Statistical Data Analysis for Internet-of-Things

Scalability, Reliability, and Robustness

XIUMING LIU
Internet-of-Things is a set of sensing, communication, and computation technologies to connect physical objects, such as wearable devices, vehicles, and buildings. From those connected “Things”, a large amount of data is generated. Data analysis plays a central role in the automated and intelligent decision-making process to manage and optimize IoT systems. In this thesis, we focus on tackling the challenges of analyzing large, incomplete, and corrupt IoT data. This thesis consists of three topics. In the first topic, we study scalable GP regression for big IoT data. We propose a novel scalable GP model for urban air quality modeling and prediction. Comparing to the existing scalable GP models, the proposed scalable GP model enables tractable analysis of approximation errors. The second topic is to handle the missing data problem. In the case of missing labels in training data, we investigate different missing data mechanisms. We propose a reliable semi-supervised learning approach, which provides accurate predictive error probability. In the case of missing features in testing data, we design a robust predictor. The predictor significantly reduces the prediction error caused by rare values of missing features, while incurring only a small loss on the overall performance. The third topic is information fusion for IoT systems under false data injection attacks. We propose a robust and distributed information fusion method. This proposed information fusion method only requires exchanging the latest local posterior distributions, instead of synchronizing the full historical measurements. Furthermore, we design a false data detector based on the clustering of local posterior distributions. The distributed information fusion method and false data detector enable secure state estimation for mobile IoT networks with probabilistic communication links. Altogether, this thesis is a step to scalable, reliable, and robust IoT data analysis.

Xiuming Liu, Department of Information Technology, Computer Systems, Box 337, Uppsala University, SE-75105 Uppsala, Sweden.

© Xiuming Liu 2020

ISSN 1651-6214
urn:nbn:se:uu:diva-421618 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-421618)
To my family
Acknowledgements

First of all, I would like to thank my main supervisor, Dr. Edith C.H. Ngai. Thank you, Edith, for giving me the opportunity to pursue my doctoral education and guiding me through the journey. And thank you for your constant support for my education, research, and life in Uppsala. You have been not only a great supervisor but a good friend to me.

I am very grateful to Dr. Dave Zachariah for being my co-supervisor. Thank you, Dave, for your inspiring lectures on statistical estimation theory. And thank you for all the insightful discussions. Many research ideas and results have been created on that whiteboard in your office!

I am also very grateful to Prof. Bengt Jonsson for being my co-supervisor. Thank you for guiding my doctoral education. And thank you for showing me your rigorous attitude towards scientific research.

I want to thank Prof. Peter Stoica for co-authoring two of the research papers included in this thesis. Your contributions have improved the quality of those papers and my understanding of good scientific research. I want to thank Prof. Christian Rohner for providing support to my teaching duties. It is a pleasure to work with you. I want to thank Prof. Thomas Schön for a series of inspiring graduate courses on statistical machine learning.

It has been a great experience to study and work in the embedded system research group. I want to thank Prof. Wang Yi, Dr. Philipp Rümmer, Dr. Pontus Ekberg, Dr. Morteza Mohaqeqi, Dr. Amir Amini-far, Dr. Vera van Zoest, Dr. Aleksandar Zeljic, Dr. Peter Backeman, Syed Md Jakaria Abdullah, Gaoyang Dai, Chencheng Liang, Shenghui Li, and Nikolaus Huber. Thank you for creating a friendly and open work environment. I also want to thank my friends Dr. Yunyun Zhu, Dr. Lei You, Alex Yuan Gao, and Wenqing Yan, for their support.

I am very grateful to Vinnova, the Swedish Research Council, and Uppsala University for supporting my doctoral education and research.

Finally, I want to thank my family. Many thanks to my parents for their constant support. A thousand thanks and love to my dear wife Mengjia and my daughter Emily. The home is where the heart is. The home is where you are.

Xiuming Liu
Uppsala, October 2020
List of papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.


Reprints were made with permission from the publishers.
## Contents

1 Introduction .......................................................... 11
  1.1 Statistical Data Analysis ........................................ 12
  1.2 Research Questions ............................................. 15
  1.3 Thesis Structure and Contributions .......................... 17
  1.4 Notations .......................................................... 20

2 Learning with Big Data ............................................. 21
  2.1 Gaussian Processes .............................................. 21
    2.1.1 GP Learning ............................................... 22
    2.1.2 GP Prediction ............................................ 23
  2.2 Update Belief Distributions ................................... 24
    2.2.1 General Belief Update ................................... 25
    2.2.2 Scalable Belief Update .................................. 26
  2.3 Scalable GP Regression ......................................... 27
    2.3.1 Scalable GP Learning .................................... 27
    2.3.2 Scalable GP Prediction .................................. 28
  2.4 Literature Review .............................................. 30
  2.5 Summary .......................................................... 31

3 Handling Missing Data ............................................. 34
  3.1 Missing Labels .................................................. 34
    3.1.1 Missing Labels Patterns ................................ 35
    3.1.2 Semi-Supervised Learning ............................... 37
    3.1.3 Reliability of Error Probabilities ..................... 39
  3.2 Missing Features ............................................... 42
    3.2.1 Optimistic Predictor ................................... 42
    3.2.2 Constrained Predictor ................................. 44
    3.2.3 Robust Predictor ....................................... 46
  3.3 Literature Review .............................................. 47
  3.4 Summary .......................................................... 48

4 Detecting False Data ............................................... 50
  4.1 Distributed State Estimation ................................. 50
    4.1.1 Distributed State Estimation .......................... 50
    4.1.2 Consensus Based Distributed Estimation .......... 53
  4.2 False Data Detector ........................................... 56
    4.2.1 Centralized Detector ................................. 57
    4.2.2 Distributed Detector ................................. 58
  4.3 Literature Review .............................................. 60
1. Introduction

The ubiquitous communication and computation technologies have created a digital twin of the physical world. Things – such as buildings, vehicles, and wearable devices – have been represented and connected in the digital world, hence the name Internet-of-Things (IoT). IoT generate various types and large amounts of data, enabling the design and deployment of intelligent systems on a large scale.

Data analysis plays a central role in IoT applications. In a typical IoT application, we apply data analysis to monitor a system’s status, model its dynamics, and adjust its configurations. For example, using low-cost air quality sensors to monitor urban environments, using smart meters to report real-time energy consumption, and using wearable devices to help individuals to improve their physical conditions.

When designing data models and analysis methods for IoT applications, the following properties of IoT shall be considered:

- **Resource constraint**: IoT devices typically have minimal computation capability, communication capacity, and energy storage. While cloud computing is a viable solution, IoT applications often require the data analysis to be executed locally or at network edges, for latency and privacy concerns;

- **Distributed architecture**: IoT applications are often operated over distributed networks. In a distributed network, there is no centralized data center. Therefore data models and analysis methods must be designed to adapt the distributed architecture;

- **Missing data**: When sensors or communication fail in IoT, data will be lost. Missing data require careful treatment in the analysis. Missing data can be interpolated under certain assumptions. Besides interpolation, it is essential to analyze the missing data patterns and be aware of rare events in missing data.

- **False data**: When deploying a large scale IoT, not every device is equipped with high-quality sensors or secure software. Therefore IoT data models and analysis methods must be robust to false data.

In light of those challenges, this thesis presents several new data models and analysis methods for IoT systems in order to achieve scalability, reliability, and robustness. This chapter presents a background of the statistical data analysis and an overview of this thesis.
1.1 Statistical Data Analysis

Statistical data analysis is a process of pattern discovery, data modeling, and prediction based on statistics principles. Data analysis has intersections with other research fields, such as signal processing and machine learning. On the other hand, data analysis research covers the entire data life circle, including collection, cleaning, modeling, and visualization. This thesis presents results related to the following three steps of statistical data analysis: exploratory data analysis, data modeling, and statistical inference.

Exploratory Data Analysis

Exploratory data analysis (EDA) aims to visualize the statistical properties of data. Anscombe’s quartet [Anscombe 1973] well demonstrates the importance of EDA. EDA helps us to discover data properties and patterns visually. For example, in Chapter 2, the visualization of air quality data gives us hints about transformation techniques and model selection; in Chapter 3, data visualization in a low-dimensional feature space helps us identify missing data patterns.

\[\text{Figure 1.1. Anscombe’s quartet. The four datasets show large differences when they are visualized. However, the datasets share the same basic statistics (mean, standard deviation, and correlation coefficient) and the same linear regression model.}\]
Data Modeling
Suppose we observed a sequence of measurements from an IoT device:
\[
x_{t_1:N} = [x_{t_1}, x_{t_2}, \ldots, x_{t_N}]^\top.
\]

The goal is to find a statistical model for the underlying process. In this thesis, we use two types of models, parametric and non-parametric models.

Parametric Models
A parametric model has a predetermined input-output structure of the underlying process. Define the inputs as the past \(k\) measurements of the process at time \(t_N\): \(x_{t_{N-k:N-1}} = [x_{t_{N-k}}, \ldots, x_{t_{N-1}}]^\top\), and the output as the measurement \(x_{t_N}\), a general form of parametric models is
\[
x_{t_N} = f_\Theta(x_{t_{N-k:N-1}}) + \epsilon, \quad (1.1)
\]
where \(f_\Theta(\cdot)\) is a linear or nonlinear function with a finite set of parameters \(\Theta\), and \(\epsilon\) is a random variable to represent the measurement noise. Examples of parametric models are linear regression models and neural networks. In Chapter 3 of this thesis, we study the robust prediction problem with parametric models. In Chapter 4, we design false data detector based on parametric models.

Non-parametric Models
A non-parametric model does not have a predetermined input-output structure. The term ”non-parametric” does not imply the non-existence of parameters, but rather that the input-output structure is flexible and not defined by a finite set of parameters. An example of non-parametric models is a Gaussian process (GP) [Rasmussen and Williams 2006]. A GP does not have a fixed input-output structure and is not defined by a finite set of parameters. Instead, a GP is uniquely defined by its mean and covariance functions, \(m(t)\) and \(\text{cov}(t,t')\):
\[
x_t \sim \text{GP}(m(t), \text{cov}(t,t')). \quad (1.2)
\]
The observed measurements \(x_{t_{1:N}} = [x_{t_1}, x_{t_2}, \ldots, x_{t_N}]^\top\) is a realization of the GP. Given the previous \(N\) measurements, the future value follows a conditional Gaussian distribution
\[
x_* \sim p(x_* \mid x_{t_{1:N}}), \quad (1.3)
\]
where its conditional mean and variance are determined by the mean and covariance functions of the GP.

In this thesis, we apply GP to model and predict air pollution in an urban environment. We propose scalable GP learning and prediction methods based on the general belief updating framework and composite likelihood theory.
Estimation, Prediction, and Detection

After EDA and data modeling, the next task is to infer unknown variables or test hypotheses for decision support. Examples are forecasting future air quality for municipal and healthcare services, fault diagnosis for electrical power networks. Those applications are built based on solving three fundamental problems in statistical data analysis: estimation, prediction, and detection.

**Estimation**

Estimation is the process of using data to infer unknown quantities, such as model parameters or system states. These unknown quantities can be viewed as either deterministic or random, which leads to the frequentist or Bayesian approaches, respectively.

The frequentists approach considers the unknown quantity to be fixed in the parameter or state space. However, the mapping from the parameter or state-space to the observation space is typically probabilistic (for example, due to random noises in sensor measurements) [Van Trees et al. 2013]. Therefore, given finite observations, the estimation rule (e.g., maximum likelihood estimation MLE) returns probabilistic results even for fixed unknown quantities. The expected error between an estimate and the unknown target is called the bias, which is a measure of an estimator’s quality. Another important measure of an estimator’s quality is the variance of the estimates.

The Bayesian approach considers the unknown quantity to be random and assigns a prior probability distribution to the unknown quantity. A posterior distribution is then derived for the unknown quantity based on data sampled from the observation space. The maximum a posterior (MAP) estimation gives a point estimate for the unknown quantity. Moreover, the posterior variance can be used to measure the estimate’s uncertainty.

Chapter 2 of this thesis uses a weighted combination of MLEs to estimate the model hyper-parameters based on a large amount of historical air quality data. In Chapter 4, we design a recursive and distributed Bayesian filter for state estimation in IoT networks.

**Prediction**

Prediction is a process of using data to infer outcomes of new random variables. An example is forecasting future air quality based on historical data. Classification is also a type of prediction, where the outcome of the new random variable is categorical.

An important remark about prediction is that we assume the target variable is random in itself. Usually, we cannot repeat the same experiment multiple times to sample the variable. In the air quality forecast
example, the concentration of air pollutants in a city is affected by vehicle emissions, industrial activities, and weather conditions. Therefore there is a high level of randomness in the future air quality. Furthermore, once the air quality is observed, we cannot repeat the experiment with the same conditions. Therefore, a predictive probability represents our belief about the outcome, rather than the frequency in which the outcome occurs.

