intuitively observed that its intensity maps are more plausible to identify the real disaster areas since they are able to effectively avoid the spikes of the observed point

intensities. Furthermore, it also provides useful information about the trustworthiness of each category that can form a basis for emergency planning using crowdsourced reports.

In general, our model provides a solution to many problems that involve reasoning over point patterns under the uncertainty of the data reliability. For example, other related applications can be thoughts as predicting intensity of taxi calls from the requests of taxi users or the spatial distribution of failures in a distributed public service from the reports of the customers. In this respect, other research directions to potentially integrate the requirements of these domains in our work are discussed in the next chapter.

Chapter 7

Conclusions

In this thesis, we developed a novel trust-based approach for making reliable inference over crowdsourced information in participatory sensing applications. Specifically, we focused on the key challenges of computing reliable aggregations of crowdsourced estimates in stationary settings, i.e., learning a fixed value, and non–stationary settings, i.e., learning spatial functions. The motivation behind this research is that the data collected through crowdsourcing systems is often untrustworthy because it is provided by unreliable human sources who may be inaccurate, uncommitted to the task and/or can also strategise which information to report. In particular, we studied this problem of aggregating crowd reports representing continuous estimates collected through mobile phone–based crowdsourced sensors. To tackle this problem, we proposed an agent–based architecture in which an output mediator software agent is dedicated to the tasks of fusing multiple reports and inferring latent trust features of the users (see Section 1.1). The key requirement for the design of such an agent are the abilities to (i) compute probabilistic aggregations of multiple crowd reports, (ii) reason about the trustworthiness of the individual users, (iii) analyse spatial–temporal trends and (iiii) analysing point patterns in the reports of the crowd. In this context, the major challenge was outlined as producing good quality aggregations in the presence of untrustworthy users whose reports complicate the task of computing reliable data aggregation. Given this, we considered the approach of modelling the unobserved user's trustworthiness as a key element of our fusion framework designed for crowdsourcing systems. Based on this framework, we developed a new family of trust–based algorithms for fusing crowdsourced continuous data while simultaneously learning the trustworthiness of individuals.

In more detail, in Chapter 2, the relevant literature in the field of machine learning for reliable crowdsourcing systems, including information fusion, spatial regression and point process models, was reviewed. Emphasis was given to the class of models that focus on the joint learning of the user's trustworthiness and the aggregated value from crowdsourced datasets. However, the main limitation of the existing models lies in the

fact that they do not consider the' reported precision of the users as part of the input data. This is a problem because crowdsourced sensor estimates typically include a precision value that quantifies the uncertainty by the precision of the user reported for its observation. As such, crowdsourcing must permit solutions in which inference takes such reported uncertainties into account in a crowdsourced dataset. In addition, reviewing models from the fiends of probabilistic data fusion, we discussed a number of methods that deal with the fusion of continuous estimates in the single-hypothesis and the multiple-hypothesis setting (see Section 2.3.2). 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 valid basis for the design of one of our trust-based fusion methods due to its property of reducing noise in Gaussian estimates. Moreover, to address the requirement of identifying untrustworthy reports within a crowdsourced dataset, we discussed a density-based outlier detection method and sensor fusion algorithms for untrustworthy sensors (see Section 2.4.2). In particular, the local outlier factor (LOF) and the Reece et al.'s algorithm (RM) 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 regression with crowdsourced data when presenting the Gaussian process (GP) model (see Section 2.5). In particular, we considered the class of heteroskedastic Gaussian processes (HGP) to handle spatial data with constant noise variance. In fact, these models are suitable for learning continuous functions from noisy observations as our problem requires. However, the standard HGP regression model does not provide any support against estimates of varying trustworthiness as defined by crowdsourced estimates. To address this shortcoming, we developed a new trust-based HGP model (TrustHGCP) for spatial regression that considers different levels of trustworthiness over the observed spatial estimate.

Finally, we discussed the class of Bayesian non–parametric models for learning for spatial point processes from crowdsourced data (see Section 2.6). Specifically, these models are designed for learning point patterns by means of an intensity function that maps a continuous space into a number of points that are expected to be observed at each region. In particular, we described the theory of the Log Gaussian Cox Process (LGCP) that applies a GP treatment of the intensity function of a spatial Cox point process. Similarly to the limitation of the GP regression model, also the LGCP does not deal with the untrustworthy estimates as it is necessary within a crowdsourcing domain. Therefore, using our trust–based approach, we designed a new LGCP model that incorporates the ability of modelling categories of points with different trustworthiness within a point process learning model.

7.1 Summary of Results

The results of the work presented in this thesis were detailed in four chapters. Specifically, Chapter 3 presented our trust-based model for fusing crowdsourced estimates for stationary quantities, within the application of crowdsourced cell tower localisation. The key feature of our model is to represent the user's trustworthiness as a scaling parameter of the reported precisions in a way that untrustworthy estimates are turned into uninformative contributions to the fusion process. Then, we developed an efficient algorithm (MaxTrust) to simultaneously learn the true value of the observed item and the trustworthiness of each user from the crowd reports. Using the OpenSignal–cell tower dataset containing cell tower estimates provided by Android phones, we showed that MaxTrust improves the accuracy by 21% compared to established approaches from data fusion and sensor networks. Furthermore, we showed that it also achieves comparable accuracy with 10% more untrustworthy users through experiments on synthetic data. In fact, our uncertainty scaling trust model provides more flexibility, and in turn better accuracy, compared to the evaluated threshold–based trust modelling approaches.

