Institutionen för systemteknik
Department of Electrical Engineering

Examensarbete

Nowcasting using Microblog Data

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan vid Linköpings universitet
av

Christian Andersson Naesseth

LiTH-ISY-EX-ET--12/0398--SE

Linköping 2012

Linköpings universitet
TEKNISKA HÖGSKOLAN

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

Nowcasting using Microblog Data

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan vid Linköpings universitet
av

Christian Andersson Naesseth

LiTH-ISY-EX-ET--12/0398--SE

Handledare: **Fredrik Lindsten**
ISY, Linköpings universitet

Examinator: **Thomas Schön**
ISY, Linköpings universitet

Linköping, 18 september 2012

Avdelning, Institution
Division, Department

Automatic Control
Department of Electrical Engineering
SE-581 83 Linköping

Datum
Date

2012-09-18

Språk
Language

□ Svenska/Swedish
⊠ Engelska/English

□ ________________

Rapporttyp
Report category

□ Licentiatavhandling
⊠ Examensarbete
□ C-uppsats
□ D-uppsats
□ Övrig rapport
□ ________________

ISBN
—

ISRN
LiTH-ISY-EX-ET--12/0398--SE

Serietitel och serienummer
Title of series, numbering

ISSN
—

URL för elektronisk version

`http://www.ep.liu.se`

Titel
Title

Nowcasting med mikrobloggdata
Nowcasting using Microblog Data

Författare
Author

Christian Andersson Naesseth

Sammanfattning
Abstract

The explosion of information and user generated content made publicly available through the internet has made it possible to develop new ways of inferring interesting phenomena automatically. Some interesting examples are the spread of a contagious disease, earth quake occurrences, rainfall rates, box office results, stock market fluctuations and many many more. To this end a mathematical framework, based on theory from machine learning, has been employed to show how frequencies of relevant keywords in user generated content can estimate daily rainfall rates of different regions in Sweden using microblog data.

Microblog data are collected using a microblog crawler. Properties of the data and data collection methods are both discussed extensively. In this thesis three different model types are studied for regression, linear and nonlinear parametric models as well as a nonparametric Gaussian process model. Using cross-validation and optimization the relevant parameters of each model are estimated and the model is evaluated on independent test data. All three models show promising results for nowcasting rainfall rates.

Nyckelord
Keywords twitter, statistical learning, machine learning, gaussian process, nowcasting, social media

Abstract

The explosion of information and user generated content made publicly available through the internet has made it possible to develop new ways of inferring interesting phenomena automatically. Some interesting examples are the spread of a contagious disease, earth quake occurrences, rainfall rates, box office results, stock market fluctuations and many many more. To this end a mathematical framework, based on theory from machine learning, has been employed to show how frequencies of relevant keywords in user generated content can estimate daily rainfall rates of different regions in Sweden using microblog data.

Microblog data are collected using a microblog crawler. Properties of the data and data collection methods are both discussed extensively. In this thesis three different model types are studied for regression, linear and nonlinear parametric models as well as a nonparametric Gaussian process model. Using cross-validation and optimization the relevant parameters of each model are estimated and the model is evaluated on independent test data. All three models show promising results for nowcasting rainfall rates.

Acknowledgments

First of all I would like to thank my examinator Dr. Thomas Schön and supervisor Lic. Fredrik Lindsten for giving me this opportunity. Thanks also for all your guidance during the process of writing this thesis.

I also want to thank my family for their support during all my years of studying.

Linköping, September 2012
Christian Andersson Naesseth

Contents

Notation

Sets

Notation	Meaning
$\mathbb{R}$	Set of real numbers
$\mathcal{X}$	Set of possible inputs
K	Set of keywords
T	Set of tweets

Symbols

Symbol	Meaning
$\mathbf{x}$	Column vector
$\lvert\cdot\rvert$	Size of a set or absolute value of a scalar
$\lVert\cdot\rVert$	Euclidean distance, $\mathcal{L}^2$ norm
$\mathbf{y}^T$	Transpose of vector $\mathbf{y}$
$\hat{y}$	Estimate of y
X	Matrix
I	Identity matrix of relevant size
X^{-1}	Inverse of a matrix X
$E[\,\cdot\,]$	Expected value of a stochastic variable
$f_*\lvert X$	Conditional probability
$\mathrm{cov}(\,\cdot\,)$	Covariance
$\mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x},\mathbf{x}')\right)$	Gaussian Process with mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x},\mathbf{x}')$
$\mathcal{N}\left(\boldsymbol{\mu},\Sigma\right)$	Multivariate normal distribution with mean $\boldsymbol{\mu}$ and variance Σ

Abbreviations

Abbreviation	Meaning
ILI	Influenza-like Illness
RSS	Residual Sum of Squares
LSE	Least Squares Estimate
LASSO	Least Absolute Shrinkage and Selection Operator
RR	Ridge Regression
NN	Neural Network
GP	Gaussian Process
SE	Squared Exponential
RQ	Rational Quadratic
CV	Cross-validation
RMSE	Root Mean Square Error
UDL	User Defined Location (Twitter)
UTC	Coordinated Universal Time
CEST	Central European Summer Time
MCMC	Markov Chain Monte Carlo

1 Introduction

The boom in social media on the Internet has made it easy to collect and analyze large amounts of data generated by the public. This information is generallly unstructured and some processing needs to be done to infer any real world measurable quantities in a consistent way. This thesis explores some possibilities and opportunities in using unstructured textual information from the microblogging service *Twitter*[1] for inferring occurrences and magnitude of events and phenomena. By using theory and methods from statistical learning a measurement model usable in a Bayesian filtering context, i.e. on the form

$$y_i = h(x_i) + \varepsilon_i, \tag{1.1}$$

is inferred. Here y_i is a subset of all *tweets* in a region, x_i is the real world quantity to be estimated and ε_i is a *disturbance* or *error* term.

Nowcasting, most commonly used in finance, is a term that makes it clear that inference is performed on a *current* magnitude $\mathcal{M}(E)$ of an event E. As a case study, a measurement model for inferring regional daily precipitation in a few cities in Sweden is derived where actual tweets and rainfall levels are used.

1.1 Motivation

User generated content on the internet can contain a lot of information which can be used for data mining. Especially interesting, from a data mining perspective, are real world measures that are either difficult to estimate or whose estimates are delayed in some sence. One example where estimates are usually delayed and difficult to estimate are reports on influenza-like illnesses (ILI). Studies have shown

[1] http://www.twitter.com/

that early detection and early interventions can effectively contain an emerging epedemic, see e.g. Ferguson et al. [2005] and Longini Jr. et al. [2005]. This means detection and estimation of alarming changes in ILI rates can be of paramount importance.

This thesis will focus on a method for identifying a measurement model from a subset of tweets that can be used in a Bayesian filtering context. For more information on Bayesian filtering see for example Gustafsson [2010].

The Bayesian approach for filtering and estimation is widely used in many applications today. It offers many powerful methods for estimating and inferring values given a state-space model, or a more general probabilistic model. This thesis focuses on building the measurement model as this is the equation that pertains to the use of Twitter. It can also be extended with a dynamic model for prediction purposes. However, this is beyond the scope of this thesis.

1.2 Related Work

In recent years, inference based on unstructured textual information from social networks, search engines and microblogging have emerged as a popular research area. The work has been focused on exploiting user generated web content to make various kinds of inference. One example of inference based on information contained in tweets can be found in Bollen and Mao [2011], Bollen et al. [2010], where prediction of the stock market is performed. Another interesting example is detection of earth quakes in Japan, see Sakaki et al. [2010], viewing each seperate person as a type of sensor and using classification theory as a detection algorithm. Lansdall-Welfare et al. [2012] use data mining methods to analyze correlations between the recession and public mood in the UK.

A big part of the research is concentrated on inferring ILI rates based on content from the social web or search engines. An early example using search engine data is Ginsberg et al. [2009]. As was mentioned in Section 1.1, estimating ILI rates is difficult with conventional means. Since it is a very important measure to keep track of it would be interesting to use other means to achieve a better up to date estimate. A few examples using social network information are Achrekar et al. [2012], Achrekar et al. [2011], Chen et al. [2010], Chew and Eysenbach [2010] and Lampos and Cristianini [2011]. The last one, *Nowcasting Events from the Social Web with Statistical Learning* by Lampos and Cristianini [2011], requires special mention as it not only uses Twitter to predict ILI rates, but also infers daily rainfall rate which is the case study of this thesis. The model considered in Lampos and Cristianini [2011] is a linear parametric regression model. In this thesis both similar models and nonlinear and nonparametric alternatives will be considered. Differences between the linear models in this thesis and Lampos and Cristianini [2011] lie in model selection, validation and data collection methods.

1.3 Microblogging

Microblogging is a broadcast medium very similar to regular blogging. In comparison to traditional blogging, microblogging is usually smaller in both aggregate and actual file or message size. Microblogs let their users exchange small *microposts*, elements of content, containing short sentences, images or links to other content. The range of topics discussed in the microposts can vary widely. It might range from simple statements of how a person feels, to complex discussions on politics or current news.

Twitter is one of the biggest microblogging services to date with more than 500 million users[2]. Twitter is the microblogging service used for the case study in this thesis, microposts are therefore hereafter referred to as *tweets*. Tweets sent by one person to another that are forwarded or just tweeted again by a third person are called *retweets*.

Other examples of microblogging services are Tumblr [3], Plurk [4] and Sina Weibo [5].

1.4 Problem Formulation

It is assumed that daily rainfall (or ILI) rate can be inferred from frequencies of relevant keywords contained in tweets, with retweets filtered out, in a certain region on that day. The problem is then to identify the mathematical model describing the relationship between the keyword input frequencies and the output rainfall rate. It is further assumed that the model (function) is invariant to the region.

1.5 Thesis Outline

The first four chapters of this thesis are dedicated to an introduction of the topic and background theory for the case study. Results and discussions are presented in chapters five and six. The outline is summarized below,

Chapter 2 provides theory on parametric regression, both linear and nonlinear. The purpose is to give a short overview of the theory used in this thesis.

Chapter 3 focuses on nonparametric regression. Specifically the theory for machine learning with Gaussian Processes is reviewed.

Chapter 4 presents an overview of methods for model selection, validation and assessment.

[2] http://www.mediabistro.com/alltwitter/500-million-registered-users_b18842, (August 4th, 2012)

[3] https://www.tumblr.com/

[4] http://www.plurk.com/

[5] http://www.weibo.com/

Chapter 5 describes how data was collected and of the results obtained when applying theory to the case study.

Chapter 6 compares the different results, discusses the data used and improvements that can be made. It also contains a section on future work.

Chapter 2 and 3 both contain modelling sections, which describe the actual mathematical models used for inference in this thesis.

2

Parametric Regression

This chapter concerns regression theory based on a parametric approach. Parametric regression assumes a model where prediction can be made based on a finite set of learned parameters. These parameters are estimated in a training phase based on a training data set of inputs and outputs, $\mathcal{D} = \{(y_i, \mathbf{x_i})|i = 1, \ldots, N\}$. This means prediction can be made based on new inputs and the parameters without the need to save any of the data used in the training phase. The non-parametric approach, explained more in Chapter 3, on the other hand uses the training data as well as inferred parameters to predict values based on new inputs.