In Chapter 2 of this thesis, we design a scalable GP model for urban air quality forecasting. In Chapter 3, we design a reliable classifier to predict labels of images.

**Detection**
Detection is different from the estimation and prediction. The target is no longer just an unknown variable, but rather an entire data generating process, an event, or a pattern. Nonetheless, we rely on the data sampled from the underlying unknown process to select the most likely hypothesis.

For example, we want to know if a device is under false data injection attacks. Assuming the target system’s data distribution is known, we can evaluate the divergence between an observed dataset’s empirical distribution with the target distribution. If the divergence is larger than a threshold, we accept the hypothesis that the data is not generated from the target distribution – hence being corrupted by false data.

In Chapter 3, we define a detection rule to select informative unlabeled data. In Chapter 4, we design a false data detector for secure information fusion in distributed IoT systems.

### 1.2 Research Questions
This thesis studies research questions with the following three keywords: *scalability*, *reliability*, and *robustness*.

Scalability
Considering limited resources and distributed architectures of IoT systems, we aim to design scalable statistical data analysis methods and models for processing a large amount of data. Specifically, we want the computation complexity and the memory requirement of analyzing data to increase only moderately, when the data size grows.

Taking the GP regression as an example, both parameter learning and variable prediction require inversions of data covariance matrices. Suppose the data size is $N$, the Gaussian elimination based matrix inversion algorithm has the complexity of $O(N^3)$ and the memory requirement of
\( \mathcal{O}(N^2) \). Therefore it is infeasible to deploy GP regression applications on resource-constrained IoT devices when the data size \( N \) is large.

In this thesis, we design scalable GP regression methods for IoT data analysis. We adapt the general belief update framework for processing large data by using the approximated Gaussian likelihood function. The proposed scalable GP learning and prediction methods can be implemented recursively for a single device with streaming data or a distributed manner for a cluster of devices.

Specifically, in **Paper I** and **II**, we answer the following questions:

1.1: How to design a unified, scalable framework for applying GP regression to IoT data analysis? The framework needs to facilitate processing a large dataset on a single resource-constrained device and processing many local datasets cooperatively over an IoT network.

1.2: What are the connections between the proposed and the existing scalable GP methods? Based on a unified framework, we want to compare different methods from a fundamental perspective.

1.3: How to analyze the difference between results given by the exact GP model and the approximated GP model? We evaluate the difference using two metrics: the data averaged mean squared error and the data averaged KL divergence.

**Reliability**

When IoT data are used for decision support, both the predictive value and the predictive uncertainty are required. When a predictive result is associated with a relatively large uncertainty level, it should be used conservatively. Therefore the statistical data analysis model and method must provide a reliable predictive uncertainty measure.

In short, the reliability of a method or a model can be measured by using the difference between the predictive uncertainty and the empirical error distribution. For example, in regression problems, the predictive variance is compared with the empirical mean square errors; in classification problems, the predictive error probability is compared with the empirical error probability. A reliable data model or prediction method has a small difference between its predictive uncertainty and the empirical error distribution.

In **Paper III**, we propose a reliable semi-supervised learning approach. And the paper answers the following questions:

2.1: When are the unlabeled data useful for improving the quality of generative models? We discuss three missing data assumptions (missing completely at random, missing at random, and missing not at random).
2.2: How to evaluate the reliability of a classifier? Moreover, how to build reliable classifiers using unlabeled data? We design a reliable semi-supervised learning method that provides accurate predictive error probabilities.

Robustness
The robustness is a broad concept, and it needs to be discussed in a specific context. In this thesis, we aim to design IoT data analysis models and methods which are:

- Robust to rare events in missing data. Data models are often learned by minimizing the average cost function. This objective implies that the model will perform well for new data which have "typical" features. When there are rare values in unobserved features, the system shall switch to a robust model.
- Robust to the communication failure. In distributed IoT systems, each device communicates with its neighbors to exchange information. In case of communication failure, the system needs to operate based on limited local information.
- Robust to false data. False data endanger the safety of an IoT system by introducing unexpected errors to state estimation. Therefore we need to design a false data detector and a robust state estimator for distributed IoT systems.

Generally speaking, the robustness of IoT data analysis can be achieved by constraining the worst-case scenario’s impact or by adding redundant information to the data. However, the cost of those measures needs to be considered in the overall design.

Papers IV, V and VI answer the following questions:
3.1: When there are unobserved features in new data, how to design a predictor which is robust to possible rare events? In Paper IV, we design a robust predictor for data with missing features.
3.3: How to design a distributed state estimation method which is robust to communication failure for mobile IoT systems? In Paper VI, we design a distributed estimator which only exchanges the latest local posterior instead of raw measurement data.
3.3: How to design a distributed false data detector based on the local information? In paper VI, we design a false data detector based on the clustering of local posteriors.

1.3 Thesis Structure and Contributions
This thesis is structured as follows. In Chapter 2, we present a scalable GP regression method. We apply the method to predict urban air qual-
Figure 1.2. Connections between chapters and papers in this thesis.

ity based on a large amount of historical data. In Chapter 3, we discuss the missing data problem. First, we present a reliable semi-supervised learning method for training data with missing labels; second, we design a robust predictor for handling rare events in test data with missing features. In Chapter 4, we present results about distributed state estimation under false data injection attacks. We design a secure information fusion method for distributed IoT systems. Figure 1.2 illustrates the thesis structure.

This thesis presents IoT data models and analysis methods to achieve scalability, reliability, and robustness. The contributions are summarized here.

Chapter 2:
In Paper I, we propose a scalable GP learning and prediction method, based on a general belief updating framework and the composite likelihood theory. We show the fundamental connection between the proposed scalable GP model and some existing scalable GP models, such as the sparse GP [Quiñonero-Candela and Rasmussen 2005] and the Bayesian committee machine [Tresp 2000]. We design recursive GP parameter learning and latent variable prediction algorithms. Furthermore, we analyze the difference between the approximate GP posterior and the exact GP posterior by comparing their data averaged KL divergences.

In Paper II, we apply the scalable GP model to predict urban air quality. The model is learned from a real-world dataset that consists of two air pollutants and several meteorological measurements. We show
that the model can produce accurate prediction results based on a large amount of historical data.

Chapter 3:
In Paper III, we propose a reliable semi-supervised learning method for data with labels missing at random. We design a selective self-labeling approach, which only uses informative unlabeled data to enhance the supervised classifier. A classifier’s reliability is measured by the Brier score, which is the averaged difference between the predictive error probability and the actual error probability.

In Paper IV, we design a robust predictor for data with missing features. The robust predictor is an adaptive combination of the optimistic and conservative predictors. The combination weights are designed based on the estimated probability that the missing feature value is located in the tail region. The experiment results show that the robust predictor achieves low MSE when the missing features take rare values, while only incurring a small loss to the overall performance.

Chapter 4:
In Paper V, we study the distributed GP hyper-parameter estimation problem when the network is under false data injection attacks. We investigate the impact of false data on the estimation accuracy and convergence speed when the attack is implemented at different network nodes.

In Paper VI, we study the distributed state estimation problem for IoT systems under false data injection attacks. We propose an information fusion method, based on the consensus of local posteriors. The information fusion method is integrated with a distributed false data detector, which clusters local posteriors using the symmetrized KL divergence. The proposed method is simulated with a distributed and mobile sensor network for spatial-temporal signal monitoring.
1.4 Notations

$x, y, z, \alpha, \beta, \theta$  Scalars
$x, y, z, \alpha, \beta, \theta$  Vectors
$X, Y, Z, \Sigma$  Matrices
$\mathbb{R}^d$  A real coordinate space of dimension $d$
$D = \{(x_i, y_i)\}$  A set of data which consists of input-output pairs $(x_i, y_i)$

$X^\top$  Transposes of a matrix
$X^{-1}$  Inverse of a matrix
$X^\dagger$  Moore-Penrose inverse of a matrix
$|X|$  Determinant of a matrix
$\text{Tr}(X)$  Trace of a matrix
$\text{rank}(X)$  Rank of a matrix
$||x||_p$  $p$-norm of vector $x$. $p = 2$ if unspecified

$\bar{x}$  Mean of random variable $x$
$\hat{\theta}$  Estimate of parameter $\theta$
$p(x)$  Probability density of random variable $x$
$p(x | y)$  Conditional probability of $x$ given $y$
$\mathbb{E}[f(x)]$  Expectation of function $f(x)$, where $x$ is a random variable
$\mathbb{E}_n[f(x)]$  Sample expectation of function $f(x)$
$D_{\text{KL}}(p \ || \ q)$  Kullback-Leibler (KL) divergence between probability distributions $p$ and $q$

$max f(x)$  The maxima of function $f(x)$
$min f(x)$  The minima of function $f(x)$
$\arg \max f(x)$  The value of $x$ at which function $f(x)$ is maximized
$\arg \min f(x)$  The value of $x$ at which function $f(x)$ is minimized
2. Learning with Big Data

In this chapter, we discuss the first challenge of the IoT data analysis – large amounts of data and limited computation resources. We will focus on the GP regression and present new results of making the GP learning and prediction scalable for IoT data analysis.

Roadmap
Chapter 2 starts with introducing some basic concepts of Gaussian processes, including model hyper-parameter estimation, sample covariance estimation, and latent variable prediction. Next, we introduce a generalized belief updating framework [Bissiri et al. 2016], and discuss how scalability can be achieved by using an alternative loss function within the framework. In the third section of this chapter, we propose new scalable GP learning and prediction methods, based on the belief updating framework and the composite likelihood method [Varin et al. 2011]. Chapter 2 is an ensemble of results from Paper I and II.

2.1 Gaussian Processes
Consider an IoT device which has collected a large sequence of data of a physical process over time $t = [t_1 \ldots t_N]^T$:
\[
x_t = [x_{t_1} x_{t_2} \ldots x_{t_N}]^T.
\] (2.1)
The goal is to build a statistical model based on the historical data and use the model to predict future values of the process.

Gaussian processes (GPs) are a type of stochastic models which are widely used in modelling dynamics of time-series and spatial data. A GP model is specified by its mean and covariance functions,
\[
x_t \sim \text{GP}(m(t), \text{cov}(t,t')).
\] (2.2)

An exemplary GP is shown in Figure 2.1. Regression analysis with GPs usually consists of two steps: first, using historical data to determine the mean and covariance functions (GP learning); second, using the learned GP model together with historical data to predict the future values of the process (GP prediction). The GP learning and prediction methods are summarized in the following subsections.
Figure 2.1. An example of GP prior and posterior. In (a), a GP prior $x_t \sim \text{GP}(m(t),\text{cov}(t,t'))$ is defined, where $m(t) = 0$ and $\text{cov}(t,t') = \exp\{-\frac{(t-t')^2}{2}\}$. The black line represents the mean and the gray area represents the interval between mean plus and minus two standard deviations. The red lines are ten realizations of the GP prior. In (b), five observations from the process are made. The posterior mean and plus/minus two deviation interval are visualized by the black line and the gray area, respectively. The red lines are ten realizations of the GP posterior.

2.1.1 GP Learning

There are two approaches to learn the mean and covariance functions. The first approach is to select template functions for the mean and covariance functions, and then estimate the values of hyper-parameters in the selected template functions. Examples of template functions are linear functions for the mean function,

$$m(t) = at + b,$$

(2.3)

and squared-exponential functions for the covariance function,

$$\text{cov}(t,t') = \alpha^2 \exp[-\frac{(t-t')^2}{2\beta^2}].$$

(2.4)

The parameters $\theta = [a \ b \ \alpha \ \beta]^\top$ in the mean and covariance functions are called the hyper-parameters of the GP model. The estimate of the hyper-parameters can be obtained by minimizing the negative-log likelihood function:

$$\hat{\theta} = \arg\min_{\theta} -\ln p(x_t | \theta)$$

$$= \arg\min_{\theta} (x_t - \mu_t)^\top \Sigma_{t,t}^{-1} (x_t - \mu_t) + \ln |\Sigma_{t,t}| + N \ln 2\pi,$$

(2.5)

where $\mu_t$ is the mean vector and $\Sigma_{t,t}$ is the covariance matrix. The values of the elements in $\mu_t$ and $\Sigma_{t,t}$ are determined by the mean and the covariance functions which are parameterized by $\theta$. An example of learning GP hyper-parameters is visualized in 2.2.
Figure 2.2. An example of GP hyper-parameter estimation. In (a), the GP is fit to five observations, using the squared-exponential covariance function. In (b), the log likelihood is evaluated and plotted in the space of hyper-parameters. The maximum likelihood estimate (black dot) of hyper-parameters is $\hat{\alpha} = 1.43$ and $\hat{\beta} = 0.906$. The true hyper-parameter values are $\alpha = 1$ and $\beta = 1$.