In Chapter 4, we improved MaxTrust in situations where the user's trustworthiness is correlated to observations of a set of items. Specifically, we designed a Bayesian trust–based fusion model (BACE) that addresses the same problem as MaxTrust, i.e., aggregate crowdsourced estimates of stationary items, by introducing the following two refinements. Firstly, the model uses trust parameters that are shared among observations of different items. By so doing, it incorporates the ability to transfer the learned user's trustworthiness to inform inference over new items or items with only a few reports. Secondly, the model accepts prior distributions over the trust parameters and the items' true value that allows it to exploit existing knowledge over these random variables to achieve better inference. We showed that BACE is more effective than MaxTrust, and the a number of benchmarks, in an evaluation on crowdsourcing WiFi hotspots. In particular, due to its efficient exploitation of correlations in users' trust behaviours in observing multiple items, it is able to improve accuracy by up to 45% on real data and be more robust to untrustworthy crowds by 15% on synthetic data.

Furthermore, in Chapter 5, we addressed the problem of learning non–stationary quantities such as continuous functions and environmental phenomena from crowdsourced data. In particular, we detailed our trust-based heteroskedastic Gaussian process model to perform regression over crowdsourced estimates. This model integrates the trust approach underpinning MaxTrust within the principled Bayesian inference framework of heteroskedastic Gaussian process models. Then, by training the trust parameters on the reports gathered from the crowd, the model is able to estimate the trustworthiness of each user as well as the spatial function describing the phenomena observed by the crowd. Experiments on synthetic data show that our method outperforms the non-trust GP methods, improving accuracy by up to 34% and reducing uncertainty by up to 80%.

Moreover, using real–world radiation data collected from the 2011 Fukushima nuclear disaster provided by crowdsourced sensors, our method outperforms the benchmarks in making more accurate predictions, by 13%, and with significantly lower uncertainty, by 89% of nuclear radiation levels.

Finally, in Chapter 5, we further extended our trust–based approach for modelling spatial data. In particular, we tackled the problem of learning spatial patterns from crowdsourced point reports. Our solution was the TrustLGCP model that applies the trust modelling approach to deal with categories of untrustworthy points reported from the crowd in learning spatial intensity functions. This model is able to simultaneously learn the spatial intensities of the reported points, as well as the trustworthiness of each point type (i.e., category). In particular, our experimental validation within the scenario of the Haiti earthquake showed that our model can efficiently learn the intensity map of crowdsourced emergency reports as well as estimate the trustworthiness of each emergency category. Furthermore, we showed that the TrustLGCP is 10% more accurate than the standard LGCP in learning spatial intensities on synthetic data.

When taken together, these results make a significant contribution to computing reliable aggregations of data generated within crowdsourcing and participatory sensing applications.

7.1.1 Impact of our Results

Apart from crowdsourcing, there are a number of application areas that can take advantage from the trust–based information fusion approach developed in this thesis. For example, a central problem in recommendation systems is how to aggregate multiple judgments that the users provide while rating a number of items (Stern et al., 2009). Similarly, automated peer reviewing systems deal with inferring a true score of a submission (e.g., a conference paper or student's coursework) based on the subjective continuous scores provided by the reviewers (Flach et al., 2010). All these systems, and many others besides, face the problem of making decisions based on subjective inputs provided by human–generated estimates, and therefore they are suitable applications for our methods.

7.1.2 Limitations

Although we developed a number of methods for aggregating crowdsourced estimates by learning the trustworthiness of the users, there are still some open issues for which we do not provide a solution. In particular, we identify the following three limitations:

- **Uncertainty over Trustworthiness in the Spatial Models.** Both the TrustGP and the TrustLGCP assess the trustworthiness of the users and categories of reports as a scalar value. In general, it might required to provide confidence intervals around such estimated trust values. To do so, the method would require to consider the uncertainty over the trust parameters but this would also require an extra level of complexity, which is not currently present in our models.

- **Learning without Gold Standards.** Though our methods, we provided solutions to make inference over the aggregated output by only looking at the crowd reports, without ever observing the true value of the estimated output. In some cases, it might be desirable for these methods to be able to to integrate gold standards in their learning process. This would allow them to reduce the uncertainty in the data fusion process but it also requires extra mechanisms to adapt our current models to semi–supervised learning settings.

- **Batch Learning.** All our methods are designed to train in batch over a set of estimates that reported offline by the crowd. Although this fits the requirement of our applications, other situations might require to train the methods online over estimates that are reported sequentially. This use case is is not currently available in our methods.

In addition, there are a number of ways in which our work could be extended, to better support data collection and scalable inference in such applications. We examine a number of possible promising directions for future research in the following section.

7.2 Future Work

As crowdsourcing continues to expand across many applications of Artificial Intelligence (AI), trust–based fusion algorithms will need to accommodate increasingly larger amounts of data and seek more efficient ways of data collection. Moreover, since the same users could participate in different tasks proposed by various crowdsourcing projects, a key question is to look at combining signals of user's trustworthiness emerging from various tasks. Therefore, we identify the following areas in which further research is warranted to extend the scope of our work.

- **Trust–Based Active Learning**
  An important extension of our trust–based fusion algorithms would be to incorporate the ability of reasoning about which user could provide the best contribution to our learning system by providing a new estimate. In particular, this topic has been extensively studied in the machine learning community in the area of Active Learning (AL) problems. Specifically, the AL paradigm focuses identifying the

next task that a learning system wants to be observed by a human subject. While AL seems to be a natural fit to crowdsourcing, there is an important element which is not considered. In crowdsourcing, instead of just selecting a particular item to be observed, an AL algorithm could also choose the most trustworthy users that is particularly appropriate to take such an observation. However, while this additional flexibility seems to be good for getting informative estimates, there are computational and practical challenges. In fact, the search of the best item–worker pair becomes computationally unfeasible as it involves searching over the product space of users and items. Furthermore, the traditional AL approaches based on requesting observations by a single best worker might loose the advantage of parallelising crowdsourcing tasks. Therefore, methods to address the AL problem in crowdsourcing would require solutions to both these challenges.