The first type of models discussed will be models linear in the parameters, which are a very important class of models in statistical learning. Let y_i be a random variable that is observed. The linear model can then be expressed in the form:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_{p-1} x_{i,p-1} + \varepsilon_i \ , \quad i = 1, \ldots, N, \tag{2.1}$$

where $x_{i,j}$ are the known *input variables*, ε_i is the *error* and β_j are the *parameters* to be estimated. An observation to be made is that the model need not be linear in the input variables, these can come from different sources:

- quantitative measurements
- general transformations of inputs, such as log, square-root, etc.
- basis expansions or interactions between variables, for example, $x_{i,2} = x_{i,1}^2$, $x_{i,3} = x_{i,1} x_{i,2}$
- many other forms linear in the parameters β

A simple example is a resistor with a controllable current and measurable voltage, see Example 2.1, where the resistance and measurement noise is to be estimated.

2.1 Example

Ohm's law states that the current through a conductor is proportional to the potential difference across the conductor. In mathematical terms:

$$U = RI \tag{2.2}$$

This can be expressed as a linear model on the form

$$y_i = \beta_0 + \beta_1 x_{i,1} + \varepsilon_i, \tag{2.3}$$

where $y_i = U$, $x_{i,1} = I$, ε_i is zero-mean measurement noise and β_0, β_1 are the parameters to be estimated. β_0 in this case is usually called an intercept and can estimate a constant term present in the system, for example the mean of the measurement noise. The main goal is to estimate β_1 which corresponds to the resistance, R in Ohm's law, of the resistor.

Ways of estimating the parameters in these kinds of models are discussed in Section 2.1. The second part of this chapter, Section 2.2, concerns a special case of the general nonlinear parametric model

$$y_i = f(x_{i,1}, \ldots, x_{i,p-1}, \beta_0, \ldots, \beta_{p-1}) + \varepsilon_i. \tag{2.4}$$

The special case of the above general nonlinear parametric model considered is commonly referred to as a neural network. The last part, Section 2.3, discusses the actual models used for inference in this thesis.

2.1 Least Squares

One of the most common ways to estimate the parameters in (2.1) is by minimizing the residual sum of squares (RSS) with respect to all the parameters. With batch form notation, i.e.

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad X = \begin{bmatrix} x_{1,0} & \cdots & x_{1,p-1} \\ \vdots & \ddots & \vdots \\ x_{N,0} & \cdots & x_{N,p-1} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{bmatrix},$$

where $x_{i,0} = 1$, the estimate of $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \, \mathrm{RSS}(\boldsymbol{\beta}). \tag{2.5}$$

RSS ($\boldsymbol{\beta}$) is given by

$$\mathrm{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{N} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 = (\mathbf{y} - X\boldsymbol{\beta})^T (\mathbf{y} - X\boldsymbol{\beta}). \tag{2.6}$$

Differentiating this with respect to β gives

$$\frac{\partial \mathrm{RSS}(\beta)}{\partial \beta} = -2X^T(\mathbf{y} - X\boldsymbol{\beta}), \tag{2.7a}$$

$$\frac{\partial^2 \mathrm{RSS}(\beta)}{\partial \beta^2} = 2X^T X. \tag{2.7b}$$

Provided that X has full rank the unique solution is obtained by setting (2.7a) equal to zero, and solving for β, which gives

$$\widehat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T \mathbf{y}. \tag{2.8}$$

This is the *Least Squares Estimate* (LSE) of $\boldsymbol{\beta}$. In the case that X does not have full rank one can still obtain a solution using a generalized inverse, see Seber and Lee [2003, p. 38].

2.1.1 Properties of the Least Squares Estimate

Assuming that the errors are zero-mean, i.e. that $\mathrm{E}[\varepsilon_i] = 0$, and that X has full rank the following relation holds

$$\mathrm{E}[\widehat{\boldsymbol{\beta}}] = \left(X^T X\right)^{-1} X^T \mathrm{E}[\mathbf{y}] = \left(X^T X\right)^{-1} X^T X \boldsymbol{\beta} = \boldsymbol{\beta}. \tag{2.9}$$

Hence, the LSE is an *unbiased* estimate of the parameters. Also assuming that the errors are uncorrelated, $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0$, and have the same variance , $\mathrm{Var}(\varepsilon_i) = \sigma^2$, the variance of the estimate is given by

$$\mathrm{Var}(\widehat{\boldsymbol{\beta}}) = \sigma^2 \left(X^T X\right)^{-1}. \tag{2.10}$$

An unbiased estimate of the variance is given, see [Hastie et al., 2009, p. 47], by

$$\hat{\sigma}^2 = \frac{1}{N - p - 1} (\mathbf{y} - \mathbf{X}\widehat{\beta})^T (\mathbf{y} - \mathbf{X}\widehat{\beta}). \tag{2.11}$$

N is the total number of data points and p is related to the model order, see (2.1). It can also be shown that the LSE has minimum variance among all linear and unbiased estimates of β. This is called the *Gauss-Markov Theorem*, see Hastie et al. [2009, p. 51].

2.1.2 Shrinkage Methods

Shrinkage methods, defined by (2.12), are in machine learning literature commonly referred to as regularization methods. Roughly speaking, they are a way of controlling overfitting to get a model that generalizes better. One way of describing all shrinkage methods very elegantly in one equation is

$$\tilde{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\} \tag{2.12}$$

with $q \geq 0$ and λ being a parameter that controls the amount of shrinkage. The cases $q = 1$ and $q = 2$ give the well known *Least Absolute Shrinkage and Selection Operator* (LASSO) and *Ridge Regression* (RR), respectively. Given $q \geq 1$, a convex optimization problem is obtained. In the ridge regression case, following the reasoning in Section 2.1, the closed form solution can be found to be

$$\widehat{\boldsymbol{\beta}}^{\mathrm{RR}} = \left(X^T X + \lambda \mathbf{I} \right)^{-1} X^T \mathbf{y}. \tag{2.13}$$

2.2 Neural Networks

In this section the following model is studied

$$a_j(\mathbf{x}_i) = \sum_{k=1}^{p-1} w_{jk}^{(1)} x_{i,k} + w_{j0}^{(1)}, \tag{2.14a}$$

$$z_j(\mathbf{x}_i) = h\left(a_j(\mathbf{x}_i) \right), \tag{2.14b}$$

$$a(\mathbf{x}_i) = \sum_{j=1}^{M} w_j^{(2)} z_j(\mathbf{x}_i) + w_0^{(2)}, \tag{2.14c}$$

$$y_i(\mathbf{x}_i, \mathbf{w}) = \sigma(a) + \varepsilon_i. \tag{2.14d}$$

This is the general expression for a neural network (NN) model with one target variable. The a_j are referred to as *activations*, these are then nonlinearly transformed via the *activation function* $h(\cdot)$ into the *hidden units* z_j. The variable a is known as *output activation*. This is transformed by yet another activation function $\sigma(\cdot)$ which gives the final output y_i. For regression problems $\sigma(\cdot)$ is usually set to the identity function, which gives $y_i = a$. This model formulation will be used for the case study in this thesis. Hence, the output is a nonlinear function in the parameters $w_{jk}^{(1)}$ and the inputs. The parameters $w_{jk}^{(1)}$ and $w_j^{(2)}$ are often referred to as *weights*. The NN can easily be represented in a network diagram as can be seen in Figure 5.9, where parameters are applied along the arrows. Additional input features $x_0 = z_0 = 1$ are added to capture the bias parameters $w_{j0}^{(1)}$ and $w_0^{(2)}$.

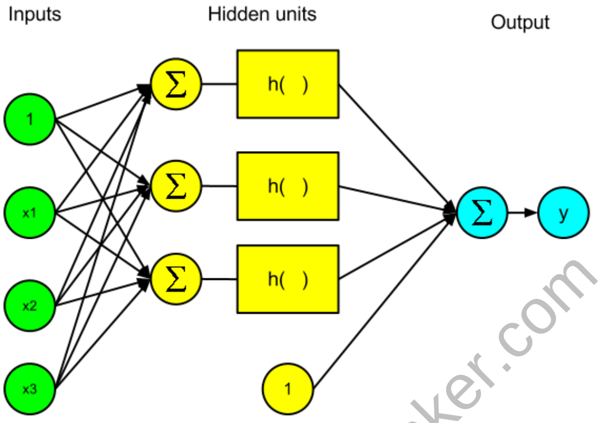


Figure 2.1: *Neural network diagram.*

This gives a compact notation for y_i on the form

$$y_i(\mathbf{x}_i, \mathbf{w}) = \sum_{j=0}^{M} w_j^{(2)} h\left(\sum_{k=0}^{p-1} w_{jk}^{(1)} x_{i,k}\right) + \varepsilon_i. \tag{2.15}$$

The activation function $h(\cdot)$ is usually set to either the logistic sigmoid or the hyperbolic tangent function [Bishop, 2006, pg. 227]. In this thesis the logistic sigmoid function, $h(x) = (1 + e^{-x})^{-1}$, will be used.

2.2.1 Learning a Neural Network

To train the network and estimate the parameters in a regression problem, the RSS is used. With $\mathbf{w}$ denoting the complete set of weights

$$\mathbf{w} = (w_{00}^{(1)}, \dots, w_{Mp-1}^{(1)}, w_0^{(2)}, \dots, w_M^{(2)})^T$$

with a total of $(M + 1)(p + 1)$ parameters, the error function becomes

$$\mathrm{RSS}(\mathbf{w}) = \sum_{i=1}^{N}\left(y_i - \sum_{j=0}^{M} w_j^{(2)} h\left(\sum_{k=0}^{p-1} w_{jk}^{(1)} x_{i,k}\right)\right)^2 \tag{2.16}$$

Finding the **w** that minizes (2.16), or the loss function described in Section 2.2.2, is a nonconvex optimization problem which means that it is often impossible to find an analytical solution. Usually it is not necessary, nor possible, to find the global optima to (2.18) and iterative numerical procedures must be used. A general equation for an iterative numerical optimization solver is

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \Delta\mathbf{w}^{(\tau)}, \tag{2.17}$$

which is initialized with $\mathbf{w}^{(0)}$ and the update of the weights at each iteration step, τ, is denoted $\Delta\mathbf{w}^{(\tau)}$. Different algorithms have different ways of selecting this update. Iteration is performed until convergence or until a satisfactory value has been found [Bishop, 2006].

2.2.2 Regularization and Neural Networks

Typically the global minima of (2.16) is not the best solution as this will often result in an overfit to the training data points [Hastie et al., 2009, pg. 398]. It is also very difficult to know whether the global optima has been reached. To alleviate the problem of overfit to training data a regularization method called *weight decay* is performed. The new problem formulation follows the same principles and is on the same form as (2.12). The loss function, denoted by L, to be minimized thus becomes

$$L(\mathbf{w},\lambda) = \sum_{i=1}^{N}\left(y_i - \sum_{j=0}^{M} w_j^{(2)} h\left(\sum_{k=0}^{p-1} w_{jk}^{(1)} x_{i,k}\right)\right)^2 + \lambda \sum_{j=0}^{M}\left(|w_j|^q + \sum_{k=0}^{p-1} |w_{jk}|^q\right) \tag{2.18}$$

Where λ and $q \geq 0$ are tuning parameters that controls the amount of shrinkage imposed on the parameters **w**.