The second approach is to use the sample mean and covariance functions. In this case, we assume that the IoT device has a fixed data collection rate. For example, an air quality monitoring station reports hourly averaged data. We can estimate the daily mean vector by folding the historical data into a $24 \times D$ matrix with elements $\{x^d_t \mid t \in 1, \ldots, 24 \text{ and } d \in 1, \ldots, D\}$, where $D$ is the number of days in the data history. The daily sample mean vector is then given by averaging each column of hour over the rows of days:

$$
\mu(t) = \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_{24} \end{bmatrix}^T
= \frac{1}{D} \left[ \sum_{d=1}^{D} x^d_1 \sum_{d=1}^{D} x^d_2 \cdots \sum_{d=1}^{D} x^d_{24} \right]^T.
$$

Furthermore, assuming that the covariance function is only dependent of the time difference $\tau = t - t'$, the sample covariance function is given by

$$
cov(\tau) = \frac{1}{N} \sum_{t=1}^{N-\tau} (x_t - \bar{\mu})(x_{t+\tau} - \bar{\mu}),
$$

where $\bar{\mu}$ is the sample mean of the process, $\bar{\mu} = \frac{1}{N} \sum_{t=t_1}^{t_N} x_t$.

2.1.2 GP Prediction

The GP model defines a prior distribution for the target variable from the process. Let the future measurement $x_*$ be the target variable, we have

$$
x_* \sim p(x_*) = \mathcal{N}(\mu_*, \sigma_*^2),
$$
Figure 2.3. An example of learning the GP using the sample auto-covariance function (ACF). In (a), a week of NO$_2$ data is visualized. In (b), the sample ACF is estimated based on a year of NO$_2$ data. Comparing to the method of learning GP hyper-parameters, the sample ACF is a non-parametric method of modelling the covariance function of GP.

where $\mu_*$ and $\sigma^2_{N+1}$ are the mean and covariance of the variable $x_*$, respectively. Furthermore, based on the property of the GP, the measurements $x_t$ and $x_*$ follow a joint Gaussian distribution,

$$
\begin{bmatrix} x_* \\ x_t \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \mu_* \\ \mu_t \end{bmatrix}, \begin{bmatrix} \sigma_*^2 & \Sigma_{*,t} \\ \Sigma_{t,*} & \Sigma_{t,t} \end{bmatrix} \right),
$$

(2.9)

where $\Sigma_{*,t} = \Sigma_{t,*}^T$ is the cross-covariance between the variables $x_*$ and $x_t$. Given the observed data $x_t$, the posterior distribution for $x_*$ is

$$p(x_* \mid x_t) = \mathcal{N}(\mu_* \mid t, \sigma_*^2 \mid t_{N+1} \mid t_{1:N}),$$

(2.10)

where

$$\mu_* \mid t = \mu_* + \Sigma_{*,t} \Sigma_{t,t}^{-1}(x_t - \mu_t),$$

(2.11)

$$\sigma_*^2 \mid t = \sigma_*^2 - \Sigma_{*,t} \Sigma_{t,t}^{-1} \Sigma_{t,*},$$

(2.12)

are the posterior mean and variance of the variables $x_*$, respectively.

In the hyper-parameter estimation and variable prediction, it is required to invert the covariance matrix $\Sigma_{t,t}$, which has a computation complexity of $\mathcal{O}(N^3)$. When $N$ is large, GP regression becomes infeasible on resource constrained IoT devices.

2.2 Update Belief Distributions

To overcome the scalability challenge of GP regression and apply it to big IoT data analysis, we present a scalable belief update framework in the following. First, we re-derive the GP prediction (from equation 2.8 to equation 2.10), using the general framework for updating belief distributions [Bissiri et al. 2016].
2.2.1 General Belief Update

The basic idea of the general belief updating framework is to search for a belief distribution \( \hat{q}(x_*) \) of the target quantity \( x_* \), which minimize a loss function \( \ell \) specified by data \( x_t \) and regularized by a prior distribution \( p_0(x_*) \):

\[
\hat{q}(x_*) = \arg \min_{q(x_*)} \mathbb{E}[\ell(x_*; x_t)] + D_{KL}(q(x_*) \| p_0(x_*)),
\]

where \( D_{KL}(q(x_*) \| p_0(x_*)) \) is the Kullback-Leibler divergence (KLD) between the belief distribution \( q(x_*) \) and the prior distribution \( p_0(x_*) \).

Define the loss function as the negative-log likelihood

\[
\ell(x_*; x_t) = -\ln p(x_t \mid x_*).
\]

The result of the belief updating rule in equation 2.13 gives

\[
\hat{q}(x_*) = \arg \min_{q(x_*)} - \int q(x_*) \ln p(x_t \mid x_*) \, dx + \int q(x_*) \ln \frac{q(x_*)}{p_0(x_*)} \, dx
\]

\[
= \arg \min_{q(x_*)} \int q(x_*) \ln \frac{q(x_*)}{p(x_t \mid x_*)p_0(x_*)} \, dx
\]

\[
= \arg \min_{q(x_*)} D_{KL}(q(x_*) \| p(x_t \mid x_*)p_0(x_*)).
\]

Using the property of KLD that

\[
D_{KL}(q(x_*) \| p(x_t \mid x_*)p_0(x_*)) \geq 0
\]

and the minimum value of zero is achieved if and only if

\[
q(x_*) = p(x_t \mid x_*)p_0(x_*),
\]
we have
\[
\tilde{q}(x_*) = \frac{p(x_t \mid x_*)p_0(x_*)}{\int p(x_t \mid x_*)p_0(x_*)dx},
\]
which is the same result obtained by using the Bayes’ theorem.

The general belief updating framework indicates that the belief distribution in equation 2.16 is a special case when the log-negative likelihood is selected as the loss function. If the exact likelihood function is unknown or too expensive to be computed, we will need to select an alternative loss function.

### 2.2.2 Scalable Belief Update

Following the above belief updating framework, we identify that the computation bottleneck of GP regression is the evaluation of the exact likelihood with a large data. Therefore, we introduce the composite likelihood for updating the belief distribution.

To begin with, we divide the large data \( x_t \) into \( K \) segments. Note that the lengths of segments need not to be exactly the same, as long as the maximum segment length is much smaller than \( N \) and can be handled by the resource constrained IoT device. However, for the simplicity of notations, we assume that each segment has the same length of \( M = \frac{N}{K} \ll N \). Denote the following sequence as the segments of total data \( x_t \):

\[
[x_1 \ldots x_k \ldots x_K],
\]

where the \( k \)th data segment is

\[
x_k = [x_{t(k-1)M+1} \ x_{t(k-1)M+2} \ldots x_{kM}].
\]

The composite likelihood function of the variable \( x_* \) is defined as the following:

\[
\prod_{k=1}^{K} p(x_k \mid x_*).
\]

Using the negative-log composite likelihood as the loss function for the belief updating, we have

\[
\tilde{q}(x_*) = \arg \min_{q(x_*)} - \int q(x_*) \ln \prod_{k=1}^{K} p(x_k \mid x_*)dx + \int q(x_*) \ln \frac{q(x_*)}{p_0(x_*)}dx,
\]

which gives

\[
\tilde{q}(x_*) = \frac{\prod_{k=1}^{K} p(x_k \mid x_*)p_0(x_*)}{\int \prod_{k=1}^{K} p(x_k \mid x_*)p_0(x_*)dx}.
\]
The belief updating rule in equation 2.18 greatly reduces the computational complexity of GP prediction. By limiting $M$ to a small value, the computational complexity of scalable GP prediction grows linearly with the number of data segments $O(KM^3)$.

2.3 Scalable GP Regression

In this section we present scalable methods for GP hyper-parameters learning and random variables prediction.

2.3.1 Scalable GP Learning

Consider the GP hyper-parameter estimation problem in equation 2.5, a common solution is to apply the gradient-based numerical optimization [Rasmussen and Williams 2006, Chapter 5]. When the size of data $N$ is large, evaluating the gradient of the likelihood function is computationally expensive.

Following the scalable belief updating rule in equation 2.18 without the regularization term, we have

$$
\tilde{q}(\theta) = \arg\min_{q(\theta)} - \int q(\theta) \ln \prod_{k=1}^{K} p(x_k | \theta) d\theta.
$$

(2.20)

The resulting belief distribution becomes a Dirac delta function which has the infinite density at the point

$$
\tilde{\theta} = \arg\min_{\theta} - \ln \prod_{k=1}^{K} p(x_k | \theta),
$$

(2.21)

which is the maximum composite likelihood estimate of $\theta$. The computation complexity of evaluating the composite likelihood becomes $O(KM^3)$, which will be significantly less than $O(N^3)$ if $M \ll N$.

To further accelerate the hyper-parameter learning, we re-write the negative-log composite likelihood as a linear combination of log likelihoods of data segments,

$$
- \ln \prod_{k=1}^{K} p(x_k | \theta) = - \sum_{k=1}^{K} \ln p(x_k | \theta).
$$

(2.22)

And we consider the following combined estimator:

$$
\hat{\theta} = \sum_{k=1}^{K} w_k \hat{\theta}_k,
$$

(2.23)
Figure 2.5. An example of learning GP hyper-parameters using the method in equation 2.23. In (a), the spatial data of $100 \times 100$ pixels is visualized using a heat map. The data is simulated from a 2-dimensional GP with the isotropic squared exponential covariance function ($\alpha = 0.5$, $\beta = 2$). In (b), the local MLEs, combined estimation, and the true value of hyper-parameters are shown. Each local MLE is based on 1% random segmentation of the total data. The combined estimator is using the Fisher information matrices as the weights of the local MLEs.

where $\hat{\theta}_k = \arg\min_{\theta} -\ln p(x_k \mid \theta)$ is the local MLE using the $k$th data segment, and $w_k$ is the weight which shall reflect the quality of the local MLE. The structure of the combined estimator in equation 2.23 allows a distributed implementation of GP hyper-parameters learning.

Remark. Based on the result of linear combination of estimators [Kailath et al. 2000, Lemma 3.4.1], the optimal weights are the error covariance matrices of local MLEs, under the assumption of mutually uncorrelated data segments $[x_1 \ldots x_K]$. However, the assumption does not hold in general and the error covariance matrices are unknown. A near-optimal design of weights is to use the inverses of Fisher information matrices of local MLEs, which are the lower bounds of the error covariances.

2.3.2 Scalable GP Prediction

In papers I and II, we presented a recursive GP prediction method, using the scalable belief updating rule in 2.18. The recursive implementation of GP prediction follows a similar idea of the recursive Bayesian method [Särkkä 2013, Chapter 3]. We start by using the prior distribution and the first data segment to produce an updated belief distribution, and then use the updated belief as the new prior to produce the next updated belief, together with the next data segment. The update of belief is
repeated until all data segments are utilized. The detail belief update rule at the $k$th iteration is described as the following.

Let the updated belief distribution from iteration $k - 1$ be

$$
\tilde{q}(x_* | x_{1:k-1}) = \mathcal{N}(\mu_* | 1:k-1, \sigma_*^2 | 1:k-1),
$$

(2.24)

where $\mu_* | 1:k-1$ and $\sigma_*^2 | 1:k-1$ are the mean and variance of the updated belief distribution, based on the previous data segments 1 to $k - 1$. The joint conditional distribution of the new data segment $x_k$ and the variable $x_*$, given the previous data segments, is

$$
p(x_k, x_* | x_{1:k-1}) = p(x_k | x_*, x_{1:k-1})p(x_* | x_{1:k-1}).
$$

(2.25)

Replacing the posterior distribution $p(x_* | x_{1:k-1})$ with the updated belief distribution $\tilde{q}(x_* | x_{1:k-1})$ and assuming the conditional independence of $x_k$ and $x_{1:k-1}$ given $x_*$, we have

$$
p(x_k, x_* | x_{1:k-1}) \approx p(x_k | x_*)\tilde{q}(x_* | x_{1:k-1}),
$$

(2.26)

where $p(x_k | x_*) = \mathcal{N}(\mu_k | *, \Sigma_k | *)$, and

$$\begin{align*}
\mu_k | * &= \mu_k + \Sigma_{k,*} \mu_*^k (x_* - \mu_*), \\
\Sigma_k | * &= \Sigma_k - \Sigma_{k,*} \sigma_*^2 \Sigma_{*,k}.
\end{align*}
$$

(2.27)

(2.28)

Define $h_k = \Sigma_{k,*} \sigma_*^2$. Recall the properties of Gaussian distributions [Särkkä 2013, Lemma A.1], the approximated jointed distribution in equation 2.26 is given by

$$
p(x_k, x_* | x_{1:k-1}) \\
= \mathcal{N}\left(\begin{array}{c}
\mu_k + h_k (\mu_* | 1:k-1 - \mu_*) \\
\mu_* | 1:k-1
\end{array}\right), \left[\begin{array}{c}
\Sigma_k | * + h_k \sigma_*^2 | 1:k-1 h_k^\top & h_k \sigma_*^2 | 1:k-1 \\
\sigma_*^2 | 1:k-1 h_k & \sigma_*^2 | 1:k-1
\end{array}\right].
$$