- **Community–Based Trust Models**
  To deal with the increasing scale of crowdsourcing systems, community based trust models (Falcone and Castelfranchi, 2008) could inspire ways to deal with the large volume of data in trust–based fusion algorithms. Specifically, the key feature of such community models is to reason about groups of workers that naturally form within the crowd based on some common reporting behaviours. In this context, the communities define types of users (e.g., accurate users, conservative users, biased users, etc.) that approximate the structure of an arbitrarily large group of users. Given these communities, probabilistic inference could benefit from hierarchically learning community profiles and using them to reduce the number of users to reason about, i.e., communities can be used to identify representative users within a large crowd. However, an open question is how to design effective community models that can be learned from large datasets of continuous estimates and meet the requirements of our trust–based fusion approach.

- **Multi–Task Crowdsourcing**
  As mentioned earlier, crowdsourcing is likely to move towards the reality of having the same users who participate in many outsourced tasks. For example, each user could take part in jobs such as classification, weather sensing, image tagging, cell–tower mapping, etc. As a result, crowdsourcing platforms will be able to observe cross–task user signals such as the average time spent on producing data, their accuracies and other features. When taken together, these signals could reveal informative patterns that would allow a system to transfer the user's trustworthiness from existing tasks to new ones. Thus, a key direction is to investigate extensions to our current techniques to perform trust–based fusions of heterogeneous types of data (i.e., discrete labels, continuous estimates, etc.) that are produced from these tasks.

Appendix A

Approximate Continuous Rank Probability Score for Sampled Distributions

This appendix provides the details of our approximate computation of the continuous rank probability score (CRPS) (Kohonen and Suomela, 2006) for univariate sampled probability distributions that we used to evaluate the performance of all the trust-based algorithms presented in this thesis.

In more detail, the CRPS is a rank probability score that compares a distribution over μ_i to the point mass distribution centred on the true value μ^*. Its calculation is given by the finite integral of the square difference between the cumulative distribution function (c.d.f) of μ_i, $F_i(\mu)$ and the *Heaviside function* of μ_i^*, $H_i(\mu)$ i.e. the single step-function centered on μ_i^*:

$$\text{CRPS} = \int_{-\infty}^{+\infty} (F_i(\mu) - H_i(\mu))^2 d\mu$$

Using the definition of the Heaviside function, this integral divides into two terms:

$$\text{CRPS} = \int_{-\infty}^{\mu_i^*} F_i(\mu)^2 d\mu + \int_{\mu_i^*}^{+\infty} (1 - F_i(\mu))^2 d\mu$$

Now, assume we are given N samples of $F_i(\mu)$. Let us sort the samples in ascending order $\mu'_1, \ldots, \mu'_N$ and approximate $F(\mu)$ as constant within two adjacent samples. Then, $F_i(\mu)$ can be written as:

$$F_i(\mu) = \begin{cases} 0 & \mu < x_1 \\ s/N & x_s \leq \mu < x_{s+1} \\ 1 & \mu \geq x_N \end{cases}$$

Therefore the CRPS for the sampled $F(\mu)$ is given by the following sum:

$$\begin{aligned}\mathrm{CRPS} = &\sum_{s=1}^{s'-1} \left(\frac{s}{N}\right)^2 (\mu'_{s+1} - \mu'_s) \\ &+ \sum_{s=s'}^{N-1} \left(1 - \frac{s}{N}\right)^2 (\mu'_{s+1} - \mu'_s)\end{aligned}$$

where s' is the index of the greatest sample smaller than μ_i^*.

Appendix B

The OpenSignal–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 B.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.

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 Maritime 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 B.1: The number of reports for each network operator, device types, network types and location sources.

- **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 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