2.3 Modelling

For the purpose of model estimation $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,p-1})^T$ will be a vector with the frequencies of keywords for a time instance i. The set of candidate keywords are denoted $K = \{k_l\}, l \in \{1, \ldots |K|\}$, where $|K|$ is the size of set K. The retrieved tweets for time instance i in region r are denoted $T_i^{(r)} = \left\{t_j\right\}, j \in \left\{1, \ldots, |T_i^{(r)}|\right\}$. The indicator function $\mathbf{1}_{t_j}(k_l)$ indicates whether a keyword k_l is contained in a tweet t_j or not, i.e.

$$\mathbf{1}_{t_j}(k_l) = \begin{cases} 1 & \text{if } k_l \in t_j \\ 0 & \text{otherwise.} \end{cases} \tag{2.19}$$

This gives the frequencies ω of the keywords in a region r:

$$\omega(k_l, T_i^{(r)}) = \frac{1}{|T_i^{(r)}|} \sum_{j=1}^{|T_i^{(r)}|} \mathbf{1}_{t_j}(k_l)\,. \tag{2.20}$$

Inputs $\mathbf{x}_i^{(r)}$, part of the set $\mathcal{X} = \mathbb{R}^{|K|}$, are given by:

$$\mathbf{x}_i^{(r)} = \left(\omega(k_1, T_i^{(r)}), \ldots, \omega(k_{|K|}, T_i^{(r)})\right)^T \tag{2.21}$$

Target (output) variables, i.e. the daily rainfall rates, are denoted by $y_i^{(r)}$. The model (function f) estimated in this thesis, for region r with noise $\varepsilon_i^{(r)}$, is then formulated as:

$$y_i^{(r)} = f^{(r)}(\mathbf{x}_i^{(r)}) + \varepsilon_i^{(r)}. \tag{2.22}$$

The assumption, regarding a region invariant property of the function, mentioned in Section 1.4 means that the model estimated is the same for each region. Because of this the superscript (r) will be supressed in this thesis. This gives the general model

$$y_i = f(\mathbf{x}_i) + \varepsilon_i. \tag{2.23}$$

2.3.1 Linear Model

The specific linear model employed in this thesis is on the form

$$\begin{aligned} y_i =& \left(\beta_0, \ldots, \beta_{\frac{|K|(|K|+3)}{2}}\right)\Big(1, \omega(k_1, T_i), \ldots, \omega(k_{|K|}, T_i), \omega^2(k_1, T_i), \ldots, \omega^2(k_{|K|}, T_i), \\ & \omega(k_1, T_i)\omega(k_2, T_i), \omega(k_1, T_i)\omega(k_3, T_i), \ldots, \omega(k_{|K|}, T_i)\omega(k_{|K|-1}, T_i)\Big)^T + \varepsilon_i \\ =& \boldsymbol{\beta}\mathbf{x}_i + \varepsilon_i. \end{aligned} \tag{2.24}$$

This means that not only the direct keyword frequencies but also the second degree terms are considered as inputs to get a more general model.

2.3.2 Nonlinear Model

The nonlinear model, with inputs

$$\mathbf{x}_i = \left(1, \omega(k_1, T_i), \ldots, \omega(k_{|K|}, T_i)\right)^T , \tag{2.25}$$

considered is exactly the one given in (2.14). It is repeated here in compact form with additive Gaussian noise:

$$y_i = \sum_{j=0}^{M} w_j^{(2)} h\left(\sum_{k=0}^{|K|} w_{jk}^{(1)} x_{i,k}\right) + \varepsilon_i \ , \ \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \tag{2.26}$$

3

Nonparametric Regression using Gaussian Processes

This chapter explains nonparametric regression theory using Gaussian processes. In Chapter 2 a parametric approach to statistical learning was employed. The nonparametric approach assumes that the function structure is unknown and should also be learned from information contained in the training data set, $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, N\}$. Whencombining the nonparametric approach and the theory of Gaussian Processes, the function is modelled as a stochastic *process*. Roughly speaking this can be seen as an extension of probability distributions to *functions*. This part of the thesis will consider inference directly in a function space.

Section 3.1 first explains what a Gaussian Process (GP) is. Section 3.2 moves on to explain how inference in a function space works. Decision theory, in Section 3.3, briefly explains how point estimation is performed based on the model of the function. Section 3.4 gives some examples of commonly used covariance functions and Section 3.5 gives an example of the calculations involved in GP inference. The last section, Section 3.6, concludes this chapter with a few comments regarding the actual model used for the case study.

3.1 Gaussian Processes

To describe distributions over functions the Gaussian process is introduced:

3.1 Definition. A Gaussian process *is a collection of random variables, for which any linear functional applied to it is normally distributed.*

Following the notation and reasoning by Rasmussen and Williams [2006], a GP can be described completely by its mean function, $m(\mathbf{x})$, and its covariance function, $k(\mathbf{x}, \mathbf{x}')$. These are, for a real process $f(\mathbf{x})$, defined as

$$m(\mathbf{x}) = E[f(\mathbf{x})], \tag{3.1a}$$

$$k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]. \tag{3.1b}$$

This means that the Gaussian process can be written in the following way

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right). \tag{3.2}$$

Often, especially in the areas of control and communication theory, Gaussian processes are defined over time. In this thesis the index set will be the set of possible inputs, $\mathcal{X}$, as defined in Section 2.3. Figure 3.1 shows an example of three functions drawn at random from a zero-mean GP prior.

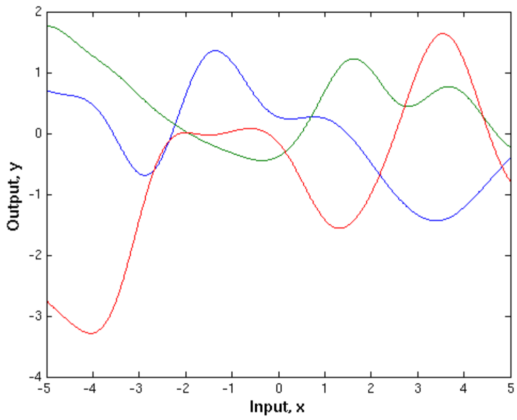


Figure 3.1: *Samples from a zero-mean Gaussian Process prior.*

3.2 Inference

To make inference in function space one first needs to make assumptions on which types of functions to consider. Here the following form is assumed:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad \text{where} \quad f(\mathbf{x}) \sim \mathcal{GP}\left(0, k(\mathbf{x}, \mathbf{x}')\right), \tag{3.3}$$

where $f(\mathbf{x})$ is a zero-mean GP and ε is zero-mean measurement noise with covariance σ^2. This problem formulation may look very similar to the ones employed in Chapter 2. Now, however, the function f is modelled as a GP and the structure and shape is learnt from training data. Modelling the function space as a zero-mean process is not as much of a restriction as might be expected. This because the posterior distribution does not necessarily have to be zero-mean.

For notational convenience X and $\mathbf{y}$ collects all training data, input and output, in a matrix and vector respectively. $K(X,X)$ is the $N \times N$ covariance matrix defined by applying the covariance function $k(\mathbf{x},\mathbf{x}')$ element wise to the inputs in X. A star, $*$, denotes inputs and predictions for independent test data. $K(X,X_*)$ analogously to $K(X,X)$ is, if there are M test data points, an $N \times M$ covariance matrix.

Assuming additive indepedent identically distributed Gaussian noise, the prior on the observations $\mathbf{y}$ becomes

$$E[\mathbf{y}] = \mathbf{0}, \tag{3.4a}$$

$$\operatorname{cov}(\mathbf{y}) = K(X,X) + \sigma^2 I. \tag{3.4b}$$

The prediction for novel inputs, $\mathbf{x}_*$, with notation $\mathbf{f}_* = \mathbf{f}(\mathbf{x}_*)$, is given as in Rasmussen and Williams [2006] by

$$\mathbf{f}_*|X,\mathbf{y},X_* \sim \mathcal{N}\left(\bar{\mathbf{f}}_*, \operatorname{cov}(\bar{\mathbf{f}}_*)\right), \quad \text{where} \tag{3.5a}$$

$$\bar{\mathbf{f}}_* = K(X_*,X)\left(K(X,X) + \sigma^2 I\right)^{-1}\mathbf{y}, \tag{3.5b}$$

$$\operatorname{cov}(\bar{\mathbf{f}}_*) = K(X_*,X_*) - K(X_*,X)\left(K(X,X) + \sigma^2 I\right)^{-1} K(X,X_*). \tag{3.5c}$$

3.3 Decision Theory

Decision theory concerns making point estimation based on distributions. Up until now, only distributions over functions have been considered. In practical applications, however, sooner or later it is necessary to make a decision based on this distribution. This usually means a point-like prediction of y_*, with knowledge of $\mathbf{x}_*$, is needed which is optimal in some sense. A loss function, $L(y_{\text{true}}, y_{\text{pred}})$, is employed that specifies the penalty incurred when predicting y_{pred} when the actual value is y_{true}. The most common loss functions are $|y_{\text{true}} - y_{\text{pred}}|$ and $(y_{\text{true}} - y_{\text{pred}})^2$, i.e. absolute deviation and squared loss. The true output, y_{true}, is generally not known and so the *expected loss* or *risk*

$$R_L(y_{\text{pred}}|\mathbf{x}_*) = \int L(y_*, y_{\text{pred}})p(y_*|\mathbf{x}_*, X, \mathbf{y})dy_*, \tag{3.6}$$

with respect to y_{pred} is usually minimized. This is then the optimal point estimate of y

$$y_{\text{opt}}|\mathbf{x}_* = \underset{y_{\text{pred}}}{\operatorname{argmin}} R_L(y_{\text{pred}}|\mathbf{x}_*) \tag{3.7}$$

The optimal point estimate for the model structure assumed in Section 3.2 with squared error loss function is the expected value of the conditional predictive distribution, i.e. (3.5b).

3.4 Covariance Functions

An important part in the learning of Gaussian processes is designing the covariance function, $k(\mathbf{x}, \mathbf{x}')$. The important part is not only picking an appropriate function to capture the relevant structure of the underlying function to be estimated, but also estimating the relevant hyperparameters, i.e. the parameters of the function. The first part is briefly discussed here, for a more complete treatment see Rasmussen and Williams [2006], and the second part is explained in Section 4.3.

For $k(\mathbf{x}, \mathbf{x}')$ to be a valid covariance function it must be positive semidefinite. In this thesis only *stationary* and *isotropic* covariance funtions will be considered. A stationary covariance function is a function of only $\mathbf{x} - \mathbf{x}'$. The isotropic attribute further restricts the covariance function to be a function of $r = \|\mathbf{x} - \mathbf{x}'\|$. Another common name used instead of covariance function is a *kernel*. This term is more general than a covariance function, but under a few conditions it can be seen as equivalent to the covariance function, [Rasmussen and Williams, 2006, pg. 80]. Hence, the two expressions kernel and covariance function will be used interchangeably in the rest of the thesis.

3.4.1 Common Kernels

In this section a few examples of commonly used kernels are mentioned. The ones considered in this section are all isotropic and have one or more hyperparameters. Estimation of these hyperparameters will be discussed in Section 4.3.

Constant Kernel

The constant covariance function consists of a positive constant

$$k_{\text{Const.}} = m^2, \tag{3.8}$$

where m is referred to as magnitude.

Squared Exponential Kernel

The *squared exponential* (SE) covariance function has the form

$$k_{\text{SE}}(r) = \exp\left(-\frac{r^2}{2l^2}\right), \tag{3.9}$$

where the hyperparameter is the *characteristic length-scale* l. This covariance function is infinitely differentiable resulting in a very smooth regressor, i.e. estimated function.

Rational Quadratic Kernel

The *rational quadratic* (RQ) is given by

$$k_{\text{RQ}}(r) = \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha}. \tag{3.10}$$

The hyperparameters of this kernel are α, $l > 0$. This covariance function can also be seen as an infinite sum (*scale mixture*) of SE kernels with different characteristic length-scales. If $\alpha \to \inf$ the RQ kernel approaches the SE covariance function with characteristic length-scale l [Rasmussen and Williams, 2006, sec. 4.2.1].