(2.29)

Define $G_k = \Sigma_k | * + h_k \sigma_*^2 | 1:k-1 h_k^\top$. The updated belief distribution at iteration $k$ is

$$
\tilde{q}(x_* | x_{1:k-1}) = p(x_* | x_{1:k-1}, x_k) = \mathcal{N}(\mu_* | 1:k, \sigma_*^2 | 1:k),
$$

(2.30)

where

$$\begin{align*}
\mu_* | 1:k &= \mu_* | 1:k-1 + \sigma_*^2 | 1:k-1 h_k^\top G_k^{-1}(x_k - \mu_k - h_k (\mu_* | 1:k-1 - \mu_*)), \\
\sigma_*^2 | 1:k &= \sigma_*^2 | 1:k-1 - \sigma_*^2 | 1:k-1 h_k^\top G_k^{-1} h_k \sigma_*^2 | 1:k-1.
\end{align*}
$$

(2.31)

(2.32)

The above recursive GP prediction procedure is summarized in Algorithm 1.
Algorithm 1: Recursive GP prediction

**input:** Data segments $\mathbf{x}_{1:K}$ and the prior mean and variance $\{\mu_*, \sigma_*^2\}$

$h_1 = \mathbf{\Sigma}_{1,*} \frac{1}{\sigma_*^2}$;

$G_1 = \mathbf{\Sigma}_{1 | *} + \mathbf{h}_1 \sigma_*^2 \mathbf{h}_1^\top$;

$\mu_* | 1 = \mu_* + \sigma_*^2 \mathbf{h}_1^\top \mathbf{G}_1^{-1} (\mathbf{x}_1 - \mu_1)$;

$\sigma_*^2 | 1 = \sigma_*^2 - \sigma_*^2 \mathbf{h}_1^\top \mathbf{G}_1^{-1} \mathbf{h}_1 \sigma_*^2$;

for $k = 2 : K$ do

$h_k = \mathbf{\Sigma}_{k,*} \frac{1}{\sigma_*^2}$;

$G_k = \mathbf{\Sigma}_{k | *} + \mathbf{h}_k \sigma_*^2 \mathbf{h}_k^\top$;

$\mu_* | 1:k-1 = \mu_* | 1:k-1 + \sigma_*^2 \mathbf{h}_k^\top \mathbf{G}_k^{-1} (\mathbf{x}_k - \mu_k - \mathbf{h}_k (\mu_* | 1:k-1 - \mu_*))$;

$\sigma_*^2 | 1:k = \sigma_*^2 | 1:k-1 - \sigma_*^2 | 1:k-1 \mathbf{h}_k^\top \mathbf{G}_k^{-1} \mathbf{h}_k \sigma_*^2 | 1:k-1$;

end

**output:** The updated mean and variance $\{\mu_* | 1:K, \sigma_*^2 | 1:K\}$

2.4 Literature Review

From the perspective of statistics and probability theory, Gaussian processes are a special category of stochastic processes [Karlin et al. 1975]. A stochastic process can be characterized by its $k$-th order moment functions. For instance, the first order raw moment function is the mean function; the second order central moment function is the covariance function. A stochastic process is strictly stationary if all of its moment functions are index-invariant; a stochastic process is weakly (or wide-sense) stationary if its first and second order moment functions are index-invariant. GPs have the following three special properties:

- Any finite set of samples from a GP are jointly Gaussian distributed;
- A GP is uniquely defined by its mean and covariance function;
- If a GP is weakly stationary, then it is also strictly stationary.

Due to the above appealing properties, GPs are widely applied in various fields of studies, such as the target tracking and positioning [Zhao et al. 2019], energy consumption forecasting [van der Meer et al. 2018], finance market modeling [Han and Zhang 2015; d. Wolff et al. 2020], and remote sensing [Bazi and Melgani 2010].

Nevertheless, the scalability issue of GP learning and prediction has been an effective blocker for applying GP to large datasets. Many studies have been carried out to reduce the computation complexity of GP [Liu et al. 2020]. The most widely applied technique is known as the sparse GP [Quiñonero-Candela and Rasmussen 2005; Snelson and
Ghahramani 2006]. The sparse GP technique relies on a set of special data (pseudo inputs or inducing points). It assumes that the training and testing data are conditionally independent, given the inducing points. Intuitively, if the inducing points contain all the information from the training data, the sparse GP will recover the full GP (hence the name "sparse"). However, finding a relatively small set of inducing points that carries the maximum information from the training data is a non-trivial task [Krause et al. 2008]. Furthermore, the approximation of the conditional distribution of training data given the inducing points introduces extra information loss, which is difficult to measure quantitatively.

From the Bayesian point of view, GP prediction updates the prior distribution to a posterior distribution based on the observed data. As we discussed in this chapter, this procedure’s computational bottleneck is the evaluation of the full Gaussian likelihood with a large dataset. Bis- siri et al. [2016] presented a general belief updating framework that encapsulates the standard Bayesian prediction as a particular case. Using composite likelihoods [Varin et al. 2011] in the belief updating framework, GP prediction’s computational complexity can be significantly reduced. Composite likelihood methods are widely applied in solving inference and estimation problems [Varin and Vidoni 2005; Bevilacqua et al. 2012; Coffman et al. 2016], when the exact likelihoods are difficult or expensive to evaluate. Interestingly, the composite likelihood-based GP prediction algorithm has a very similar form of the other branches of scalable GP methods, namely the Bayesian committee machine (BCM) and the product-of-experts (PoE) [Tresp 2000; Deisenroth and Ng 2015]. Using the general belief updating framework and the composite likelihood, we can recover the BCM-style distributed GP prediction with proper weights of the prior and data likelihoods.

2.5 Summary

In the end of chapter we present a summary of papers in the thesis which are related to learning with big data:

Paper III: Composite GP

In this paper, we present a scalable GP learning and prediction method using composite likelihoods. The proposed method can be implemented in a recursive algorithm, which significantly reduces GP regression’s computation complexity. We also presented a performance analysis of the approximated GP predictive result. The impact of this work is threefold:

- The proposed scalable GP learning and prediction method is a tractable approximation of the exact GP method. The approximation is derived from a general belief updating framework using the composite likelihood, enabling quantitative analysis of the information loss.
- We show that the popular sparse GP method can also be derived from the belief updating framework, using an approximated conditional joint distribution of data and the inducing variable. This reformulation shows that sparse GP’s information loss is larger or equal to the proposed scalable GP method using composite likelihood.
- We show that GP’s predicted uncertainty with composite likelihood can be over- or underestimated, depending on whether the information of data segments is synergistic or redundant. This result provides a theoretical guideline of data segmentation in applications of scalable GP regression.

Paper VI: Air Quality

This paper applies the scalable GP to predict future 24-hour air quality based on large historical data. The dataset contains ten years of hourly measurements of nitrogen-dioxides ($\text{NO}_2$) and particulates with a diameter of smaller than 10 micrometers ($\text{PM}_{10}$), at five different monitoring stations deployed in an urban area. In this work, we presented the following results:

- A comprehensive EDA of the air quality dataset. Using the spectral power density, we show that the $\text{NO}_2$ and $\text{PM}_{10}$ in urban area have strong weekly, daily, and rush hour periodic patterns. We visualize the shift of distributions of air quality measurements when the meteorological condition changes.
- We apply the fully Bayesian scalable belief updating framework with the composite likelihood. This framework is capable of providing accurate point estimates and reliable predictive uncertainty, which is highly valuable for using the predictions in decision support systems.
We show how the Bayesian belief updating framework could be combined with deep neural networks to enhance the predictive results. We apply a deep neural network to approximate the conditional distribution of air quality measurements under different meteorological conditions, and use the result as an informative prior for the Bayesian belief updating framework.
3. Handling Missing Data

In this chapter, we discuss the second challenge of IoT data analysis—missing data. We discuss the missing data in two contexts: learning classifiers using unlabeled data and predicting new outcomes with missing features. We want to achieve reliability in terms of accurate estimation of predictive error probability for the classification task. For the prediction problem, we want to achieve robustness against rare events in unobserved features.

Roadmap
Chapter 3 is divided into two sections. In 3.1 we start by discussing different patterns of missing labels. Then we introduce a machine learning paradigm for learning models using unlabeled data, namely semi-supervised learning (SSL). We discuss how SSL works under different assumptions of missing labels. Finally, we present a new SSL approach that provides reliable predictive error probability under a relaxed assumption about the missing labels. Chapter 3.1 provides background and summarized results for Paper III. In 3.2, we present three types of predictors when using data with missing features: the optimistic predictor, the constrained predictor, and the robust predictor. We show how the proposed robust predictor achieves low mean squared errors when rare events happen in missing features, while only incurring a small loss to the overall performance. Chapter 3.2 presents results from Paper IV in a pedagogical way.

3.1 Missing Labels

In this section, we present a semi-supervised learning method which provides reliable predictive error probabilities. Consider the following classification problem. A training dataset is given, \( D_{\text{labeled}} = \{(x_i, y_i)\} \), where \( x_i \in \mathbb{R}^d \) is the vector of features of training sample \( i \), and \( y_i \in \{1, \ldots, K\} \) is the corresponding label. The task is to learn a classifier \( f \), such that

\[
\hat{y}_* = f(x_*),
\]

is the predicted label of the testing data \( x_* \). The labeled data is often expensive to obtain, whereas the unlabeled data is readily available. Suppose that the size of labeled data, \( |D_{\text{labeled}}| \), is small. Another unlabeled dataset \( D_{\text{unlabeled}} = \{(x_j)\} \) is available and \( |D_{\text{unlabeled}}| \gg |D_{\text{labeled}}| \).
3.1.1 Missing Labels Patterns

To begin with, we raise the following question: is there any pattern of missing labels? In other words, does the data distribution in $D_{\text{labeled}}$ match the distribution in $D_{\text{unlabeled}}$? Next we discuss the three patterns of missing labels.

**Missing Completely at Random (MCAR)**

Define a binary variable $l$ for each data sample, such that

$$l = \begin{cases} 
0 & \text{this sample is unlabeled;} \\
1 & \text{this sample is labeled.} 
\end{cases} \quad (3.2)$$

In semi-supervised learning, the MCAR assumption states that the data distributions are identical in the labeled and unlabeled datasets:

$$p(x, y \mid l = 0) = p(x, y \mid l = 1). \quad (3.3)$$

The MCAR is a very strict assumption. In real-world, it is very rare that the small set of labeled data are uniformly distributed in the entire feature space.

**Missing at Random (MAR)**

The MAR assumption does not force the marginal distributions of features to be identical in the labeled and unlabeled datasets. That is, in general,

$$p(x \mid l = 0) \neq p(x \mid l = 1). \quad (3.4)$$

Instead, the MAR states that the conditional distributions of labels given specific features are identical in the labeled and unlabeled datasets:

$$p(y \mid x, l = 0) = p(y \mid x, l = 1). \quad (3.5)$$

The MAR assumption admits that the missing label may have a feature-selective pattern. For example, a sub-population with certain features are more frequently being labeled. Under the MAR assumption, it is also possible that a class of data with certain features are not labeled at all and hence not being observed.

**Missing Not at Random (MNAR)**

The MNAR assumption states that both the marginal distributions of features and the conditional distributions of labels are generally non-identical in the labeled and unlabeled datasets:

$$p(x \mid l = 0) \neq p(x \mid l = 1), \quad (3.6)$$

$$p(y \mid x, l = 0) \neq p(y \mid x, l = 1). \quad (3.7)$$
Figure 3.1. An example of different missing label mechanism in semi-supervised learning: MCAR, MAR, and MNAR. In (a), the full training dataset is visualized in the feature space of $x_1$ and $x_2$, where red circles represent data of class 0 and blue crosses represents class 1. The unlabeled training data are represented by the gray dots. In (b), the MCAR mechanism samples the training data uniformly in the feature space and the classes. That is, $p(x, y \mid l = 0) = p(x, y \mid l = 1)$. In (c), the data in the forth quadrant are not included in the labeled data, and therefore $p(x \mid l = 0) \neq p(x \mid l = 1)$. Meanwhile, the rest of the data are sampled without bias to the classes, $p(y \mid x, l = 0) = p(y \mid x, l = 1)$. In (c), not only that the missing label mechanism is selective with respect to the feature space $p(x \mid l = 0) \neq p(x \mid l = 1)$, but also biased to the classes $p(y \mid x, l = 0) \neq p(y \mid x, l = 1)$. 
In other words, the labeling process is feature-selective and label-selective. In this case, it is impossible to learn any useful model from the data without the information of the data labeling process. For example, given data with the same feature, only a certain class is being labeled. The dataset will therefore produce a model which predicts that all future data with this feature belong to the labeled class.