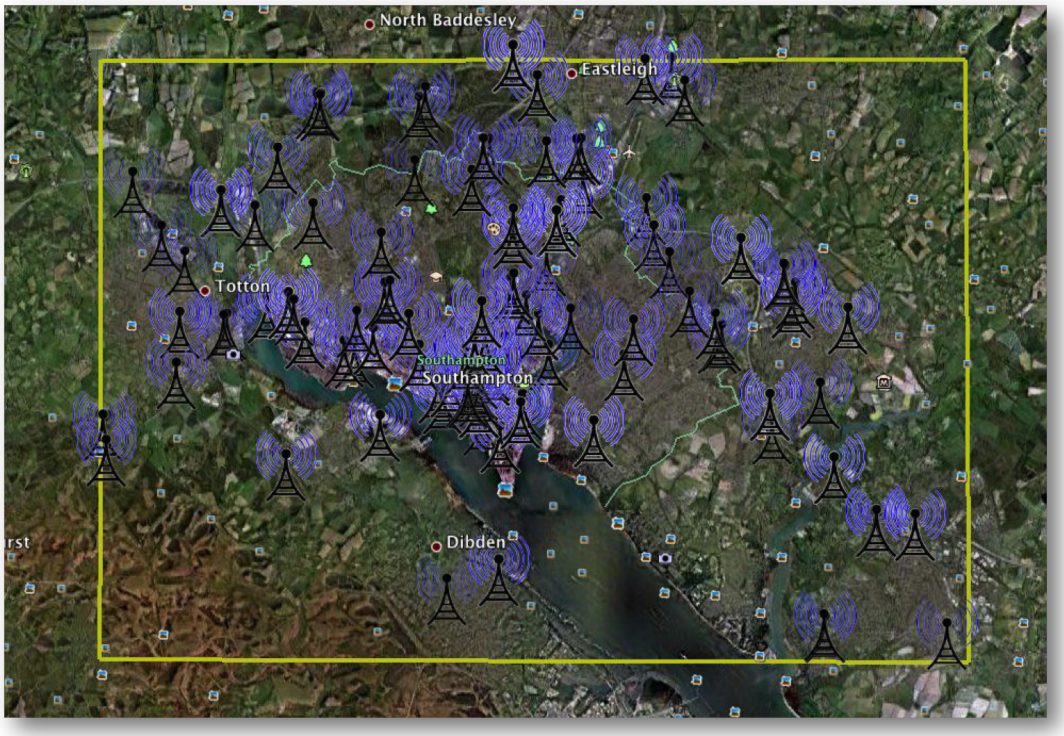
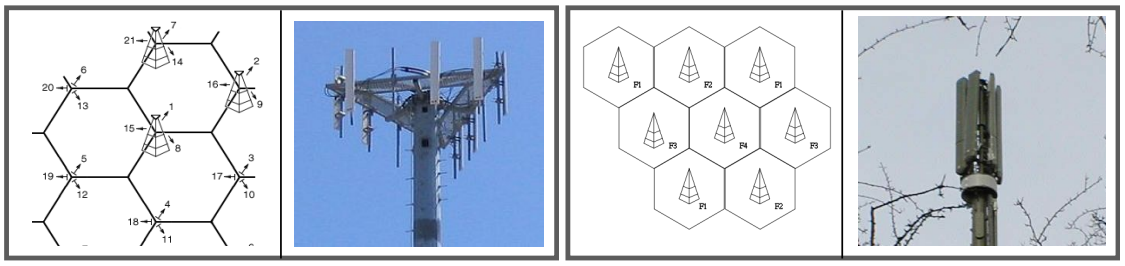


Figure B.1: Screenshot showing the bounding box of the Southampton, UK area and the location of the masts (based on the cell_lat and cell_lon fields) tagged within the OpenSignalMaps dataset.

reduced by 66% after filtering, see Table B.1 which shows the 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.

Furthermore, the reports tagged a total of 2291 base stations whose locations are shown in Figure B.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 B.2. In an omni-directional cellular network, the land area is divided into regular hexagonal cell. 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

[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) (b)

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

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 three 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 C

Xively Radiation Dataset

This appendix describes the radiation dataset provided by the Xively 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 Xively 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 Xively.
- **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 percentages of feeds for each of these categories found in this dataset is showed in the pie chart in Figure C.1.