3.4.2 Combining Kernels

To make new kernels from old kernels there are a few attributes that are useful. Here they are stated as a theorem:

3.2 Theorem. *Composite Kernels*

1. *The sum of two kernels is a kernel.*
2. *The product of two kernels is a kernel.*

Proof: Let $f_1(\mathbf{x})$, $f_2(\mathbf{x})$ be two independent zero-mean stochastic processes

1. Consider $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$. Then $\text{cov}(f(\mathbf{x}), f(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}') = E[f_1(\mathbf{x})f_1(\mathbf{x}')]$ $+E[f_2(\mathbf{x})f_2(\mathbf{x}')] - E[f_1(\mathbf{x})]E[f_2(\mathbf{x}')] - E[f_1(\mathbf{x}')]E[f_2(\mathbf{x})] = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$.
2. Consider $f(\mathbf{x}) = f_1(\mathbf{x})f_2(\mathbf{x})$. Then $\text{cov}(f(\mathbf{x}), f(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}') =$ $E[f_1(\mathbf{x})f_2(\mathbf{x})f_1(\mathbf{x}')f_2(\mathbf{x}')] = E[f_1(\mathbf{x})f_1(\mathbf{x}')]E[f_2(\mathbf{x})f_2(\mathbf{x}')] = k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$.

□

This means that several covariance functions can be combined for a better regression result, i.e. to capture potentially more complex structures from the data. However, with additional complexity, the risk of overfit to the training data is larger. This will be discussed further in Chapter 4.

3.5 GP Inference Example

As all the basic theory to perform inference in function space has now been explained, an example will illustrate these principles for greater clarity.

3.3 Example

Assuming the true function, force of drag of an object through a fluid, is given by (3.11)

$$f(x) = \frac{1}{2}C_d\rho Ax^2, \tag{3.11}$$

where C_d is drag coefficient, ρ is the density of the fluid, A is reference area and x is speed of object relative to fluid. For simplicity all constants, i.e. C_d, ρ and A, are set to 1. Generating some data from this model with added zero-mean Gaussian noise, with variance $\sigma^2 = 0.01$, results in the plot seen in Figure 3.2. The inputs, x, are 20 points evenly spaced on the interval $[0, 1]$.

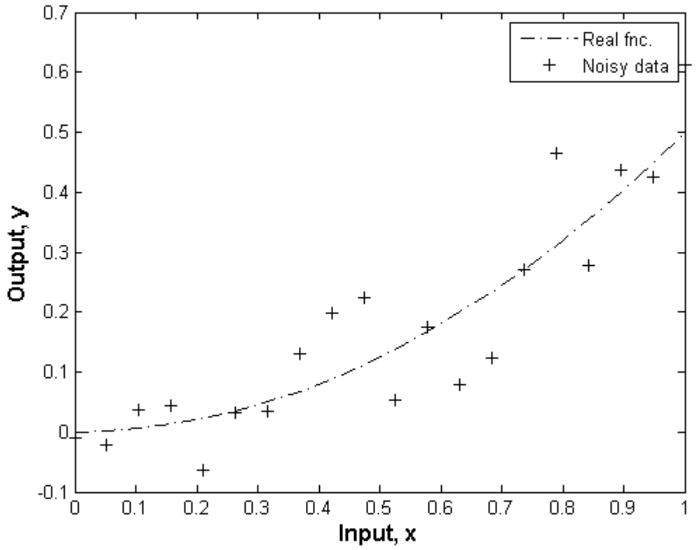


Figure 3.2: *Noisy data,* $y = f(x) + \varepsilon$.

Assuming $f \sim \mathcal{GP}(0, k(x, x'))$ where the kernel is a combination of the SE and constant kernel types. Characteristic length-scale is $\frac{1}{4}$ and the magnitude is 1. Observe that these two hyperparameters and noise variance are generally not known and must be learned from training data as well. However, learning of hyperparameters will be covered in Chapter 4. Performing inference and evaluating the model on independent data, $x_* = (0\ 0.11\ 0.22\ \dots\ 1)^T$, gives the plot in Figure 3.3.

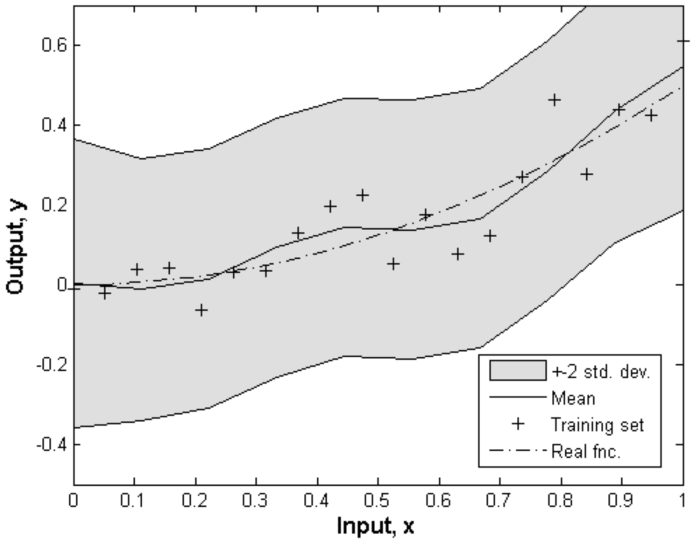


Figure 3.3: *GP inference with SE kernel.*

The training data is denoted by + and the real function by a dash-dotted line. Inferred prediction mean is denoted by a continuous line and its ±2 standard deviation (corresponding to a 95% confidence interval) by the grey area. Another illustrating example can be found in [Rasmussen and Williams, 2006, pg. 15] where the impact of data on the posterior covariance is clearly displayed.

3.6 Modelling

The inputs, $\mathbf{x}_i$, are formed as in (2.25) in Section 2.3. The output, y_i, is the daily rainfall rate (in mm). In mathematical terms

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \text{ where } f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')) \text{ and } \varepsilon_i \sim \mathcal{N}(0, \sigma^2). \tag{3.12}$$

This means the function is modelled as a GP and the errors, ε, as a Gaussian zero-mean random variable. The covariance function, $k(\mathbf{x}, \mathbf{x}')$, is modelled by a combination of kernels. The ones considered in this thesis are:

$$k(\mathbf{x}, \mathbf{x}') = m^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right), \tag{3.13a}$$

$$k(\mathbf{x}, \mathbf{x}') = m^2 \left(1 + \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\alpha l^2}\right)^{-\alpha}. \tag{3.13b}$$

Measurement noise is modelled as independent, normal distributed, white noise with covariance σ^2. Model (3.13a) and (3.13b) has in total, including the noise covariance, 3 and 4 hyperparameters respectively. These are examples of SE and RQ kernels combined with the constant kernel.

4

Model Validation, Selection and Assessment

This chapter concerns theory regarding model *validation*, *selection* and *assessment*. In the best case scenario, with ample amount of data available, the data is usually split into three independent parts, one for training, one for validation and one to test the model, see Figure 4.1. The training and validation sets contain N data points in total and the test set M data points. Model validation can be seen as the process of assigning a measure of fit, from the trained model, on the validation data set. Model selection is the process of selecting the, in some sence, best model using training and validtion data. Model assessment then analyzes how this model performs on independent test data. However, data is usually scarce and therefore other methods need to be used. One very common method, that will be used in this thesis, is called *cross-validation* and is described in Section 4.1. The problem of model assessment is then explained in Section 4.2 and comments and extension to the Bayesian case and Gaussian Processes is then discussed in Section 4.3.

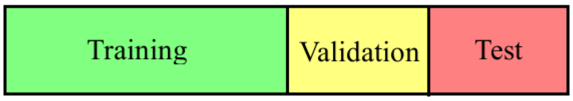


Figure 4.1: *Data split illustration for model selection, validation and assessment.*

4.1 Cross-validation

K-fold cross-validation is the process of taking all data in the training and validation set and split it into K roughly equal parts, see Figure 4.2 for K = 10.

1
2
3
4
5
6
7
8
9
10

Figure 4.2: *Data split illustration for 10-fold cross-validation.*

Training is performed on all the parts except the k-th, for $k = 1, \ldots$,K. The loss function, or the prediction error, is then evaluated on the k-th data set and we average over the data. Mathematically, denoting the estimated function with tuning parameter α by $\hat{f}^{-k}(x, \alpha)$ and the loss function by L, the cross-validation estimate of the prediction error becomes

$$\mathrm{CV}(\hat{f}, \alpha) = \frac{1}{N} \sum_{i=1}^{N} L\left(y_i, \hat{f}^{-\kappa(i)}(x_i, \alpha)\right), \tag{4.1}$$

where $\kappa : \{1, \ldots, N\} \mapsto \{1, ..., K\}$ is an indexing function that indicates which of the K parts an observation y_i belongs to. For a more detailed explanation see [Hastie et al., 2009, pg. 242]. The tuning parameter $\hat{\alpha}$ that minimizes $\mathrm{CV}(\hat{f}, \alpha)$ is selected, giving the final model $f(x, \hat{\alpha})$. All data in the training and validation set is then used to fit this model.

4.2 Model Assessment

Model assessment is performed on independent test data. It is interesting to find out how well the estimated model, found by cross-validation, generalizes. This is done by calculating, or estimating, the *root mean square error* (RMSE). An estimate is given by:

$$\mathrm{RMSE} = \sqrt{\frac{1}{M} \sum_{i=N+1}^{N+M} \left(y_i - \hat{f}(x_i)\right)^2}\ , \tag{4.2}$$

i.e. averaging the squared-error loss function over the test data set and taking the square root of the result.

4.3 Gaussian Processes

Gaussian Processes gives a predictive distribution for novel inputs, see (3.5). To get a point estimation usable for validation and assessment of the model, the theory discussed in Section 3.3 is employed.

The hyperparameters in the GP model are measurement noise covariance σ^2 and the parameters that define the covariance function of the GP, $k(\mathbf{x}, \mathbf{x}')$. These are estimated by performing CV and selecting the parameters that give the minimum value of the CV prediction error estimate (4.1). For assessment, all training data, optimal hyperparameters estimated by CV and the novel inputs are used to form a point prediction

$$\hat{f}(x_i, \text{hyp}) = K(x_i, X)\left(K(X, X) + \sigma^2 I\right)^{-1} \mathbf{y}. \tag{4.3}$$

Based on this point prediction, the RMSE is estimated in the same way as for the parametric regression methods.

5

Experiments and Results

This chapter presents the results, when nowcasting rainfall rates, obtained using theory from Chapters 2 through 4 and data collected from Twitter. Section 5.1 first explains how the data was collected, stored and retrieved for regression purposes. Section 5.2 then continues by describing the specific results obtained for each type of regression.

5.1 Data Collection

Millions of tweets have been collected using Twitter's Search API for the purposes of the case study in this thesis. The ground truth data has been obtained from the National Climate Data Center's (NCDC) online database[1]. The ground truth data for rainfall rate, originally given in inches, was converted to millimeters (mm). Tweets where collected by exploiting *JSON* feeds returned by executing a query[2] to the Twitter Search API. The returned information was parsed using a PHP script and subsequently stored and indexed in a MySQL database.

Cities used in this case study where Göteborg, Malmö, Linköping, Norrköping and Helsingborg. Data was collected for the time interval June 6, 2012 to July 28, 2012. This means the total amount of data points used for CV and assessment was $N + M = (\#\text{regions}) \cdot (\#\text{days}) = 5 \cdot 53 = 265$.