3.1.2 Semi-Supervised Learning

Semi-supervised learning (SSL) is a category of machine learning methods which use both labeled and unlabeled datasets. In our classification problem, the aim is to learn a classifier by using both $D_{\text{labeled}}$ and $D_{\text{unlabeled}}$.

When Does SSL Work?

A fundamental question in SSL is when the unlabeled data can be used to enhance a model learned by supervised methods. Many SSL literature are based on the famous smoothness assumption [Chapelle et al. 2006, Chapter 1.2]: ”If two points $x_1$ and $x_2$ in a high-density region are close, then so should be the corresponding output $y_1$ and $y_2$”. However, this is a rather vague and heuristic claim. A recent work by Oliver et al. shows that the performance of SSL decreases when the data distributions of the labeled data and the unlabeled data mismatch [Oliver et al. 2018], which indicates that the fundamental question of when SSL will work shall be linked to the missing data patterns.

From the missing data point of view, it is self-evident that SSL does not work when the labels are MNAR. In fact, even supervised methods will give misleading results when the labels are MNAR. We need at least MAR equation 3.5 to hold for learning a classifier which can predict testing data.

How to Use Unlabeled Data?

Under the MAR assumption, the next question is how to use unlabeled data to enhance the classifier learned by supervised methods?

Consider the Bayesian optimal classifier

$$\hat{y} = \arg \max_y p(y \mid x), \quad (3.8)$$

where

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}. \quad (3.9)$$

The probability densities $p(x \mid y)$, $p(y)$, and $p(x)$ are of course unknown and need to be learned from data. The unlabeled data obviously cannot be used to learn $p(y)$, but it is very useful for learning a better $p(x)$.
Can unlabeled data be used to learn $p(x \mid y)$? This is however non-trivial to answer. At the first look, it is not obvious that how $D_{\text{unlabeled}} = \{(x_j)\}$ contribute to the conditional distribution of features given a label. Recall that the MAR assumption states that

$$p(y \mid x, l = 0) = p(y \mid x, l = 1).$$

When we observe a sample $(x_j)$ from $D_{\text{unlabeled}}$ which is very close to a sample $(x_i, y_i)$ from $D_{\text{labeled}}$, we may infer the missing label of $(x_j)$ by using

$$q(y \mid x_j), \quad (3.10)$$

which is the initial probabilistic model learned by using $D_{\text{labeled}}$ and supervised methods. Using the information from $q(y \mid x_j)$, we might be able to use $(x_j)$ to enhance $p(x \mid y)$. This idea is similar to the self-labeling or self-training methods in SSL literature.

However, the above raises two open questions which are not answered in the SSL literature:

- How to measure the closeness between a sample $(x_j)$ from $D_{\text{unlabeled}}$ and a sample $(x_i, y_i)$ from $D_{\text{labeled}}$? In other words, what is the rule to determine whether $(x_j)$ is useful or not for learning $p(x \mid y)$?
- Given that $(x_j)$ is useful for learning $p(x \mid y)$, how to use the information in $q(y \mid x_j)$ while preserve the uncertainty?

The second question leads to the concept of reliability of a classifier, which is a larger topic and will be discussed in the next subsection. In the rest of this subsection, we focus on answering the first question by defining a label-informative region in the feature space.

**Selective Self-Labeling**

Let

$$q(x \mid y, l = 1), \quad (3.11)$$

be the initial generative models learned for all classes $y \in \{1, \ldots, K\}$, using supervised learning methods with the labeled data. Furthermore, we can also learn a generative model for the unlabeled data,

$$q(x \mid l = 0). \quad (3.12)$$

To answer the question of whether $(x_j)$ is useful or not for learning $p(x \mid y)$, the key is to test whether $(x_j)$ is generated from the same distributions as the labeled data. Mathematically, if

$$\max_{y \in \{1, \ldots, K\}} \ln \frac{q(x_j \mid y, l = 1)}{q(x_j \mid l = 0)} > \kappa, \quad (3.13)$$

where $\kappa$ is a decision threshold, then $(x_j)$ is generated from the same distributions as the labeled data; otherwise, $(x_j)$ is generated from a
Figure 3.2. An example of selective-self labelling with MAR data. In (a), the labelled and unlabelled data are visualized. The class 0 data are represented by red circles; the class 1 data are represented by blue crosses; the unlabeled data are represented by gray dots. The data consists of three clusters in the feature space, where the upper cluster is a mixture of class 1 and 0. In (b), the label-informative region in equation 3.14 is visualized. By setting $\kappa = 0$, only the unlabeled data in the bottom left and right clusters are considered useful for enhancing the initial models learned from the labeled training data.

Different distribution and therefore cannot be used to enhance the initial generative models.

Using the decision rule in equation 3.13, we partition the feature space and define the following label-informative region:

$$
D_{\text{informative}} = \{ (x_j) \mid \max_{y \in \{1, \ldots, K\}} \ln \frac{q(x_j \mid y, l = 1)}{q(x_j \mid l = 0)} > \kappa \}. \quad (3.14)
$$

For any $x_j \in D_{\text{informative}}$, the initial probabilistic model $q(y \mid x_j)$ can be applied to infer the missing label.

3.1.3 Reliability of Error Probabilities

Before we proceed to the task using $x_j \in D_{\text{informative}}$ to enhance the initial probabilistic model $q(y \mid x_j)$, we introduce the concept of reliability.

**Definition of Reliability**

When applying data analysis or machine learning to decision support systems, we often desire predictive results with uncertainty measures. For example, machine learning used to support diagnosis of patients by analyzing X-ray images [Majkowska et al. 2020]. In safety-critical applications, we prefer the computer system gives a probabilistic result, rather than a yes-or-no answer.
In case of classification, the uncertainty of a result can be measured by the predictive error probability:

\[ q_e(x_*) = 1 - q(\hat{y}_* \mid x_*), \]  
(3.15)

where \( q(\hat{y}_* \mid x_*) \) is the predictive posterior learned from the data. Note that \( q_e(x_*) \) is only an estimation of the following true error probability:

\[ p_e(x_*) = \Pr\{y_* \neq \hat{y}_* \mid x_*\}. \]  
(3.16)

Intuitively, a definition of reliability is by using the mean squared error between the estimated and the true error probabilities:

\[ \mathbb{E}[(q_e(x_*) - p_e(x_*))^2]. \]  
(3.17)

A small difference between \( q_e(x_*) \) and \( p_e(x_*) \) indicates a reliable predictive error probability \( q_e(x_*) \).

However, since \( p_e(x_*) \) is unknown, we cannot use the above metric to measure the reliability of a predictive error probability. Instead, we can use the Brier score

\[ B(q_e) = \mathbb{E}[(q_e(x_*) - 1(y_* \neq \hat{y}_* \mid x_*))^2], \]  
(3.18)

where \( 1(\cdot) \) is an indicator function. Note that the Brier score can be decomposed into two terms:

\[
B(q_e) = \mathbb{E}[(q_e(x_*) - 1(y_* \neq \hat{y}_* \mid x_*))^2] = \mathbb{E}[(q_e(x_*) - p_e(x_*))^2] + \mathbb{E}[p_e(x_*)(1 - p_e(x_*))]. \]  
(3.19)

Therefore, the Brier score is lower bounded by \( \mathbb{E}[p_e(x_*)(1 - p_e(x_*))] \).

**Reliable SSL**

Finally, we present a SSL method which aims to provide reliable predictive error probabilities, when the labels are MAR.

For \( x_j \in \mathcal{D}_{\text{informative}} \), we can infer the missing label of \( x_j \) by using the initial model \( q(y \mid x_j) \). However, if we consider

\[ \hat{y}_j = \arg\max_y q(y \mid x_j), \]  
(3.20)

as the predicted label and use \((x_j, \hat{y}_j)\) as a new labeled data, the uncertainty in \( q(y \mid x_j) \) will be lost. The new model learned based on the self-labeled data will therefore systematically underestimate the uncertainty and give unreliable predictive error probabilities.

Therefore, instead of the deterministic \( \hat{y}_j \), we consider the following labels

\[ \tilde{y}_j \sim q(y \mid x_j), \quad \forall x_j \in \mathcal{D}_{\text{informative}}, \]  
(3.21)
Figure 3.3. An example of the underestimated uncertainty caused by deterministic self-labelling in SSL. In (a), the training data with both labelled and unlabeled points are visualized. The data are binary and overlapped to some extent. The class 0 data are represented by red circles; the class 1 data are represented by blue crosses; the unlabeled data are represented by gray dots. In (b), the predictive probability of class 1 is visualized, using the deterministic self-labelling rule in SSL. The uncertainty levels along the decision border are underestimated. In (c), the predictive probability recognizes the overlapping of two classes of data, which is achieved by the stochastic self-labelling strategy.
which are draws from the initial probabilistic model. Note that when
the size of $D_{\text{informative}}$ is large, this method becomes the well known mul-
tiple imputation method [Little and Rubin 2019]. See Figure 3.3 for an
example of the difference between the deterministic and the stochastic
self-labelling strategies.

For $x_j \in D_{\text{unlabeled}} \setminus D_{\text{informative}}$, the initial model cannot be applied
to infer the missing labels. Therefore we assign $\tilde{y}_j \sim q(y)$ for data
outside of the informative region.

### 3.2 Missing Features

In this section, we discuss the problem of missing features in testing
data. We present a predictor which is robust to rare events in missing
features.

Consider random variables $x \in \mathbb{R}^d$, $y \in \mathbb{R}$, $z \in \mathbb{R}^q$ with a joint
probability distribution $p(x, y, z)$ and $d > q$. Given observations $D = \{(x_i, y_i, z_i) | i = 1, \ldots, N\}$, the task is to predict $y_*$ based on only $x_*$,
when $z_*$ is missing. The minimum mean square error (MMSE) predictor
for $y_*$ is given by

$$
\hat{y}_* = \int y_* p(y_*, z_* | x_*) dy_* dz_*
$$

(3.22)

where $p(y_*, z_* | x_*)$ need to be learned from $D$. The problem with the
above prediction is that, the missing feature $z_*$ has been marginalized
out and therefore its impact on $y_*$ is not considered. Although the
above predictor achieves the best performance on average, the condi-
tional MSE given a specific value of $z_*$ may vary significantly, especially
when $p(z)$ is a fat-tailed distribution.

#### 3.2.1 Optimistic Predictor

We refer the predictor in equation 3.22 as the optimistic predictor. The
optimistic predictor aims to minimize the loss on average and does not
consider the possible large conditional loss given a rare value of $z_*$ –
hence its name. In this thesis, we consider a linear MMSE predictor

$$
\hat{y}_* = w^\top \phi(x_*).
$$

(3.23)

Without loss of generality, let $\phi(x_*) = x_*$. The optimistic predictor is
given by

$$
\hat{y}_* = w_o^\top x_*,
$$

(3.24)

where $w_o^\top = \Sigma_{y,x} \Sigma_{x,x}^{-1}$. 

42
Equivalence between $w_o$ and the two-stage LMMSE predictor
A natural thought would be to firstly predict $z_*$ by using $x_*$, and then predict $y_*$ by using $x_*$ and $\hat{z}_*$. In the following we show that the two-stage LMMSE predictor is equivalent to the predictor in equation 3.24.

If $z_*$ is not missing, the linear MMSE predictor which uses $[x_*^\top z_*^\top]$ is given by

$$
\hat{y}_* = [\alpha^\top \beta^\top] \begin{bmatrix} x_* \\ z_* \end{bmatrix},
$$

(3.25)

where

$$
[\alpha^\top \beta^\top] = \left[\Sigma_{y,x} \quad \Sigma_{y,z}\right] \left[\Sigma_{x,x} \quad \Sigma_{x,z}\right]^{-1}.
$$

(3.26)

The LMMSE predictor for the missing $z_*$ is

$$
\hat{z}_* = \Sigma_{z,x} \Sigma_{x,x}^{-1} x_*.
$$

(3.27)

The two-stage predictor for $y_*$ is therefore given by

$$
\hat{y}_* = [\alpha^\top \beta^\top] \begin{bmatrix} x_* \\ \Sigma_{z,x} \Sigma_{x,x}^{-1} x_* \end{bmatrix}
$$

(3.28)

Applying the block matrix inversion, it is readily seen that

$$
\hat{y}_* = [\alpha^\top \beta^\top] \begin{bmatrix} x_* \\ \hat{z}_* \end{bmatrix}
$$

(3.29)

Finite-sample Case
When the covariance matrices $\Sigma_{y,x}$ and $\Sigma_{x,x}$ are unknown, the LMMSE predictor is

$$
w_o = \arg \min_w ||y - w^\top X||^2, \quad (3.30)
$$

where $X = [x_1 \ldots x_N]$ and $y = [y_1 \ldots y_N]^\top$.