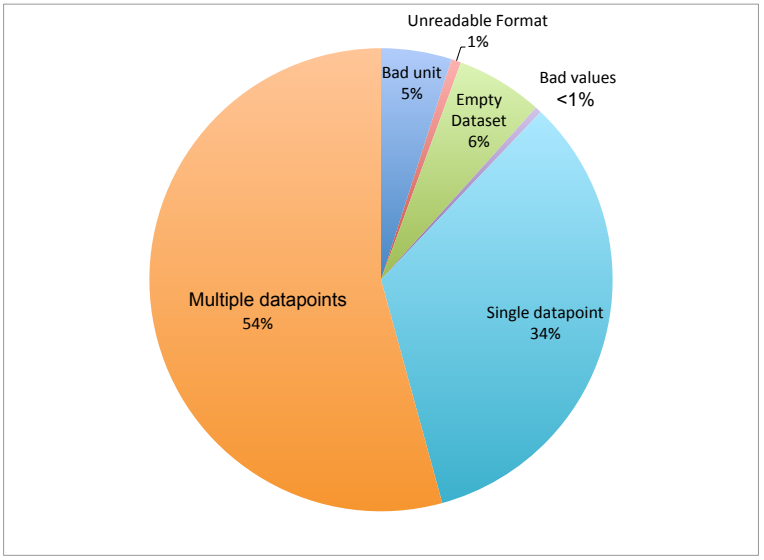


Figure C.1: Pie chart of the Xively dataset

An example sample of seven feeds taken from the Xively 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>
```

Appendix D

Details on Other Publications Written During PhD

In this appendix we provide the abstracts of the publications written by M. Venanzi during the PhD, three of them as the first author, which are not described as chapters of this thesis.

M. Venanzi, J. Guiver, G. Kazai, P. Kohli, M. Shokouhi. Community-Based Bayesian Aggregation Models for Crowdsourcing. *In the 23rd International World Wide Web Conference (WWW)*, 2014. *Best Paper Runner–up.* Microsoft, one of the partners of the ORCHID project (`www.orchid.ac.uk`), has registered the algorithm presented in this paper under a US patent. MS ref: 340522.01.

Abstract: This paper addresses the problem of extracting accurate labels from crowdsourced datasets, a key challenge in crowdsourcing. Prior work has focused on modelling the reliability of individual workers, for instance, by way of confusion matrices, and using these latent traits to estimate the true labels more accurately. However, this strategy becomes ineffective when there are too few labels per worker to reliably estimate their quality. To mitigate this issue, we propose a novel community-based Bayesian label aggregation model, CommunityBCC, which assumes that crowd workers conform to a few different types, where each type represents a group of workers with similar confusion matrices. We assume that each worker belongs to a certain community, where the workerÕs confusion matrix is similar to (a perturbation of) the community's confusion matrix. Our model can then learn a set of key latent features: (i) the confusion matrix of each community, (ii) the community membership of each user, and (iii) the aggregated label of each item. We compare the performance of our model against established aggregation methods on a number of large-scale, real-world crowdsourcing datasets. Our experimental results show that our CommunityBCC model consistently outperforms state–of–the–art label aggregation methods, gaining, on average, 8% more accuracy with the same amount of labels.

L. Tran-Thanh, M. Venanzi, A. Rogers, N.R. Jennings (2013) Efficient Budget Allocation with Accuracy Guarantees for Crowdsourcing Classification Tasks. *In the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2013, 901-908.

Abstract: In this paper we address the problem of budget allocation for redundantly crowdsourcing a set of classification tasks where a key challenge is to find a tradeÐoff between the total cost and the accuracy of estimation. We propose CrowdBudget, an agentÐbased budget allocation algorithm, that efficiently divides a given budget among different tasks in order to achieve low estimation error. In particular, we prove that CrowdBudget can achieve at most $\max\{0, K/2 - O(\sqrt{B})\}$ estimation error with high probability, where K is the number of tasks and B is the budget size. This result significantly outperforms the current best theoretical guarantee from Karger *et al.* In addition, we demonstrate that our algorithm outperforms existing methods by up to 40% in experiments based on real–world data from a prominent database of crowdsourced classification responses.

S. Ramchurn, T. D. Huynh, M. Venanzi, B. Shi. Collabmap: Crowdsourcing Maps for Emergency Planning. *In the 5th Annual ACM Web Science Conference*, 4, (2), 2013, 326-335.

Abstract: In this paper, we present a software tool to help emergency planners at Hampshire County Council in the UK to create maps for high-fidelity crowd simulations that require evacuation routes from buildings to roads. The main feature of the system is a crowdsourcing mechanism that breaks down the problem of creating evacuation routes into micro–tasks that a contributor to the platform can execute in less than a minute. As part of the mechanism we developed a consensus–based trust mechanism that filters out incorrect contributions and ensures that the individual tasks are complete and correct. To drive people to contribute to the platform, we experimented with different incentive mechanisms and applied these over different time scales, the aim being to evaluate what incentives work with different types of crowds, including anonymous contributors from Amazon Mechanical Turk. The results of the 'in the wild' deployment of the system show that the system is effective at engaging contributors to perform tasks correctly and that users respond to incentives in different ways. More specifically, we show that purely social motives are not good enough to attract a large number of contributors and that contributors are averse to the uncertainty in winning rewards. When taken altogether, our results suggest that a combination of incentives may be the best approach to harnessing the maximum number of resources to get socially valuable tasks (such for planning applications) performed on a large scale.

A. Rutherford, M. Cebrian, I. Rahwan, S. Dsouza, J. McInerney, V. Naroditskiy, M. Venanzi, N. R. Jennings, J.R. deLara, E. Wahlstedt, S. U. Miller. Targeted Social Mobilization in a Global Manhunt. *PLoS ONE*, 2013, 8(9): e74628.

Abstract: Social mobilisation, the ability to mobilise large numbers of people via social networks to achieve highly distributed tasks, has received significant attention in recent times. This growing capability, facilitated by modern communication technology is highly relevant to endeavours which require the search for individuals that possess rare information or skills, such as finding medical doctors during disasters, or searching for missing people. An open question remains, as to whether in time-critical situations, people are able to recruit in a targeted manner, or whether they resort to so-called blind search, recruiting as many acquaintances as possible via broadcast communication. To explore this question, we examine data from our recent success in the U.S. State Department's Tag Challenge, which required locating and photographing 5 target persons in 5 different cities in the United States and Europe, in under 12 hours, based only on a single mug-shot. We find that people are able to consistently route information in a targeted fashion even under increasing time pressure. We derive an analytical model for social–media fueled global mobilisation and use it to quantify the extent to which people were targeting their peers during recruitment. Our model estimates that approximately 1 in 3 messages were of targeted fashion during the most time–sensitive period of the challenge. This is a novel observation at such short temporal scales, and calls for opportunities for devising viral incentive schemes that provide distance or time-sensitive rewards to approach the target geography more rapidly. This observation of "12 hours of separation" between individuals has applications in multiple areas from emergency preparedness, to political mobilisation.