[1] http://www7.ncdc.noaa.gov/CDO/cdoselect.cmd?datasetabbv=GSOD&countryabbv=&georegionabbv= , (August 2, 2012)

[2] For a region defined by a point with latitude X, longitude Y, radius R km and the 300 most recent tweets defined by setting tweets per page (rpp) to 100 and performing the following query 3 times with page set to 1, 2 and 3 (PAGENR): http://search.twitter.com/search.json?rpp=100&geocode=X,Y,Rkm&page=PAGENR

5.1.1 Tweet Storage

The tweets collected where stored in a MySQL database. For each tweet, the features shown in Table 5.1 was stored. If a tweet was not geotagged, i.e. no information on latitude and longitude was available, the user defined location (UDL) was saved instead.

MySQL column	Explanation
id	The unique ID of the tweet
created_at	Time of creation in the form YYYY-MM-DD HH:MM:SS
from_user	Username of the tweeting person
from_user_id	ID of the tweeting person
text	Tweet content, maximum 140 characters
location	User defined location
geo	Longitude and latitude where the tweet was made
to_user_id	ID of the recipient user if applicable
iso_language_code	ISO 639 defined language codes

Table 5.1: *Definition of information stored in the MySQL database.*

The PHP crawler written for the purpose of this thesis work is given in Appendix A.1. The longitude, latitude and radii where picked to catch as many tweets from Sweden as possible. Over 150,000 tweets per day where collected. Out of these, only about 30,000 corresponded to the five regions used for inference.

5.1.2 Information Retrieval

Ground truth data was readily available for five of Sweden's major cities. These where Göteborg, Malmö, Linköping, Norrköping and Helsingborg. Regarding forming the input keyword frequencies it was decided to filter the data by user defined location, as several of the cities in the ground truth data only had a few hundred tweets with geotag each day. The assumption made here is that most people tweet from the same area as their UDL, usually a city. The keywords used can be seen in the Glossary below. These where picked manually by considering words relevant to rain.

MySQL database queries used to form inputs for the algorithms can be seen in Appendix A.2. These are based on the formal mathematical expression formulated in Section 2.3.

5.2 Results

The results are displayed and discussed in the same order as the theory in Chapters 2 and 3. First, the linear parametric model (Section 5.2.1) is displayed and discussed, then the nonlinear parametric (Section 5.2.2). The final part is spent on interpreting and discussing the nonparametric regression, i.e. the GP (Section 5.2.3). The data was divided into a training and validation (CV) set of $N = 5 \cdot 42 =$

Glossary

Swedish	English
regn	rain
regnar	raining
ösregn	pouring rain
blöt	wet
moln	cloud
paraply	umbrella
skur	shower
åska	thunder

210 data points, and a test set with $M = 5 \cdot 11 = 55$ data points. The data used for CV consists of real data from June 6, 2012 to July 17, 2012 and data used for assessment of data from July 18, 2012 to July 28, 2012. The training and validation set was randomly split into 10 seperate subsets which was used for 10-fold CV, see Chapter 4, to select an optimal value of the hyperparameters. The function was then relearned with these hyperparameters and all training data. The final estimated model was then assessed on the test data set. Each section displays this result with six plots, one for each city and one of the total result with all test data concatenated into one data set.

5.2.1 Linear Model

In this section the results for linear regression, based on the model described in Section 2.3.1, will be displayed and discussed. Shrinkage methods were employed for a better regression and generalization result as the LSE was found to perform rather poorly. First results from RR are displayed then results obtained by the LASSO method.

Ridge Regression

Figure 5.1 displays the estimated error, using CV, as a function of the hyperparameter λ. Figure 5.2 displays the model assessed on all available test data. Figure 5.3, shows estimated model applied to independent test data, July 18 to July 28, 2012 , for each city respectively.

As can be seen in the plots, the predictor does a decent job of following the actual test data output. The total RMSE of 2.64 mm and the plots seem to confirm that there is a correlation that can be used to estimate the rainfall rate. The estimated optimal parameters $\widehat{\beta}_{\text{RR}}$ are given in Table 5.2, with the left hand side being the keyword corresponding to the individual parameters.

The optimal regularization parameter, for RR, selected by CV was approximately $\lambda = 0.3 \cdot 10^{-3}$. RR usually shrinks all parameters, whereas LASSO decreases parameters to zero. This can be clearly seen when comparing Tables 5.2 and 5.3.

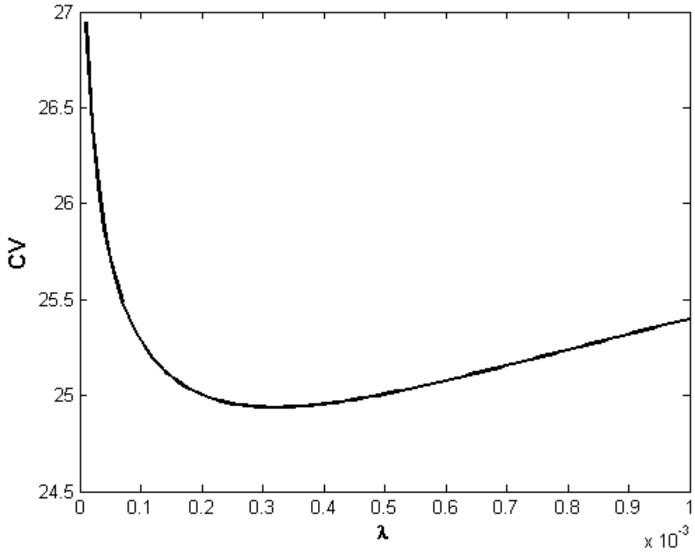


Figure 5.1: *Ridge regression cross-validation result for hyperparameter* λ

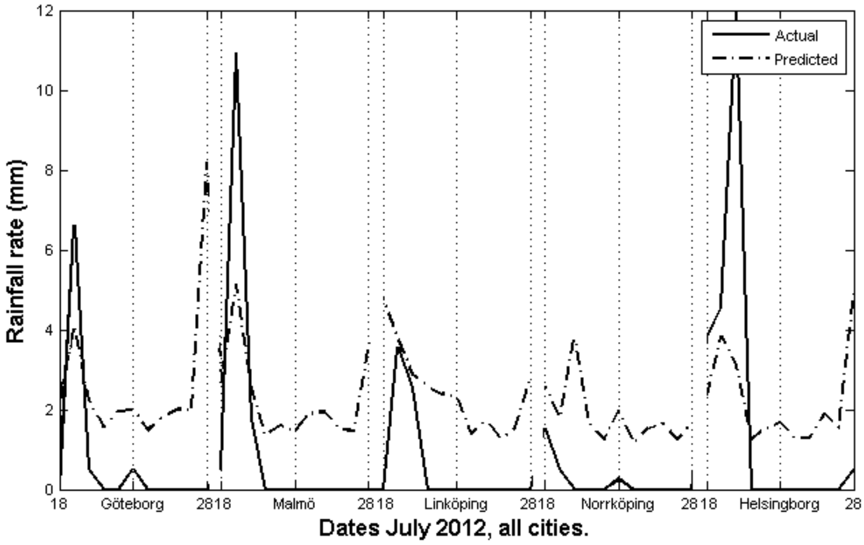
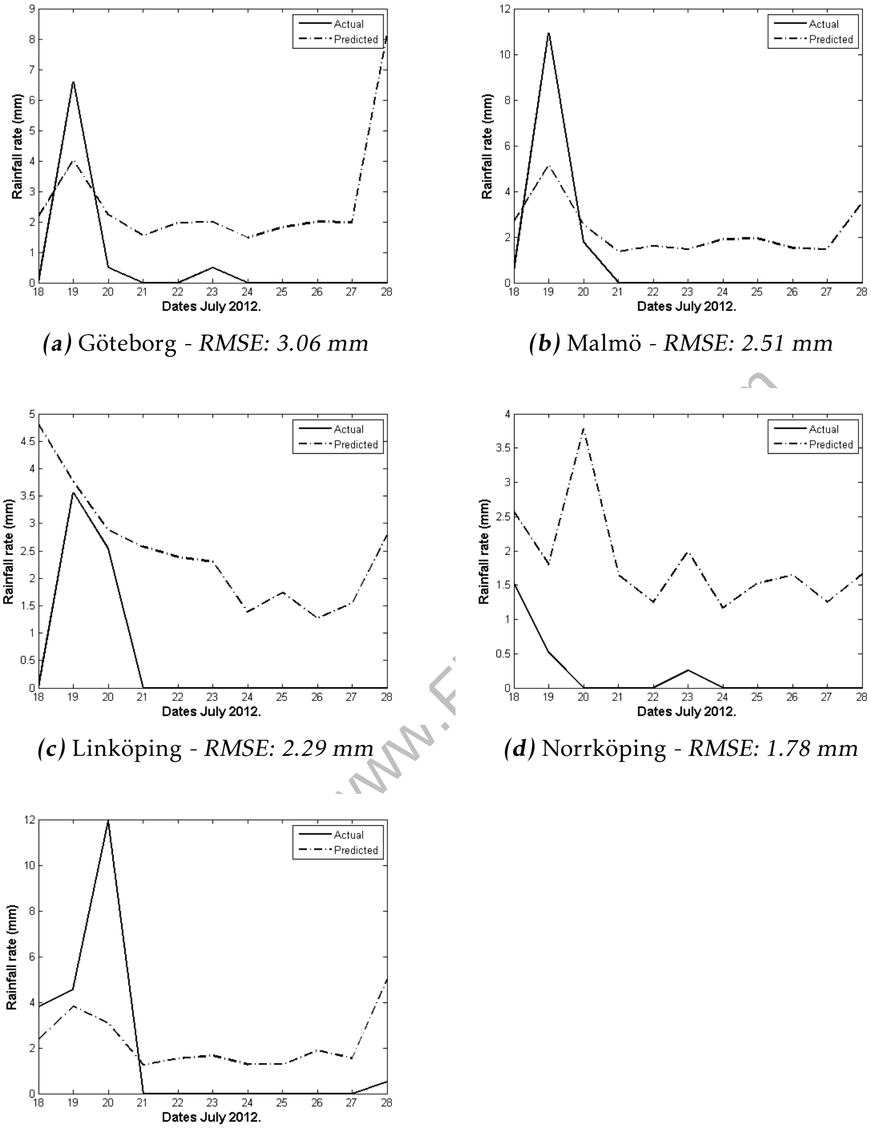


Figure 5.2: *Total ridge regression result on test data. RMSE: 2.64 mm*



(a) Göteborg - *RMSE: 3.06 mm*

(b) Malmö - *RMSE: 2.51 mm*

(c) Linköping - *RMSE: 2.29 mm*

(d) Norrköping - *RMSE: 1.78 mm*

(e) Helsingborg - *RMSE: 3.26 mm*

Figure 5.3: *Ridge regression results on independent test data.*

Table 5.2: *Parameter estimation results for* RR.