In the case of high dimensional and sparse features, regularization techniques can be applied to improve the quality of the finite-sample LMMSE predictor. For example, applying the method in [Zachariah and Stoica 2015], the regularized finite-sample LMMSE predictor is

$$
w_o = \arg \min_w ||y - w^\top X||^2 + ||Dw||_1, \quad (3.31)
$$

where $D$ is a diagonal matrix with the elements

$$
\{D_j = \sqrt{\frac{\|[x_j^1 \ldots x_j^N]\|^2}{N}} |j = 1, \ldots, d + q\}.
$$
3.2.2 Constrained Predictor

Next, we want to design a predictor such that the error of the predictor is linearly independent of the value of the missing feature $z_\ast$.

To begin with, decompose $y_\ast$ as

$$y_\ast = [\alpha^\top \beta^\top] [x_\ast \ z_\ast] + v$$

(3.32)

where $v$ is the residual which is orthogonal to the LMMSE predictor. To identify the impact of $z_\ast$ on the prediction error $\epsilon$, decompose $x_\ast$ as

$$x_\ast = \Gamma^\top z_\ast + u,$$

(3.33)

where $\Gamma^\top = \Sigma_{x,z} \Sigma_{z, z}^{-1}$, and $u$ is the orthogonal residual. Plugging in equation 3.33 into equation 3.32, we have

$$\epsilon = y_\ast - w^\top x_\ast$$

$$= \alpha^\top (\Gamma^\top z_\ast + u) + \beta^\top z_\ast + v - w^\top (\Gamma^\top z_\ast + u)$$

(3.34)

Therefore, to ensure the prediction error is orthogonal to the missing feature $z_\ast$, we have the following linear constraint on the predictor:

$$(\alpha^\top - w^\top) \Gamma^\top + \beta^\top = 0.$$  

(3.35)

The optimal linear predictor which subjects to the constraint of orthogonality is obtained by solving the following optimization problem:

$$\text{minimize } w \ E[(y - w^\top x)^2],$$

subject to $(\alpha^\top - w^\top) \Gamma^\top + \beta^\top = 0$. 

(3.36)

The objective function in the above problem can be written as

$$E[(y - w^\top x)^2] = E[(y)^2] - 2E[y w^\top x] + E[(w^\top x)(w^\top x)^\top]$$

(3.37)

which is a convex quadratic function of $w$ ($\Sigma_{x,x}$ is positive definite by definition). The orthogonality constraint is a linear equality constraint of $w$:

$$\Gamma w = \Gamma \alpha + \beta.$$  

(3.38)

Note that $w \in \mathbb{R}^d$, $\Gamma \in \mathbb{R}^{q\times d}$, and $q < d$.

Finally, the conservative predictor, $w_c$, is obtained by solving the following constrained quadratic minimization problem [Boyd and Vandenberghe 2004, Chapter 10]

$$\text{minimize } w^\top \Sigma_{x,x} w - 2w^\top \Sigma_{x,y},$$

subject to $\Gamma w = \Gamma \alpha + \beta$. 

(3.39)
Finite-sample Case

When the covariance matrices are unknown, the constraint of orthogonality to features $z$ becomes

$$
E[z(y - w^T x)] = \frac{1}{N} \sum_{j}^{N} z_i (y_i - w^T x_i) = 0.
$$

(3.40)

In other words, the projection of error vector to the space of feature $z$ is zero:

$$
\langle Z, y - w^T X \rangle = 0.
$$

(3.41)

Therefore the constrained predictor in the finite-sample case is obtained by solving the following problem

$$
\min_{w} \quad ||y - w^T X||^2 + ||Dw||_1,
$$

subject to

$$
\langle Z, y - w^T X \rangle = 0.
$$

(3.42)

Note that the orthogonality constraint needs to be regularized for the purpose of robustness to the sparse and high dimensional features.

Assume that that data has real values, the orthogonality constraint can also be written as

$$
\underbrace{Z}_{q \times N} \underbrace{(y - w^T X)^T}_{N \times 1} = \underbrace{0}_{q \times 1},
$$

(3.43)

and

$$
\underbrace{ZX^T}_{q \times d} \underbrace{w}_{d \times 1} = \underbrace{Zy^T}_{q \times 1}.
$$

(3.44)

Recall that $q \leq d$ must hold for the linear system to have at least one solution.

Re-parametrization

Using the result from [James 1978], we can represent all solutions of the linear system equation 3.44 as the following:

$$
w = (ZX^T)^\dagger Zy^T + [I - (ZX^T)^\dagger(ZX^T)]\theta,
$$

(3.45)

where $(\cdot)^\dagger$ is the pseudoinverse of a matrix and $\theta$ is an arbitrary $d \times 1$ vector. Thereafter, by plugging in the above representation of $w$ and applying the reparameterizing trick, the finite-sample conservative predictor can be obtained by solving the following unconstrained optimization problem

$$
\min_{\theta} \quad ||y - ((M)^\dagger Zy^T)^T X - \theta^T[I - (M)^\dagger(M)]^T X||,
$$

(3.46)

where $M = ZX^T$.  

45
3.2.3 Robust Predictor

The drawback of the conservative predictor is, to ensure the linear independence of the prediction error and the value of the missing feature, the incurred error around the average value of the missing feature is large. In the following we propose a robust predictor which suppresses the error caused by outlying missing features, while incurring only a small error for typical missing features.

First we define the tail region

$$ \mathcal{Z}_\alpha = \left\{ z : \| z - \mu_z \|_{\Sigma_z^\dagger} \geq \sqrt{\text{rank}(\Sigma_z)}/\alpha \right\}, $$

such that the probability of $z$ falling within it is bounded as $\Pr\{ z \in \mathcal{Z}_\alpha \} \leq \alpha$. The MSE can therefore be decomposed into two parts:

- The MSE when $z \in \mathcal{Z}_\alpha$: $\text{MSE}_\alpha$;
- The MSE when $z \notin \mathcal{Z}_\alpha$: $\text{MSE}_{1-\alpha}$.

And the overall MSE can be expressed as

$$ \text{MSE} = \Pr\{ z \in \mathcal{Z}_\alpha \} \text{MSE}_\alpha + \Pr\{ z \notin \mathcal{Z}_\alpha \} \text{MSE}_{1-\alpha}. $$

The optimistic predictor $w_o$, minimizes the overall MSE. However, the $\text{MSE}_\alpha$ can be very large. The conservative predictor $w_c$, on the other hand, suppresses the $\text{MSE}_\alpha$. But the increase of overall MSE by $w_c$ can be significant.

The robust predictor $w$ is a probabilistic combination of $w_o$ and $w_c$:

$$ w(x) = \Pr\{ z \in \mathcal{Z}_\alpha \mid x \} w_c + \Pr\{ z \notin \mathcal{Z}_\alpha \mid x \} w_o. $$

The conditional probabilities, $\Pr\{ z \in \mathcal{Z}_\alpha \mid x \}$ and $\Pr\{ z \notin \mathcal{Z}_\alpha \mid x \}$, are unknown and need to be learned from the training data. We model the conditional probabilities using the logistic regression

$$ \hat{\Pr}\{ z \in \mathcal{Z}_\alpha \mid x \} = \frac{1}{1 + \exp \kappa (\delta(x) - \delta_0)}, $$

where

$$ \delta(x) = \sqrt{\hat{z}^\top(x) E_n[zz^\top] \hat{z}(x)} \geq 0 $$

is the Mahalanobis distance between the predicted $z$ based on $x$ and the marginal mean of $z$ (zero in this case). The parameters $\kappa$ and $\delta_0$ in the logistic regression are estimated by minimizing the binary entropy loss function,

$$ \min_{\kappa, \delta_0} -E_n \left[ I(z \in \mathcal{Z}_\alpha) \ln \hat{\Pr}\{ z \in \mathcal{Z}_\alpha \mid x \} + I(z \notin \mathcal{Z}_\alpha) \ln (1 - \hat{\Pr}\{ z \in \mathcal{Z}_\alpha \mid x \}) \right]. $$
3.3 Literature Review

Missing data is a very common problem in statistical data analysis [Little and Rubin 2019], especially in the scenario of IoT. Depending on which parts of the data are missing, the challenge of dealing with missing data can be formulated into different problems.

Semi-supervised Learning

When the labels are missing in the training data, learning the model with incomplete data is known as the semi-supervised learning (SSL) [Chapelle et al. 2006]. Although existing SSL methods achieve accurate image classifications on benchmark datasets, recent studies show that many methods failed in more realistic scenarios [Oliver et al. 2018]. Furthermore, Ovadia et al. [2019] showed that many modern machine learning models failed to provide reliable uncertainty measures. Therefore, two fundamental questions about SSL are: first, when the unlabeled data are useful to enhance the initial model learned from the labeled data; and second, how to use the unlabeled data such that the final model can provide reliable uncertainty measures, which are essential for safety-critical tasks.

Historically SSL methods are categorized as self-training, co-training, and graph-based methods [Zhu 2005]. Self-training refers to the method which iteratively labels the unlabeled data and refines the model [Li et al. 2008]. Co-training refers to a type of method that trains two initial models based on the split features. It utilizes the most confident prediction results on the unlabeled data from each other to refine their models [Zhou and Li 2007]. The graph-based SSL methods define a network of the labeled and unlabeled data and assume that the points with a smaller distance in the feature space will be more likely to have the same labels [Zhu et al. 2005; Yang et al. 2016].

The more recent development of SSL adopts the deep neural networks. Kingma et al. [2014] presents a framework of SSL which uses the variational auto-encoder to compress the raw data into a latent feature space. In the low-dimensional latent feature space, statistical models are trained for classification. Inspired by this framework, several SSL methods are proposed in the literature, such as the auxiliary deep generative model [Maaløe et al. 2015, 2016] and the generative adversarial networks (GAN) Kumar et al. [2017].

However, very few SSL works have investigated the problem when the distributions of labeled and unlabeled data are different. To the best of our knowledge, [Chawla and Karakoulas 2005] is the first published work that addresses the problem of different distributions between labeled
and unlabeled data. Furthermore, very few studies have examined SSL methods in terms of the reliability of their predictive uncertainties.

Robust Prediction with Missing Features

Robustness can refer to various properties in different statistical data analysis problems. Huber [1964] defines the robustness as the asymptotic variance of an estimator when the data distribution is contaminated. In machine learning, robustness can refer to the robustness against noise or outliers in the training data, and the ability to avoid over-fitting [Ren et al. 2018]. Another category of robust learning refers to the ability to deal with adversarial samples in training or testing data [Najafi et al. 2019; Awasthi et al. 2019]. Fewer works have investigated the robustness against rare events in missing testing data. Rothenhäusler et al. [2018] proposed the anchor regression, which can be applied to robust predictions when certain exogenous variables are only available in the training data. Fletcher Mercaldo and Blume [2020] studies different missing data patterns and propose the pattern submodels for prediction.

3.4 Summary

Next we present a summary of papers related to missing data. The following two papers are included:


Paper III: Semi-supervised Learning

In this paper, we present a semi-supervised learning (SSL) method for classification when the labels are missing at random (MAR). We focus on the feature of providing accurate estimates of predictive error probabilities. The proposed method is evaluated using the Brier score. This work has an impact on the research of SSL in two aspects.

- We emphasize the importance of accurate estimation of error probability, which is a measure of uncertainty for predicted output given by a SSL method. Obtaining a reliable uncertainty measure is important for using SSL, or in general machine learning,
in decision support for safety-critical applications. We propose an SSL approach that is capable of providing accurate estimates of error probabilities under MAR, where standard assumptions break down.

- We investigate the fundamental limitation of SSL, which is determined by the missing data mechanism. We stated clearly which part of unlabelled data are useful and how they can be used to enhance a supervised classifier. We show that unlabelled data can lead to a systematically underestimated error probabilities when using standard missing data assumptions (specifically, MCAR).

Paper IV: Robust Prediction

In this paper, we study the problem of prediction with incomplete features. We present a robust predictor which suppresses the prediction error when the missing features have rare values while incurring only a small price to the overall error. In this work, we investigated the properties of three types of predictors:

- The optimistic predictor minimizes the prediction error on average. However, the conditional error given a rare value of the missing feature can be very large. We show that the two-stage predictor – first predicting the missing features, and then the target variable – is equivalent to the optimistic predictor which only uses the available features.

- The conservative predictor poses a constraint that the prediction error shall be linearly independent of the missing features. The conservative predictor is robust to the rare values of missing features but pays a significant price to the overall prediction error.

- The robust predictor interpolates the optimistic and conservative predictors. The interpolation uses a probabilistic model of whether the missing feature will fall into the tail region of its distribution, given the observed features. The proposed predictor is robust to the missing features’ rare values while incurring only a small price to the overall prediction error.
4. Detecting False Data

In Chapter 4, we discuss the third challenge of IoT data analysis – false data. False data can severely mislead the modeling and decision-making process. We present results about false data detection and secure information fusion for distributed IoT systems.