H. T. Dong, M. Ebden, M. Venanzi, S. Ramchurn, S. Roberts, L. Moreau. Interpretation of Crowdsourced Activities Using Provenance Network Analysis. *In the 1st International Conference on Human Computation and Crowdsourcing (HCOMP)*, 2013, 78-85.

Abstract: Understanding the dynamics of a crowdsourcing application and controlling the quality of the data it generates is challenging, partly due to the lack of tools to do so. Provenance is a domain-independent means to represent what happened in an application, which can help verify data and infer their quality. It can also reveal the processes that led to a data item and the interactions of contributors with it. Provenance patterns can manifest real-world phenomena such as a significant interest in a piece of content, providing an indication of its quality, or even issues such as undesirable interactions within a group of contributors. This paper presents an application-independent methodology for analysing provenance graphs, constructed from provenance records, to learn about such patterns and to use them for assessing some key properties of crowdsourced data, such as their quality, in an automated manner. Validating this method on the provenance records of CollabMap, an online crowdsourcing mapping application, we demonstrated an accuracy level of over 95% for the trust classification of data generated by the crowd therein.

V. Capraro, M. Venanzi, M. Polukarov, N.R. Jennings. Cooperative Equilibria in Iterated Social Dilemmas. *In, 6th International Symposium on Algorithmic Game Theory*

(SAGT), 2013, 146-158.

Abstract: The implausibility of the extreme rationality assumptions of Nash equilibrium has been attested by numerous experimental studies with human players. In particular, the fundamental social dilemmas such as the TravelerÕs dilemma, the PrisonerÕs dilemma, and the Public Goods game demonstrate high rates of deviation from the unique Nash equilibrium, dependent on the game parameters or the environment in which the game is played. These results inspired several attempts to develop suitable solution concepts to more accurately explain human behaviour. In this line, the recently proposed notion of cooperative equilibrium based on the idea that players have a natural attitude to cooperation, has shown promising results for single-shot games. In this paper, we extend this approach to iterated settings. Specifically, we define the Iterated Cooperative Equilibrium (ICE) and show it makes statistically precise predictions of population average behaviour in the aforementioned domains. Importantly, the definition of ICE does not involve any free parameters, and so it is fully predictive.

M. Venanzi, M. Piunti, R. Falcone, C. Castelfranchi. Facing Openness with Socio Cognitive Trust and Categories. *In the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2011, 400-405.

Abstract: Typical solutions for agents assessing trust relies on the circulation of information on the individual level, i.e. reputational images, subjective experiences, statistical analysis, etc. This work presents an alternative approach, inspired to the cognitive heuristics enabling humans to reason at a categorial level. The approach is envisaged as a crucial ability for agents in order to: (1) estimate trustworthiness of unknown trustees based on an ascribed membership to categories; (2) learn a series of emergent relations between trustees observable properties and their effective abilities to fulfil tasks in situated conditions. On such a basis, categorization is provided to recognize signs (Manifesta) through which hidden capabilities (Kripta) can be inferred. Learning is provided to refine reasoning attitudes needed to ascribe tasks to categories. A series of architectures combining categorization abilities, individual experiences and context awareness are evaluated and compared in simulated experiments.

R. Falcone, M. Piunti, M. Venanzi, C. Castelfranchi, From manifesta to krypta: The relevance of categories for trusting others. *ACM Transactions on Intelligent Systems and Technology (TIST), special issue on Trust in Multi-Agent Systems*, 2011, 1-24.

Abstract: In this paper we consider the special abilities needed by agents for assessing trust based on inference and reasoning. We analyze the case in which it is possible to infer trust towards unknown counterparts by reasoning on abstract classes or categories of agents shaped in a concrete application domain. We present a scenario of interacting agents providing a computational model implementing different strategies to assess trust. Assuming a medical domain, categories, including both competencies and dispositions of possible trustees, are exploited to infer trust towards possibly unknown

counterparts. The proposed approach for the cognitive assessment of trust relies on agents' abilities to analyze heterogeneous information sources along different dimensions. Trust is inferred based on specific observable properties (Manifesta), namely explicitly readable signals indicating internal features (Krypta) regulating agents' behaviour and effectiveness on specific tasks. Simulative experiments evaluate the performance of trusting agents adopting different strategies to delegate tasks to possibly unknown trustees, while experimental results show the relevance of this kind of cognitive ability in the case of open Multi Agent Systems.

References

S. Ahern, M. Davis, S. King, M. Naaman, and R. Nair. Reliable, user-contributed gsm cell-tower positioning using context-aware photos. In *Adjunct Proceedings of the Eighth International Conference on Ubiquitous Computing (UbiComp 2006)*, 2006.

M. Alvarez, D. Luengo, and N. Lawrence. Latent force models. In *International Conference on Artificial Intelligence and Statistics*, pages 9–16, 2009.

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

D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):58–71, 2007.

Y. Bachrach, T. Graepel, G. Kasneci, M. Kosinski, and J. Van Gael. Crowd iq: aggregating opinions to boost performance. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 535–542, 2012a.

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*, 2012b.

C. 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.

O. Bochardt and J. Uhlmann. On the equivalence of the general covariance union (gcu) and minimum enclosing ellipsoid (mee) problems. *CoRR*, abs/1012.4795, 2010.

V. Bodden. *Boston Marathon Bombings*. ABDO Publishing Company, 2014.