Parameter	Value	Parameter	Value
intercept (β_0)	1.2686	regn	604.38
regnar	281.6766	ösregn	40.9886
blöt	137.6109	moln	-34.4233
paraply	76.8176	skur	-1.2751
åska	18.5910	regn^2	5.0786
regnar^2	1.0978	ösregn^2	-0.0155
blöt^2	0.3931	moln^2	0.1107
paraply^2	0.1079	skur^2	-0.0188
åska^2	0.1895	regn · regnar	2.6487
regnar · ösregn	0.2098	ösregn · blöt	0.0603
blöt · moln	0.1390	moln · paraply	-0.0377
paraply · skur	-0.0028	skur · åska	-0.0313
ösregn · regn	0.1365	blöt · regnar	.4933
moln · ösregn	0.0654	paraply · blöt	0.0223
skur · moln	-0.0228	åska · paraply	-0.0047
blöt · regn	0.9208	moln · regnar	0.0709
paraply · ösregn	0.0371	skur · blöt	0.0272
åska · moln	0.3566	moln · regn	-0.3009
paraply · regnar	0.1356	skur · ösregn	-0.0023
åska · blöt	0.1004	paraply · regn	0.3752
skur · regnar	-0.0321	åska · ösregn	0.0485
skur · regn	0.0523	åska · regnar	0.0424
åska · regn	-0.6342		

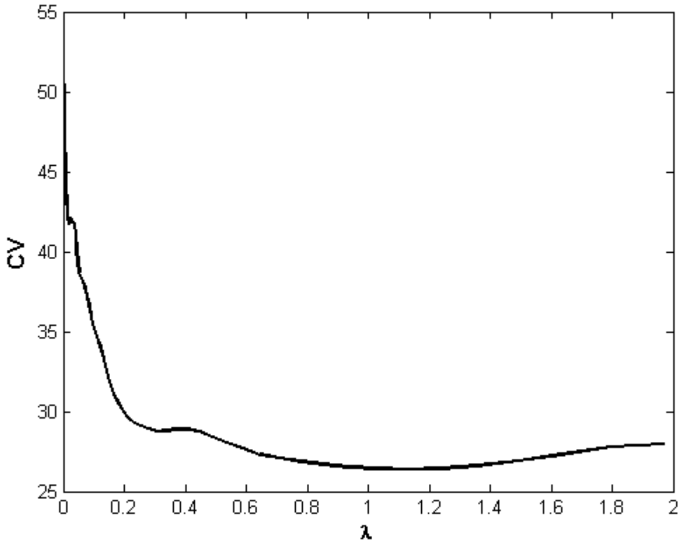


Figure 5.4: *LASSO cross-validation result for hyperparameter λ*

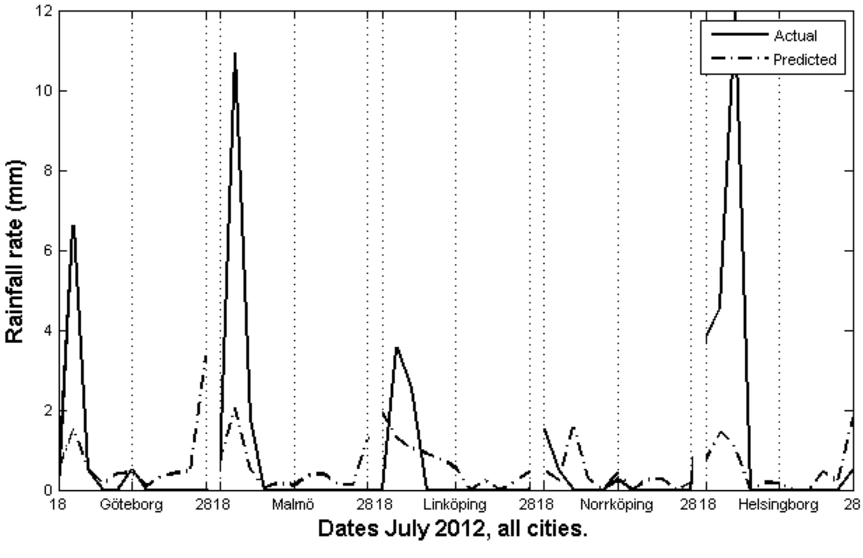


Figure 5.5: *Total LASSO result on test data. RMSE: 2.24 mm*

LASSO

Figure 5.4 shows CV estimated error as a function of λ for LASSO. The optimal value for lambda in this instance was found to be approximately 1.12 leading to most of the parameters to be set to zero. The next figure, Figure 5.5, displays estimated model evaluated on the test set with a total RMSE of 2.24 mm, an improvement over RR. However, this result is most likely due to the amount of zeros in the test data set output. As such the measure favours models that generally outputs a lower estimate. This is discussed further in Chapter 6. The model evaluated on test data for each city individually can be seen in Figure 5.6.

In Table 5.3, the estimated, non-zero, parameter values are shown. Only two parameters have survived the aggressive regularization.

Parameter	Value
regn	424
blöt^2	10447

Table 5.3: *Parameter estimation results for LASSO.*

Generally the linear regressions seem to do a decent job of catching the overall shapes of the data. There are, however, some discrepancies worth mentioning. Figures 5.3a and 5.6a show very high predicted value for July 28, 2012. Another, albeit slightly less serious, one is Norrköping July 20, 2012. After some study of the actual data it seems the input frequencies of keywords where rather high given that there was no rain that day. Further discussion on the impact of the data can be found in Chapter 6.

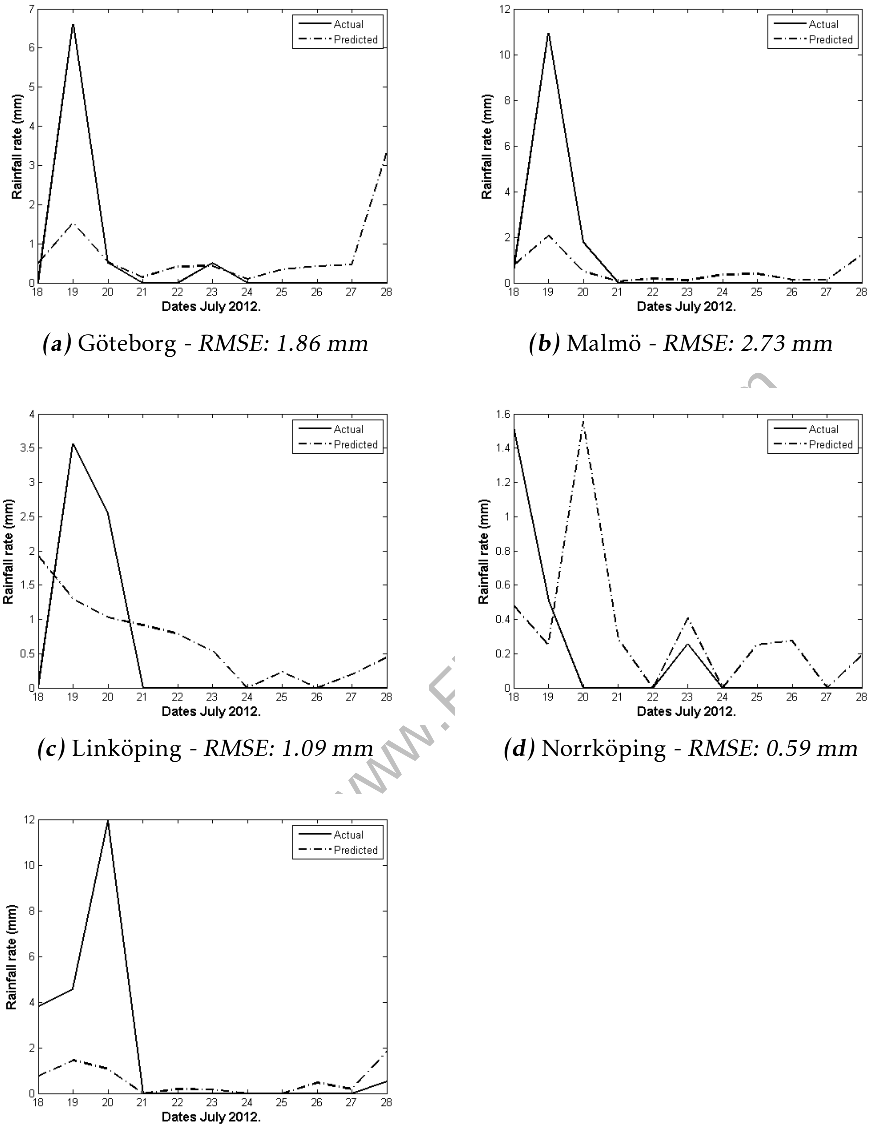
5.2.2 Nonlinear Model

The nonlinear parametric model used is defined in Section 2.3.2 and (2.14). Regularization was again applied for a better generalization result. With $q = 1$ fixed there were two hyperparameters to estimate using CV. A plot with estimated error as a function of the two hyperparameters, M (number of hidden units) and λ (regularization parameter), is shown in Figure 5.7. This is followed by Figure 5.8 which displays the total result of the neural network applied to the novel test data set. Lastly the results for each individual city is shown in Figure 5.9.

Regarding the RMSE there is a slight improvement over RR, but the regression using LASSO generalizes best in this case. Also worth pointing out is that the NN for Norrköping actually contains a negative inferred value. Knowing the non-negative properties of rainfall rate, for applications, the actual prediction would have to be amended to $\max(0, \hat{f})$.

5.2.3 Nonparametric Model

The nonparametric models used for prediction of daily rainfall rates are the two defined in Section 3.6, i.e. two zero-mean GP with SE and RQ kernels respectively. The first set of figures show the results using the SE kernel. Figure 5.10 displays



(a) Göteborg - *RMSE: 1.86 mm*

(b) Malmö - *RMSE: 2.73 mm*

(c) Linköping - *RMSE: 1.09 mm*

(d) Norrköping - *RMSE: 0.59 mm*

(e) Helsingborg - *RMSE: 3.56 mm*

Figure 5.6: *LASSO results on independent test data.*

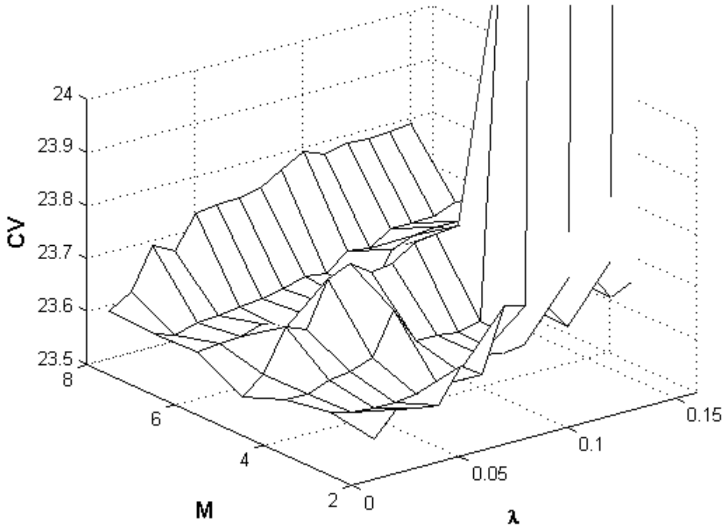


Figure 5.7: Neural network cross-validation result for hyperparameters λ and M.