Roadmap
We start Chapter 4 by discussing distributed state estimation. We focus on a user case where a distributed IoT system is deployed to monitor a spatial-temporal signal. In 4.1, we present a consensus-based distributed estimation method. We show that the consensus fusion method can be considered a particular case of the general belief updating framework introduced in Chapter 2. In 4.2, we first introduce the false data detection problem and then present a new false data detector for distributed IoT systems. Chapter 4 presents a summary of the ideas and results from Paper V and VI.

4.1 Distributed State Estimation
In this section, we present fundamental elements of distributed state estimation for IoT systems. We focus on the distributed state estimation problem with probabilistic and dynamic networks. We discuss the distributed path planning strategies for mobile IoT systems to maximize the information gain of collected data.

4.1.1 Distributed State Estimation
Consider a network of nodes deployed to monitor a spatial-temporal signal in a service area. There are three important elements in this system: the dynamic of the signal, the sensor measurements, and the communication network.

Spatial-Temporal Signal
Spatial-temporal signals are a type of signal which is dynamic in both space and time domain. Many real-world phenomena can be modelled by spatial-temporal signals, for instance, the air pollution level in a city,
the traffic volume in a road network, and the mobile data demand in a region. In those phenomena, the values of a signal at different locations and times are correlated via its spatial and temporal dynamics.

Let us simplify the spatial-temporal signal to a grid and a time index with fixed spatial and temporal sampling rate. Hence the signal is discrete in both space $s$ and time $t$. For the notation simplicity we vectorize the signal at a given time $t$, and denote

$$x_t = [x_1^t \ x_2^t \ \ldots \ x_N^t]^\top$$

as the signal values at $s = 1 : N$ space locations and time $t$.

Next we present a linear state-space model for the spatial-temporal signal. Consider the following linear equation

$$x_t = Ax_{t-1} + w_t,$$  

(4.2)

where $A$ is the (temporal) dynamic matrix and $w_t \in \mathbb{R}^N$ is a random input at time $t$. The spatial dynamic is encoded in the covariance matrix of $w_t$: $\Sigma_w$, where the $(i,j)$ element of the matrix is determined by the spatial covariance between locations $s_i$ and $s_j$. For instance, a covariance function decreases exponentially according to the Euclidean distance between locations:

$$\text{cov}(s_i, s_j) = \alpha^2 \exp(-\frac{||s_i - s_j||_2}{\theta^2}),$$

(4.3)

where the parameters $\alpha$ and $\theta$ can be learned offline from the historical data.

**Local Sensor Measurements**

Each node in the network is equipped with a sensor to collect data for monitoring the spatial-temporal signal. Considering the mobility of the
Figure 4.2. An example of spatial temporal signals. The signal $x_t$ varies at different locations in a 2 dimensional space, and changes with time steps. In the time domain, the dynamic of the signal is following the linear equation in equation 4.2; in the space domain, the covariance of the signal is described by the squared exponential function in equation 4.3.

node, the measurement matrix $H_{i,t} \in \mathbb{R}^{M \times N}$ is time-variant. The local sensor measurement of node $i$ at time $t$, $y_{i,t} \in \mathbb{R}^M$, is therefore given by

$$y_{i,t} = H_{i,t}x_t + e_{i,t}, \quad (4.4)$$

where $e_{i,t} \in \mathbb{R}^M$ is the sensor noise.

Based on the local sensor measurement, a node can only estimate the spatial-temporal signal at nearby locations, as the spatial covariance of signals quickly decreases to zero at far distances.

**Dynamic and Probabilistic Communication Network**

The nodes communicate with each other in a probabilistic manner. For instance, the success data transmission rate is exponentially decreasing as the distance between a pair of nodes increases,

$$\Pr(i,j) = \exp(-\lambda ||s_i - s_j||_2). \quad (4.5)$$

Furthermore, the above probability is time-variant due to the mobility of nodes. Therefore, the communication network is a dynamic and random graph $G_t = (\mathcal{V}, \mathcal{E}_t)$, where $\mathcal{V} = \{1, \ldots, N\}$ is the set of nodes and $\mathcal{E}_t$ is the set of communication links. The probability $\Pr((i,j) \in \mathcal{E}_t)$ depends on the locations of nodes $i$ and $j$ at time $t$.

**Centralized and Distributed Kalman Filters**

When there is a centralized node which collects data from all nodes, the state-estimation problem can be solved by using the centralized Kalman filter (KF), which is the linear minimum mean squared error (LMMSE) estimator. The Kalman filter can also be derived as a recursive Bayesian estimator [Särkkä 2013], which has a closed form solution for linear Gaussian systems.
In distributed networks, a node only communicates with its neighbors. Different variants of distributed KFs have been proposed in the literature, including the consensus based and the diffusion based KFs [Cattivelli and Sayed 2010; Battistelli and Chisci 2014]. However, most of the distributed KFs are designed for static network of trusted nodes. Few works have considered distributed information fusion for dynamic networks with potential false data. In the remaining of this section, we will present a distributed information fusion strategy which is suitable for dynamic networks and can be integrated with distributed false data detectors.

4.1.2 Consensus Based Distributed Estimation

Next we present a distributed information fusion method based on formulating the consensus of local estimates. We first reformulate the dynamic of signal equation 4.2 and the sensor measurement equation 4.4 with the following general probabilistic state-space model [Särkkä 2013]:

\[
\begin{align*}
    x_t & \sim p(x_t \mid x_{t-1}), \\
    y_{i,t} & \sim p(y_{i,t} \mid x_{t-1}),
\end{align*}
\]

where \( p(x_t \mid x_{t-1}) \) is the global dynamic model and \( p(y_{i,t} \mid x_{t-1}) \) is the local measurement model for node \( i \).

**Consensus Formulation on Local Belief Distributions**

Let \( p_i^-(x_t) \), \( p_i(x_t) \), and \( p_i^+(x_t) \) be the local predictive, filtering, and consensus distributions of node \( i \) at time \( t \). The general consensus based distributed estimation consists of the following three steps at time \( t \) [Battistelli and Chisci 2014].

- **Local Prediction**: Given the consensus distribution \( p_i^+(x_{t-1}) \) of node \( i \) at time \( t - 1 \), the local predictive distribution \( p_i^-(x_t) \) is given by the Chapman–Kolmogorov equation,

\[
p_i^-(x_t) = \int p(x_t \mid x_{t-1})p_i^+(x_{t-1})d x_{t}.
\]

- **Local filtering**: Given the latest local sensor measurement \( y_{i,t} \) of node \( i \) at time \( t \), the local filtering distribution \( p_i(x_t) \) is given by the Bayes’ rule,

\[
p_i(x_t) = \frac{p(y_{i,t} \mid x_t)p_i^-(x_t)}{p(y_{i,t})}.
\]

- **Consensus formulation**: Starting from the local filter distributions \( p_{i,t=0}^+(x_t) = p_i(x_t) \), node \( i \) exchanges information with its neigh-
bors \( \mathcal{N}_{i,t} \) and calculate the consensus distributions for \( L \) iterations,

\[
p_{i,t}^+(x_t) = \arg\min_{q(x_t)} \sum_{j \in \mathcal{N}_{i,t} \cup i} w_j D_{\text{KL}}(q(x_t) \parallel p_{j,t-1}^+(x_t)). \tag{4.10}
\]

A notable feature of the above consensus formulation is, no local data \( y_{i,t} \) but only the local belief distribution \( p_{i,t}^+(x_t) \) is exchanged with neighbors. This feature makes the above consensus formulation suitable for implementation on dynamic and probabilistic communication networks.

**Consensus as Distributed Belief Update**

In Chapter 2 we presented a scalable belief updating framework which employs composite likelihoods. Now we show that the consensus based on KLD average in equation 4.10 can be derived as a distributed version of the scalable belief updating framework with composite likelihoods.

Consider a fully connected network with a shared prior belief distribution \( p_0(x_t) \) and the set of local sensor measurements,

\[
y_t^\top = [y_{1,t}^\top \ldots y_{N,t}^\top]^\top.
\]

Using the belief updating framework and the negative-log composite likelihood

\[
- \ln \prod p(y_{i,t} \mid x_t),
\]

the updated belief distribution is given by

\[
\arg\min_{q(x_t)} - \int q(x_t) \ln \prod_{i=1}^{N} p(y_{i,t} \mid x_t)dx_t + D_{\text{KL}}(q(x_t) \parallel p_0(x_t))
\]

\[
\propto p_0(x_t) \prod_{i=1}^{N} p(y_{i,t} \mid x_t).
\tag{4.11}
\]

Let the local filtering distribution of node \( i \) be

\[
\frac{1}{Z_i} p_0(x_t)p(y_{i,t} \mid x_t), \tag{4.12}
\]

the KLD average based consensus on the local filtering distribution is

\[
\arg\min_{q(x_t)} \sum_{i \in \mathcal{V}} D_{\text{KL}}(q(x_t) \parallel \frac{1}{Z_i} p_0(x_t)p(y_{i,t} \mid x_t))
\]

\[
\propto p_0^N(x_t) \prod_{i=1}^{N} p(y_{i,t} \mid x_t).
\tag{4.13}
\]
Figure 4.3. An example of distributed Kalman filter, using the consensus formulation. In (a), two correlated signals and their measurements are visualized. Figure (b) shows the results given by a centralized KF, which observes the measurements from two nodes simultaneously. In (c) and (d), the results given by local KFs at node 1 and node 2 are shown, respectively. The estimation of the unobserved signal is made by using only the correlation between the two signals. Finally, (e) and (f) show the results given by the distributed KF using consensus formulation. At each time step, the two nodes exchange their local KF results and formulate a consensus distribution.
Comparing with the updated belief distribution in equation 4.11, the prior in above belief distribution is overweighted by \( N \), due to the fact that the shared prior distribution is double-counted. Therefore, to recover the updated belief distribution in equation 4.11, we need to down-weight the prior distribution by \( \frac{1}{N} \):

\[
p^+(x_t) = \arg \min_{q(x_t)} \sum_{i \in \mathcal{V}} D_{KL}(q(x_t) \parallel \frac{1}{Z_t} p_0^\frac{1}{N}(x_t)p(y_i,t \mid x_t))
\]

\[
= \frac{p_0(x_t) \prod_{i=1}^N p(y_i,t \mid x_t)}{\int p_0(x_t) \prod_{i=1}^N p(y_i,t \mid x_t) dx_t}.
\]

The updated belief distribution \( p^+(x_t) \) is an approximation of the Bayesian estimation

\[
\frac{p_0(x_t)p(y_t \mid x_t)}{\int p_0(x_t)p(y_t \mid x_t) dx_t},
\]

which requires to evaluate the data likelihood \( p(y_t \mid x_t) \) in a centralized manner.

### 4.2 False Data Detector

In the remaining of this chapter, we discuss the problem of distributed state estimation when false data are injected to the local sensor measurements.

To begin with, we present the local measurement model under the false data injection attack (FDIA). Assume that node \( i \) is compromised by an attack and a false data sequence is injected into the local measurements. The local sensor measurement under FDIA is

\[
y'_{i,t} = H_{i,t}x_t + e_{i,t} + v_{i,t},
\]

where \( v_{i,t} \) is the zero-mean random false data with configurable level of variance \( \sigma_v^2 \).

As a consequence, the local filtering distribution

\[
p'_i(x_t) = \frac{p(y'_{i,t} \mid x_t)p_i^-(x_t)}{p(y'_{i,t})},
\]

is corrupted. Though the consensus formulation, the errors caused by the injected data are propagated through the network and will eventually degrade the system’s performance.
4.2.1 Centralized Detector

In this subsection we present the false data injection attack and detection in a centralized system. Consider the following centralized system

\[ x_t = Ax_{t-1} + w_t, \]
\[ y_t = Hx_t + e_t, \]

(4.18)

(4.19)

where \( x_t \in \mathbb{R}^N \) and \( y_t \in \mathbb{R}^M \). The centralized FDIA is implemented by adding a false data sequence \( v_t \) into the centralized measurement

\[ y_t' = Hx_t + e_t + v_t. \]

(4.20)

To see how the FDIA will affect the state estimation, we assume that the FDIA occurs first at time \( t \) and the previous estimates up to \( x_{t-1} \) are secure. At time \( t \), the false data \( v_t \) is injected into the measurement. The Kalman filter gives a linear estimate of the system states \( x_t \) in the following equation

\[ \hat{x}_t' = A\hat{x}_{t-1} + G_t(y_t' - HA\hat{x}_{t-1}) \]
\[ = G_t(y_t + v_t) + (A - G_tHA)\hat{x}_{t-1} \]

(4.21)

where \( G_t = (A\Sigma_{t-1}A^T + \Sigma_w)H^T[H(AS_{t-1}A^T + \Sigma_w)H^T + \Sigma_e]^{-1} \) is the Kalman gain. The error of state estimate under FDIA is

\[ \epsilon_t' = \hat{x}_t' - x_t \]
\[ = (G_tH - I)w_t + G_te_t + G_tv_t + (A - G_tHA)\epsilon_{t-1}. \]

(4.22)

Comparing to the recursive error of secure KF estimate \( \hat{x}_t \),

\[ \epsilon_t = \hat{x}_t - x_t \]
\[ = (G_tH - I)w_t + G_te_t + (A - G_tHA)\epsilon_{t-1}, \]

(4.23)

the corrupted estimate \( \hat{x}_t' \) contains an additional term of \( G_tv_t \).