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

A. Brix and P. Diggle. Spatiotemporal prediction for log-Gaussian cox processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(4):823–841, 2001.

R. Brooks and S. Iyengar. *Multi-sensor fusion: fundamentals and applications with software*. Prentice-Hall, Inc., 1998.

A. Brown. GPS precision approach and landing system for aircraft, May 10 1994. US Patent 5,311,194.

J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. In *In: Workshop on World-Sensor-Web (WSWÕ06): Mobile Device Centric Sensor Networks and Applications*, pages 117–134, 2006.

C. Castelfranchi and R. Falcone. Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In *Multi Agent Systems, 1998. Proceedings. International Conference on*, pages 72–79. IEEE, 1998.

C. Castelfranchi and R. Falcone. *Trust theory: A socio-cognitive and computational model*, volume 18. John Wiley & Sons, 2010.

D. Clow and E. Makriyannis. ispot analysed: Participatory learning and reputation. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, LAK '11, pages 34–43, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0944-8. URL: http://doi.acm.org/10.1145/2090116.2090121.

C. Corbane, G. Lemoine, and M. Kauffmann. Relationship between the spatial distribution of sms messages reporting needs and building damage in 2010 haiti disaster. *Natural Hazards & Earth System Sciences*, 12(2), 2012.

D. Cox and V. Isham. *Point processes*, volume 12. CRC Press, 1980.

N. Cressie and C. K. Wikle. *Statistics for spatio-temporal data*. John Wiley & Sons, 2011.

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

M. DeGroot and M.J. Schervish. *Probability and Statistics*. Pearson Education, 4th edition, 2012. ISBN 9780321500465. URL: http://books.google.co.uk/books?id=4TlEPgAACAAJ.

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.

P. Diggle. *Statistical Analysis of Spatial and Spatio-temporal Point Patterns*. CRC Press, 2013.

P. Diggle, B. Rowlingson, and T. Su. Point process methodology for on-line spatio-temporal disease surveillance. *Environmetrics*, 16(5):423–434, 2005.

Region EPA. United states environmental protection agency, 2001.

R. Falcone and C. Castelfranchi. Generalizing trust: Inferencing trustworthiness from categories. In *Trust in Agent Societies*, pages 65–80. Springer, 2008.

P. Farmer. *Haiti after the earthquake*. PublicAffairs, 2012.

M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. *Selected Areas in Communications, IEEE Journal on*, 24(5): 1010–1019, 2006.

P. Flach, S. Spiegler, B. Golénia, S. Price, J. Guiver, R. Herbrich, T. Graepel, and M. Zaki. Novel tools to streamline the conference review process: experiences from sigkdd'09. *ACM SIGKDD Explorations Newsletter*, 11(2):63–67, 2010.

K. Friston, J. Mattout, N. Trujillo-Barreto, J. Ashburner, and W. Penny. Variational free energy and the laplace approximation. *NeuroImage*, 34(1):220–234, 2007.

D. Gambetta. *Trust: Making and Breaking Cooperative Relations,*. Basic Blackwell.

H. Gao, G. Barbier, and R. Goolsby. Harnessing the crowdsourcing power of social media for disaster relief. *IEEE Intelligent Systems*, 26(3):10–14, 2011.

A. Gelfand and A. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

E. Gertz and P. Di Justo. *Environmental Monitoring with Arduino: Building Simple Devices to Collect Data About the World Around Us*. O'Reilly Media, Inc., 2012.

Z. Ghahramani. Unsupervised learning. In *Advanced Lectures on Machine Learning*, pages 72–112. Springer, 2004.

W. Gilks. *Markov Chain Monte Carlo*. Wiley Online Library, 2005.

P. Goldberg, C. Williams, and C. Bishop. Regression with input-dependent noise: A Gaussian process treatment. *Advances in Neural Information Processing Systems*, 10: 493–499, 1997.

M. Goodchild and A. Glennon. Crowdsourcing geographic information for disaster response: a research frontier. *International Journal of Digital Earth*, 3(3):231–241, 2010a.

M. Goodchild and J. Glennon. Crowdsourcing geographic information for disaster response: a research frontier. *International Journal of Digital Earth*, 3(3):231–241, 2010b.

P. Groot, A. Birlutiu, and T. Heskes. Learning from multiple annotators with Gaussian processes. In *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 159–164. Springer, 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. Hageman and D. Young. *Applied iterative methods*. Dover Publications, 2004.

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

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

J. Heinzelman and C. Waters. *Crowdsourcing crisis information in disaster-affected Haiti*. US Institute of Peace, 2010.

E. Horvitz, J. Apacible, R. Sarin, and L. Liao. Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. *arXiv preprint arXiv:1207.1352*, 2012.

J. Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

M. Hurley. An information theoretic justification for covariance intersection and its generalization. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, volume 1, pages 505–511. IEEE, 2002.

T. D. Huynh, N. R Jennings, and N. R Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.

T.D. Huynh, N. R. Jennings, and N. Shadbolt. Fire: An integrated trust and reputation model for open multi-agent systems. In *16th European Conference on Artificial Intelligence, Valencia, Spain*, pages 18–22, 2004.

P. 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.

P. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.

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

L. Kagal, T. Finin, and A. Joshi. A policy based approach to security for the semantic web. In *The Semantic Web-ISWC 2003*, pages 402–418. Springer, 2003.

E. Kamar, S. Hacker, and E. Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 467–474, 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.

D. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *NIPS*, pages 1953–1961, 2011.

G. Kazai. In search of quality in crowdsourcing for search engine evaluation. In *Advances in information retrieval*, pages 165–176. Springer, 2011.

L. 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.

H. Kim and Z. Ghahramani. Bayesian classifier combination. In *International Conference on Artificial Intelligence and Statistics*, pages 619–627, 2012.

A. Kittur, E. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM, 2008.

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 Textual Entailment*, pages 95–116, 2006.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

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

E. Law and L. Von Ahn. *Human Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.

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