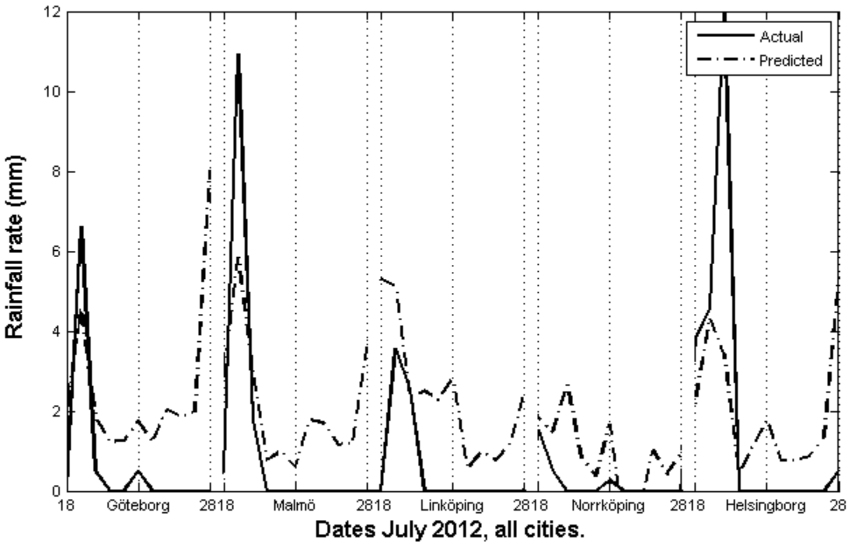
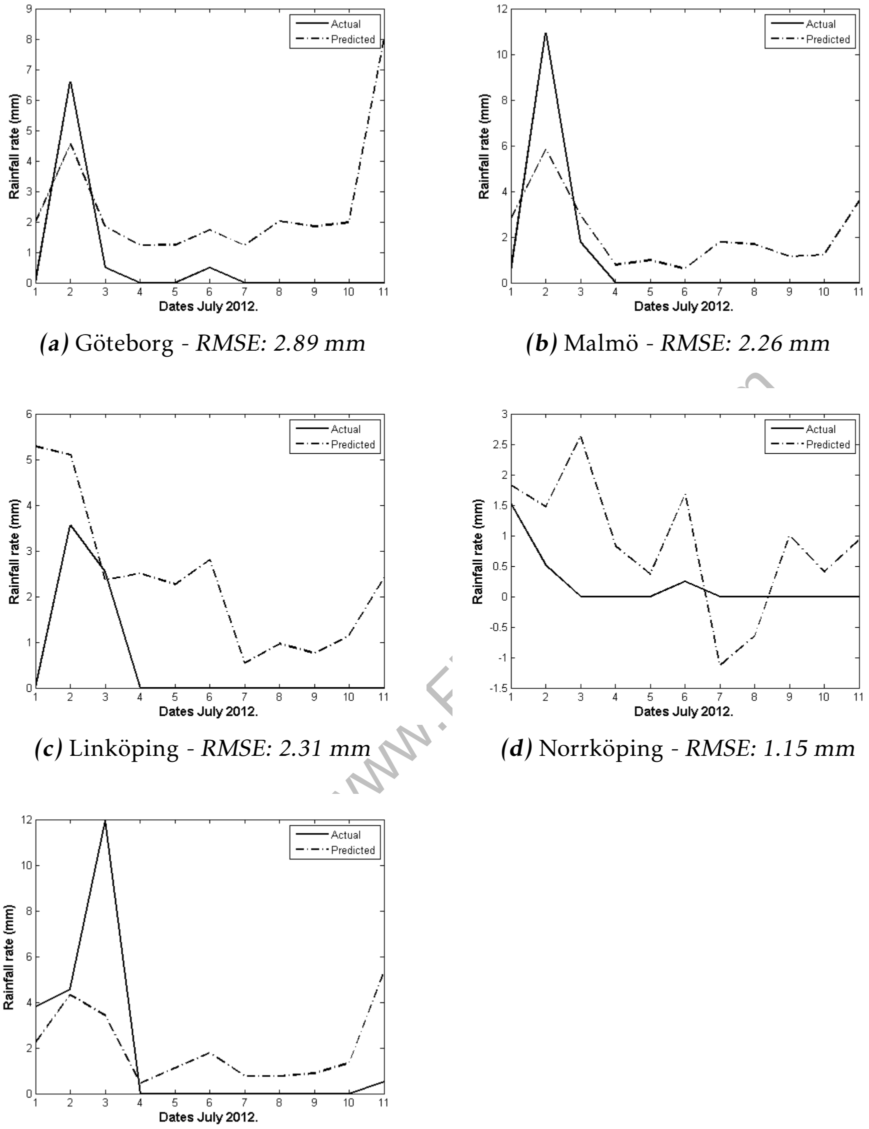


Figure 5.8: Total neural network result on test data. RMSE: 2.44 mm



(a) Göteborg - *RMSE: 2.89 mm*

(b) Malmö - *RMSE: 2.26 mm*

(c) Linköping - *RMSE: 2.31 mm*

(d) Norrköping - *RMSE: 1.15 mm*

(e) Helsingborg - *RMSE: 3.12 mm*

Figure 5.9: *Neural network results on independent test data.*

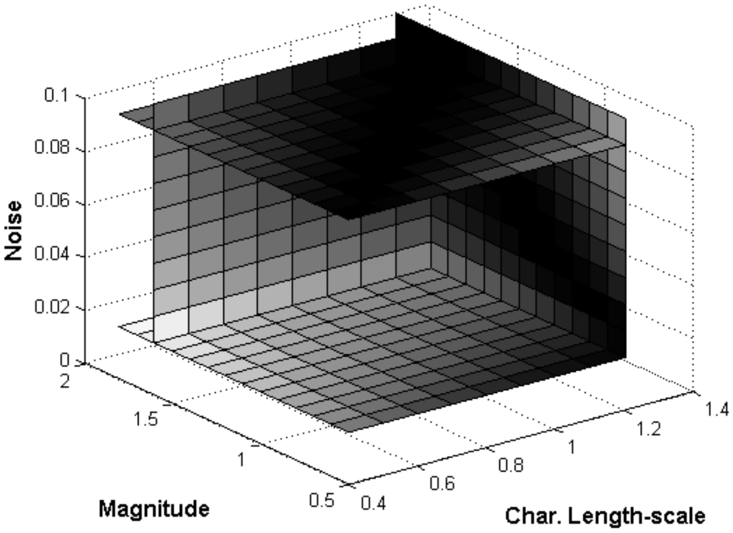


Figure 5.10: *GP, SE cov. function, cross-validation result for hyperparameters m (magnitude), l (char. length-scale) and σ (noise).*

a slice, since it is a function of 3 variables, of the CV. It is a linear grayscale map where darker (black) shades correspond to lower values and brighter (white) correspond to higher values. The slice is made with hyperparameters corresponding to the minimal value of the estimated error according to CV. Figure 5.11 shows the total results on test data. Figure 5.12 contains the five plots with the GP model assessed for each city independently.

The second model concerns inference using the RQ covariance function. As this problem formulation contains four hyperparameters it is difficult to illustrate CV estimation of the error in a plot as functions of the parameters. The optimal values for the hyperparameters was as before taken to correspond to the parameters minimizing CV over a 4D-grid. Figure 5.13 depicts the result from evaluating the GP model on all independent test data, Figure 5.14 shows the same model and data but divided up by city. From an RMSE point of view, the results seem to be on par with both the NN and GP with SE covariance function. Worth pointing out is that no negative inferred value is present in the SE model. However, the discrepancies in the Göteborg July 28, 2012 and Norrköping July 20, 2012 test data are still there.

5.2.4 Summary

To summarize the results obtained in these experiments Table 5.4 shows the estimated RMSE for each model. RMSE are displayed for each city independently and also the total. Minimum values for each column are in boldface and the

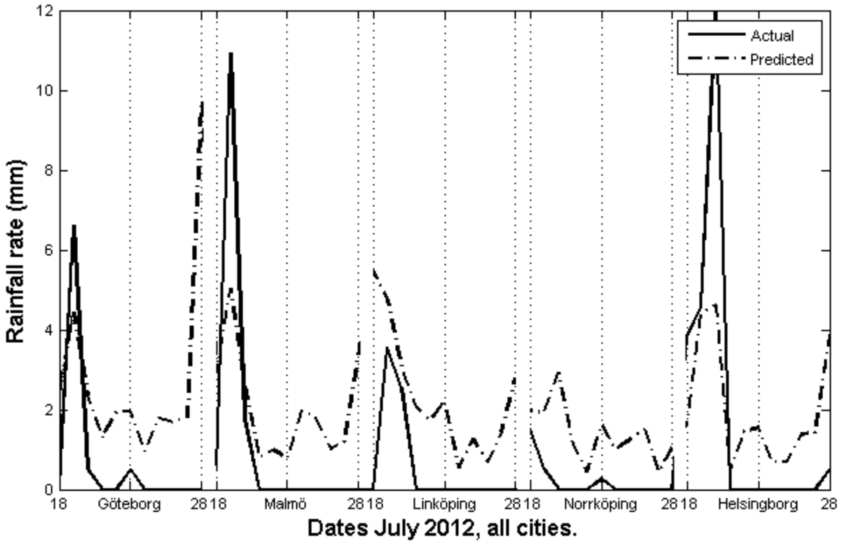
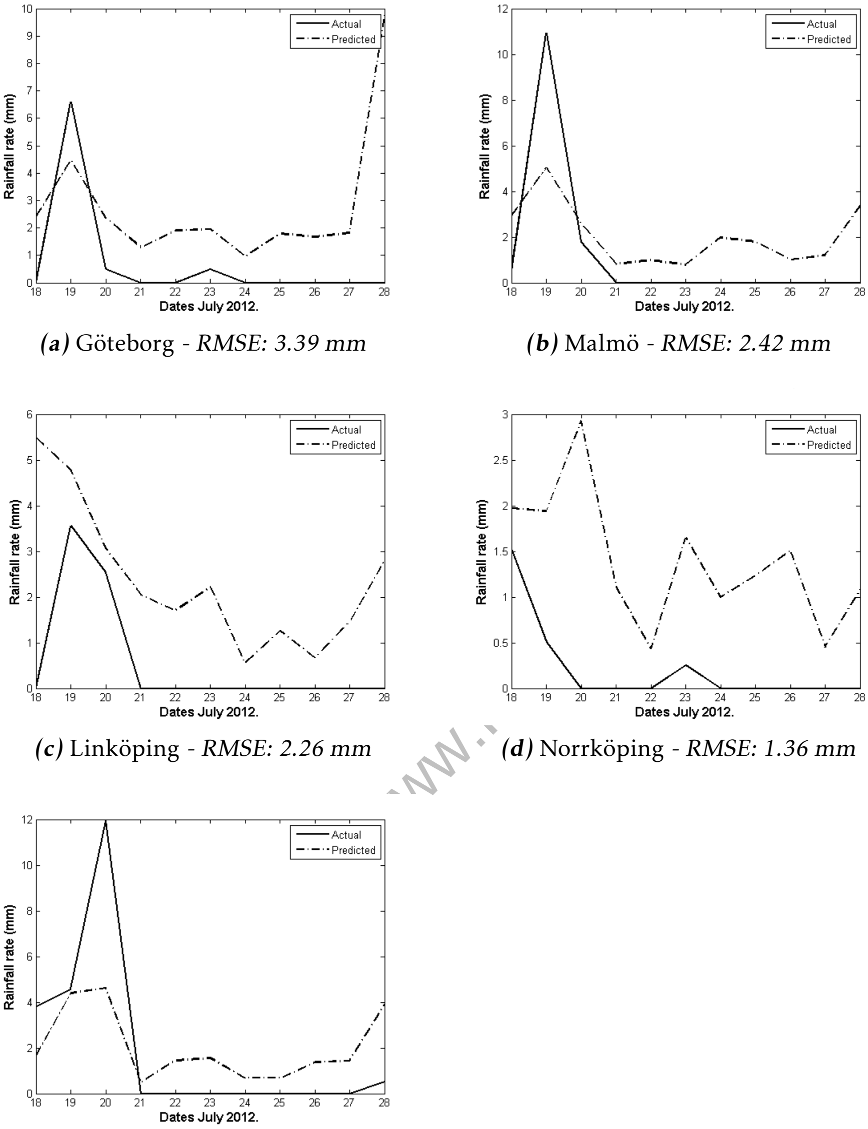


Figure 5.11: *Total GP, SE cov. function, result on test data. RMSE: 2.51 mm*

corresponding model in the last row.