However the true system state \( x_t \) is unavailable. Therefore it is not possible to use \( \epsilon_t \) for FDIA detection.

**Measurement Residual Based Detector**

Define the measurement residual of the secure Kalman filter as

\[ r_t = H\hat{x}_t - y_t, \]

(4.24)

and \( \Sigma_r \) is the covariance matrix of the measurement residual. Assume the measurement noise \( e_t \) is Gaussian, it is well-known that the quadratic form of the above residual,

\[ r_t^\top \Sigma_r^{-1} r_t, \]

(4.25)
Figure 4.4. An illustration of the decision boundary in the probability space. The local belief distributions $p_1(x_t)$, $p_2(x_t)$, and $p_4(x_t)$ are located within the decision boundary and therefore considered as secure. The distribution $p_3(x_t)$ is located outside of the boundary and therefore considered as corrupted by the false data. The distances between local belief distributions are measured by the symmetrized KL divergence.

has a $\chi^2$ distribution of $M$ degree of freedom. Many centralized FDIA detection algorithm is based on the decision rule that if

$$r_t^\top \Sigma^{-1} r_t \geq \tau,$$

then the measurement is under FDIA.

4.2.2 Distributed Detector

In distributed IoT networks, the measurement data $y_t$ is not available and therefore the centralized residual based FDIA detector is not applicable. In the rest of this chapter, we present a distributed FDIA detector which can be seamlessly integrated into the distributed state estimation method in the previous section.

Clustering of Local Belief Distributions

For node $i$ in the network, let $\mathcal{N}_{i,t}$ be its neighbour nodes at time $t$. The accessible local belief distributions for node $i$ are

$$p_i(x_t) \cup \{p_j(x_t) \mid j \in \mathcal{N}_{i,t}\}.$$  

The set of local belief distributions can be mapped to a probability space and the distances between distributions $p_i(x_t)$ and $p_j(x_t)$ can be measured by the symmetrized KL divergence:

$$D_{SKL}(p_i(x_t) \| p_j(x_t)) = \frac{1}{2}[D_{KL}(p_i(x_t) \| p_j(x_t)) + D_{KL}(p_j(x_t) \| p_i(x_t))].$$

(4.27)
Figure 4.5. An example of the distributed FDIA detection based on the distribution of symmetrized KLDs between local posteriors. Figure (a) shows three sensor measurements, where node 3 is under FDIA. In (b), the distributions of symmetrized KL divergences between posteriors of three nodes are shown. The sample mean of the symmetrized KL divergence between the secure nodes 1 and 2 is around 8, while the mean of the symmetrized KL divergence between nodes 1 and 3 is approximately 11.

The FDIA detection is therefore transformed into a task of finding a decision boundary in the probability space. Any belief distributions located outside of the decision boundary is considered as corrupted by false data. An illustration is in Figure 4.4.

Symmetrized KL Divergence and its Distribution

However, there are several challenges of clustering the local belief distributions based on the symmetrized KL divergence.

First, the symmetrized KL divergence between a pair of local belief distributions are random. Assume that $p_i(x_t)$ and $p_j(x_t)$ are two multivariate Gaussian distributions. The symmetrized KL divergence between $p_i(x_t)$ and $p_j(x_t)$ is

\[
D_{SKL}(p_i \parallel p_j) \propto (\hat{x}_{j,t} - \hat{x}_{i,t})^\top (\Sigma^{-1}_{j,t} + \Sigma^{-1}_{i,t})(\hat{x}_{j,t} - \hat{x}_{i,t}) + d(\Sigma_{j,t}, \Sigma_{i,t}),
\]

where $d(\Sigma_{j,t}, \Sigma_{i,t})$ is a deterministic function of the local filtering covariance matrices $\Sigma_{j,t}$ and $\Sigma_{i,t}$. The quadratic form of the vector $(\hat{x}_{j,t} - \hat{x}_{i,t})$, however, is a random due to the fact that $\hat{x}_{j,t}$ and $\hat{x}_{i,t}$ are linear functions of the local measurements $y_{j,t}$ and $y_{i,t}$. Therefore, clustering based on the real-time symmetrized KL divergence is not stable.

Second, considering the dynamic and probabilistic communication network, the set of local belief distributions might be small and in certain scenarios can contain more corrupted than secure distributions. With only a small set of local belief distributions, it is difficult to recognize useful cluster patterns for FDIA detection.
In light of those challenges, we propose a detection method based on clustering of local belief distributions using the averaged symmetrized KL divergence. The averaged symmetrized KL divergence can be evaluated on-the-fly by each node updating a local distance matrix of the network. Furthermore, we show that the symmetrized KL divergence between a pair of secure local belief distributions follows a generalized $\chi^2$ distribution. The averaged symmetrized KL divergence is a sample mean of the generalized $\chi^2$ distribution. Therefore, using the averaged symmetrized KL divergences, the clustering is more effective to detect FDIA.

4.3 Literature Review

False data in distributed systems, such as power grid [Liang et al. 2017], sensor networks [Mo et al. 2010], and Internet-of-Things [Bostami et al. 2019], are severe threats to the security of many functions in our modern society. Many research projects have been carried out to look for data processing techniques that are robust to the false data attacks [Liu et al. 2011; Manandhar et al. 2014; Ye and Zhang 2019].

For centralized systems, Liu et al. [2011] presented the false data injection attack (FDIA) in the state estimation task, and analyzed the system vulnerabilities in different scenarios where the attacker has various information resources to design and implement the false data attack. Using Kalman filters for the state space estimation, Mo et al. [2010] studied the FDIA problem and designed an algorithm to determine the set of estimation biases. $\chi^2$ detectors are capable of detecting false data based on Kalman filter estimates [Manandhar et al. 2014; Rawat and Bajracharya 2015]. Li et al. [2014] proposed a computationally efficient detector, which is based on the generalized likelihood ratio. More recent advance in FDIA detection involves deep neural networks which extract features in the wavelet domain as inputs [Yu et al. 2018].

The difficulty of detecting FDIA is increased for distributed systems, considering the limited information at local nodes. First, several research works are dedicated to distributed state estimation, when each node only has communication links with its neighbors [Rego et al. 2019]. Cattivelli and Sayed [2010] proposed a diffusion-based distributed Kalman filter for state estimation and smoothing. Covariance intersection (CI) is another widely applied distributed data fusion method [Julier and Uhlmann 2009]. Wang et al. [2017] proposed a CI-based method for distributed Kalman filter without exchanging the raw measurements, which mitigates the communication burden in distributed networks. Deeply related to the diffusion and the CI-based methods, consensus-based distributed Kalman filtering is adopted in this thesis.
Battistelli and Chisci [2014] presented the consensus-based method by using the average KL divergence. In this thesis, we formulate the consensus using the distributed belief updating, which gives a clear indication of down-weighting the shared prior in the network.

Based on different distributed state estimation methods, various FDIA detectors are designed and analyzed in recent studies. Guan and Ge [2017] proposed a distributed FDIA detector based on the local residual signals. Kurt et al. [2018] presented a distributed version of the online FDIA detector. Recent research trends include using machine learning methods for FDIA detection, for example, the distributed support vector machine-based detector [Esmalifalak et al. 2014] and the distributed mixture of Gaussian learning [Foroutan and Salmasi 2017]. In this thesis, we present an FDIA detector based on the distributions of KL divergences between local posteriors, which is suitable for systems with dynamic and probabilistic communication networks.

4.4 Summary

Next we present a summary of papers related to false data. The included papers are


Paper IV: Distributed GP Learning under False Data

This paper studies the problem of learning GP hyper-parameters in a distributed sensor network under false data injection attacks (FDIA). We propose a distributed learning algorithm based on formulating consensus over the network. In this work, we look into the false data problem from two perspectives:

- The local estimates are combined in a distributed manner. The corrupted estimates produced by the nodes under FDIA are down-weighted, and their impacts are gradually diminished.
- The network topology information is important for both defenders and attackers. The critical nodes are the ones that have high connectivity (measured by the betweenness centrality). When the
critical nodes are corrupted, the injected false data have a significant impact on the network.

Paper V: Secure Information Fusion

This paper considers the distributed state estimation problem in a system with mobile nodes and probabilistic communication networks. We propose an integrated solution of state estimation and false data detection. The contributions of this work are the following.

• We propose a distributed state estimation method, using consensus on local belief distributions. We show that with properly weighted prior distributions, the proposed method recovers the Bayesian belief distribution using the composite likelihood.
• We design a distributed detector which clusters the local belief distributions, based on the averaged symmetrized KL divergences. We show that the symmetrized KL divergence follows a general $\chi^2$ distribution.
• We provide an analysis of the type I and type II error rates, based on the $\chi^2$ distribution of the symmetrized KL divergence. The error rate also depends on the average number of neighbors of a node in the network.
5. Conclusion

Internet-of-Things is a fast-developing technology to make our world connected and intelligent. IoT generates massive amounts and various types of data, which has become a valuable asset to society. Therefore, data analysis plays a key role in utilizing the IoT data to achieve our society’s sustainable development.

In this thesis, we study statistical data analysis for IoT. We discuss the challenges in IoT data analysis, such as big data, missing data, and false data. To overcome these challenges, we design models and analysis methods with three important features: scalability, reliability, and robustness.

Big Data: Scalable GP
The data is a carrier of information, and the amount of information bounds the model performance (e.g., prediction accuracy). To reduce the computation complexity, we approximated the exact likelihood with the composite likelihood, which causes a loss of information. There is no free lunch. A better approximation preserves more information from the data with lower complexity. However, the task of searching for the best approximation method is often computationally expensive in itself.

In the first part of this thesis, we study GP regression with big data on resource-constrained IoT devices. We present a scalable belief updating framework using composite likelihoods. This framework can be implemented in a recursive manner, which significantly reduces GP regression’s computational complexity. Using the scalable belief updating framework, we design recursive Gaussian process learning and prediction methods. We apply the scalable GP to predict urban air quality with large historical data.

Missing Data: Reliable SSL and Robust Prediction
In safety-critical applications, reliable uncertainty measures of predictive results are essential. If a model or a method fails to provide reliable uncertainty measures, highly accurate testing results will be irrelevant. Therefore, machine learning with reliable uncertainty measure is an emerging and important research area.

Moreover, the typical "minimize average loss" approach of data modeling is only aiming for the best performance on average. However, when the data is an ensemble of subpopulations with different properties, the
best average performance model might be inapplicable. In real-world IoT applications, it is common that the system can be operated in different modes. A model trained based on minimizing average loss might fail to capture the system dynamic when operated in an unexpected mode.

The second part of this thesis discusses the reliability and robustness properties in the missing data problem. For historical data with missing labels, we propose a reliable semi-supervised learning method for solving classification problems. The proposed method provides an accurate estimate of the predictive error probability when the missing label pattern is under a relaxed assumption. We design a robust predictor for testing data with missing features, which significantly reduces the mean squared errors when rare events happen in the missing features.

**False Data: Robust and Distributed Information Fusion**

Today IoT systems have been deployed on a large scale. Moreover, cyberattacks have become significant threats to our society’s safety. An IoT system’s distributed architecture makes it difficult to eliminate false data from every local device. Thus, a robust and distributed information fusion method is critical to the IoT data security.

The third part of this thesis focuses on robust state estimation when the IoT system is under false data injection attacks. We present a distributed and robust state estimation method, based on a consensus of local information. We show that the distributed consensus fusion method can be seen as a particular case of the general belief updating framework. Furthermore, we design a distributed false data detector. The detector explores the clustering pattern of local belief distributions, using the averaged KL divergence. We apply the solution to a use case of monitoring a spatial-temporal signal with a distributed IoT system.

"You can have data without information, but you cannot have information without data.” The IoT technology enables a tremendous opportunity of digitization while imposing challenges on data analysis. We hope this thesis is a step towards scalable, reliable, and robust data analysis for IoT.
References


Acta Universitatis Upsaliensis

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 1976

Editor: The Dean of the Faculty of Science and Technology

A doctoral dissertation from the Faculty of Science and Technology, Uppsala University, is usually a summary of a number of papers. A few copies of the complete dissertation are kept at major Swedish research libraries, while the summary alone is distributed internationally through the series Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology. (Prior to January, 2005, the series was published under the title “Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology”.)