J. Jingshi Li, A. Jutzeler, and B. Faltings. Estimating urban ultrafine particle distributions with Gaussian process models. In *In S. Winter and C. Rizos (Eds.): Research@Locate'14*, pages 145–153, 2014.

M. Liggins II, D. Hall, and J. Llinas. *Handbook of multisensor data fusion: theory and practice*. CRC press, 2008.

S. Marsh. Formalising trust as a computational concept. 1994.

T. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

Barbara Misztal. *Trust in modern societies: The search for the bases of social order*. John Wiley & Sons, 2013.

J. Møller, A. R. Syversveen, and R. P. Waagepetersen. Log Gaussian cox processes. *Scandinavian journal of statistics*, 25(3):451–482, 1998.

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.

N. Morrow, N. Mock, A. Papendieck, and N. Kocmich. Independent evaluation of the ushahidi haiti project. *Development Information Systems International*, 8, 2011.

J. Myers, A. Well, and R. Lorch. *Research design and statistical analysis*. Routledge, 2010.

R. Narasimhan. *College Algebra and Trigonometry: Building Concepts and Connections*. Cengage Learning, 2008.

V. Naroditskiy, I. Rahwan, M. Cebrian, and N. R. Jennings. Verification in referral-based crowdsourcing. *PloS One*, 7(10):e45924, 2012.

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.

M. Osborne. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, Oxford University New College, 2010.

A. Overeem, J. Robinson, H. Leijnse, G. Steeneveld, B. Horn, and R. Uijlenhoet. Crowdsourcing urban air temperatures from smartphone battery temperatures. *Geophysical Research Letters*, 40(15):4081–4085, 2013.

S. Parsons, E. Sklar, M. Singh, K. Levitt, and J. Rowe. An argumentation-based approach to handling trust in distributed decision making. In *Trust and Autonomous Systems: Papers from the 2013 AAAI Spring Symposium*, pages 1951–1957, 2013.

J. Pino and J. Pezoa. Cobcel: Distributed and collaborative sensing of cellular phone coverage using google android. In *ICSNC 2012, The Seventh International Conference on Systems and Networks Communications*, pages 228–230, 2012.

I. Pinyol and J. Sabater-Mir. Computational trust and reputation models for open multi-agent systems: a review. *Artificial Intelligence Review*, 40(1):1–25, 2013.

J. Quinn, K. Leyton-Brown, and E. Mwebaze. Modeling and monitoring crop disease in developing countries. In *AAAI*, 2011.

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

S. 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/.

S. Ramchurn, C. Sierra, L. Godó, and N. R. Jennings. A computational trust model for multi-agent interactions based on confidence and reputation. In *6th International Workshop of Deception, Fraud and Trust in Agent Societies*, pages 69–75, 2003.

C. Rasmussen and C. Williams. Gaussian processes for machine learning. *Gaussian Processes for Machine Learning*, 2006.

V. Raykar, S. Yu, L. Zhao, G. Valadez, C. Florin, Luca B., 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.

P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

A. Rutherford, M. Cebrian, I. Rahwan, S. Dsouza, J. McInerney, V. Naroditskiy, M. Venanzi, N. R. Jennings, E. Wahlstedt, S. Miller, et al. Targeted social mobilization in a global manhunt. *PloS One*, 8(9):e74628, 2013.

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

A. Sadilek, H. A Kautz, and V. Silenzio. Modeling spread of disease from social interactions. In *ICWSM*, 2012.

R. Schlaifer and H. Raiffa. Applied statistical decision theory. 1961.

A. Sheshadri and M. Lease. Square: A benchmark for research on computing crowd consensus. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

B. 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. Simpson, S. Roberts, I Psorakis, and A. Smith. Dynamic Bayesian combination of multiple imperfect classifiers. *arXiv preprint arXiv:1206.1831*, 2012.

R. Sinnott. Virtues of the haversine. *Sky and telescope*, 68:158, 1984.

D. Slater, K. Nishimura, and L. Kindstrand. Social media, information, and political activism in japan's 3.11 crisis.". *The Asia-Pacific Journal*, 10(24.1), 2012.

R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics, 2008.

J. Snyder. *Map projections–A working manual*. Number 1395. USGPO, 1987.

D. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online Bayesian recommendations. In *Proceedings of the 18th international conference on World wide web*, pages 111–120. ACM, 2009.

R. Stranders. *Decentralised coordination of information gathering agents*. PhD thesis, University of Southampton, 2010.

W. Teacy. *Agent-based trust and reputation in the context of inaccurate information sources*. PhD thesis, University of Southampton, 2006.

W. L. Teacy, J. Patel, N. R Jennings, and M. Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

M. Teraguchi, S. Saito, T. Lau, M. Ohno, J. A. Cerruti, and H. Takagi. Crowdsourcing in crisis informatics for disaster relief. *Guest Editors*, page 60, 2011.

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

L. Tran-Thanh, S. Stein, A. Rogers, and N. R Jennings. Efficient crowdsourcing of unknown experts using multi-armed bandits. In *European Conf. on Artificial Intelligence*, pages 768–773, 2012.

L. Tran-Thanh, M. Venanzi, A. Rogers, and N. R Jennings. Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In *Proceedings of the 2013 International Conference on Autonomous agents and multi-agent systems*, pages 901–908, 2013.

M. Venanzi, A. Rogers, and N. R Jennings. Crowdsourcing spatial phenomena using trust-based heteroskedastic Gaussian processes. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013a.

M. Venanzi, A. Rogers, and N. R Jennings. Trust-based fusion of untrustworthy information in crowdsourcing applications. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 829–836, 2013b.

T. Wallsten, D. Budescu, I. Erev, and A. Diederich. Evaluating and combining subjective probability estimates. *Journal of Behavioral Decision Making*, 10(3):243–268, 1997.

P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *NIPS*, volume 10, pages 2424–2432, 2010.

J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, volume 22, pages 2035–2043, 2009.

J. Winn, C. Bishop, and T. Jaakkola. Variational message passing. *Journal of Machine Learning Research*, 6(4), 2005.

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

M. Wooldridge and N. R. Jennings. Software engineering with agents: Pitfalls and pratfalls. *Internet Computing, IEEE*, 3(3):20–27, 1999.

A. Yasuhiko. Safecast or the production of collective intelligence on radiation risks after 3.11 yasuhiko abe. *The Asia-Pacific Journal, Vol*, 11:2014, 2011.

B. Yu and M. Singh. An evidential model of distributed reputation management. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 294–301. ACM, 2002.