Model	Göteborg	Malmö	Linköping	Norrköping	Helsingborg	Total
RR	3.06 mm	2.51 mm	2.29 mm	1.78 mm	3.26 mm	2.64 mm
LASSO	**1.86 mm**	2.73 mm	**1.09 mm**	**0.59 mm**	3.56 mm	**2.24 mm**
NN	2.89 mm	2.26 mm	2.31 mm	1.15 mm	3.12 mm	2.44 mm
GP (SE)	3.39 mm	2.42 mm	2.26 mm	1.36 mm	**2.68 mm**	2.51 mm
GP (RQ)	3.21 mm	**2.20 mm**	2.28 mm	1.23 mm	3.14 mm	2.52 mm
Min.	LASSO	GP (RQ)	LASSO	LASSO	GP (SE)	LASSO

Table 5.4: *Total and regional RMSE results for each model used.*



(a) Göteborg - *RMSE: 3.39 mm*

(b) Malmö - *RMSE: 2.42 mm*

(c) Linköping - *RMSE: 2.26 mm*

(d) Norrköping - *RMSE: 1.36 mm*

(e) Helsingborg - *RMSE: 2.68 mm*

Figure 5.12: *Gaussian process, SE cov. function, results on independent test data.*

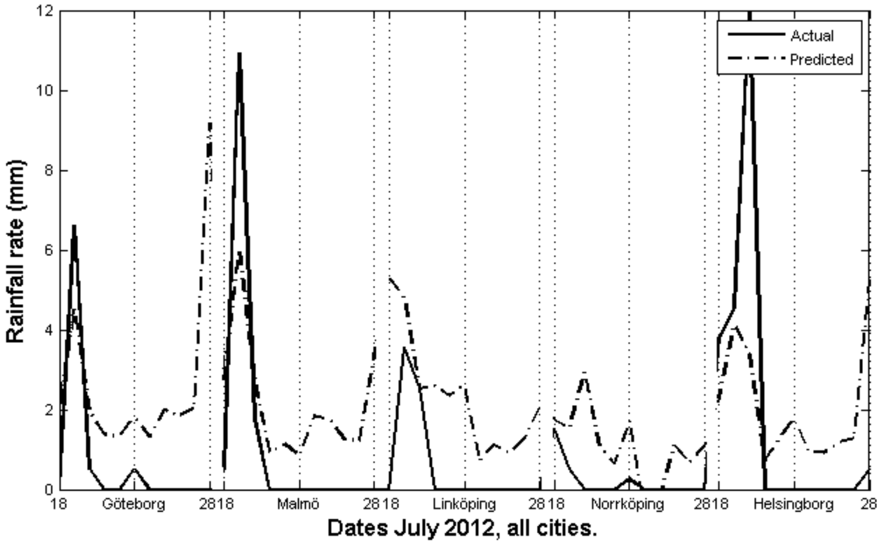
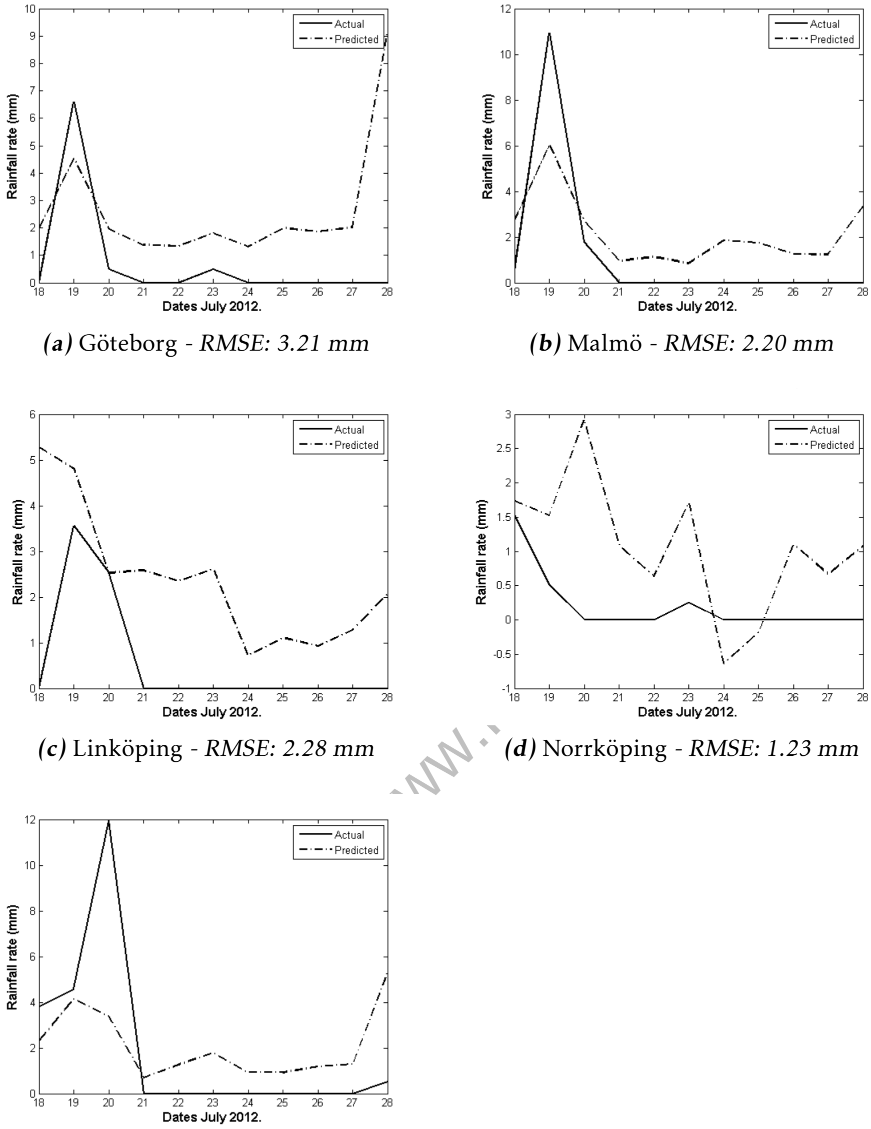


Figure 5.13: *Total* GP*,* RQ *cov. function, result on test data. RMSE: 2.52 mm*



(a) Göteborg - *RMSE: 3.21 mm*

(b) Malmö - *RMSE: 2.20 mm*

(c) Linköping - *RMSE: 2.28 mm*

(d) Norrköping - *RMSE: 1.23 mm*

(e) Helsingborg - *RMSE: 3.14 mm*

Figure 5.14: *Gaussian process,* RQ *cov. function, results on independent test data.*

6

Concluding Remarks

In this chapter the results obtained in the thesis will be summarized, compared and discussed. Section 6.1 summarizes and compares the results obtained in Chapter 5. Section 6.2 discusses the data used and the method of data collection. Section 6.3 briefly mentions future work and potential improvements to the results obtained in this thesis.

6.1 Conclusions

According to Section 5.2 the overall best result, in terms of RMSE, was given by regression using LASSO with RMSE = 2.24 mm. RMSE for all models evaluated on independent test data can be seen in Table 6.1. The RMSE for the constant zero-prediction and mean of the training outputs, $\mathbf{y}$, are also shown for comparison.

Model	RMSE
RR	2.64 mm
LASSO	2.24 mm
NN	2.44 mm
GP (SE cov.)	2.51 mm
GP (RQ cov.)	2.52 mm
0	2.58 mm
mean($\mathbf{y}$)	3.04 mm

Table 6.1: *RMSE for the various models evaluated on test data.*

However, using this measure for assessment does not give the full picture. The RMSE, in this case, greatly favours a model that generally estimates a lower rainfall rate. This as the test data and rainfall rate in general, contain a lot of zeros. This can also be seen in the RMSE result for the constant zero- and mean prediction. Constant zero-prediction actually gives a lower RMSE than RR. RMSE results for the other models indicate that there is value in using information contained in microblog data for inference. However, the LASSO model does a fairly poor job of estimating rainfall rate on days which have seen a lot of rain. The nonlinear and nonparametric models, especially the GP, does seem to do a better job in this sense from looking at the plots of estimated rainfall rates in Section 5.2. Regarding the discrepancies in the test data from Göteborg on July 28, 2012 and Norrköping July 20, 2012, it is most likely due to the fairly noisy data and perhaps a data misalignment, discussed further in Section 6.2. Just by looking at the shapes of the plots, for nowcasting the rainfall rate, the RR, NN and GP seem to give very similar results. It is only the LASSO that stands out which is explained by the aggressive regularization.

Given that the quality of collected data is far from perfect, the different regression methods and results still seem to indicate that there is information contained within the tweets that can be used for inference of everyday measures like rainfall rates or more interesting ILI rates.

6.2 Data Properties

Good data should be a paramount consideration in statistical learning and regression. This section will briefly discuss the shortcomings of the data used in this thesis and ways of how these can be remedied with respect to the effect they will have on the inference result.

The first vital assumption made is that user defined location (UDL) is also the current position of the person sending the tweet. This is of course not always true, and in general might not be true at all. Filtering by the geotag (longitude and latitude) can remidy this problem. However, this means that the subset of tweets are limited to about 5% of the total amount of tweets collected. Going one step further, a classification algorithm can be applied to tweets found by filtering by UDL. Using this way to find relevant tweets and joining this information with geotagged tweets can significantly improve regression results. The basic idea would be to consider each user as a kind of sensor as in Sakaki et al. [2010]. For this case the classification algorithm, a $1-0$ regression, would assign each tweet as relevant (1) and irrelevant (0).

Another problem is potential data misalignment. The ground truth rainfall rate data was a measurement of the total rainfall amount each day based on Coordinated Universal Time (UTC, 0000Z - 2359Z) [1]. However, the data used for regression was based on Swedish time zone, Central European Summer Time (CEST), which is UTC +02:00. This means rainfall occuring at, for example, July 28, 2012 01:00 UTC will be considered as a part of the rainfall rate on the 28th of July, but

[1] http://www7.ncdc.noaa.gov/CDO/GSOD_DESC.txt, (August 17th, 2012)

tweets discussing this until 02:00 UTC will affect the July 27, 2012 inputs. This was not as big of a problem as the amount of tweets sent during this time frame was comparatively few. This can be partly avoided by performing more complex and time consuming queries to the MySQL database. However, one problem will still remain. That is the amount of tweets collected during late night and early morning. Tweets collected between 02:00 and 05:00 CEST constitutes less than 5% of the total number collected each day, even though this time interval accounts for 12.5% of the day. It is questionable if these observations are statistically significant and if they can represent the rainfall rate during this time frame. This needs further investigation. One remedy to this problem is collecting ground truth rainfall data only for the daytime time interval.

The last input data problem discussed will be the assumption that a tweet with a certain keyword is relevant for inference. As an illustrating example the small discrepancy in the prediction of rainfall rate in Norrköping on July 20, 2012 will be used, see Section 5.2. Table 6.2 contains the actual tweets found by filtering by the *regn* (rain) keyword.

Nr.	Tweet text
1	bara hoppas vi får bra väder, inte för varmt och inget regn tack
2	Regn, hagel och solsken. Sol är ju alltid sol, liksom. http://t.co/93fWG1LC
3	Regn, so what??? http://t.co/7RxbLsCx
4	Första semesterdagen. 23-34 grader varmt sol. Regn i 10 minuter. Helt ok.
5	1-1 i halvtid påRosvalla. Dagoberto pangade in kvitteringen snyggt framklackad av Marcinho... Och nu blev det regn och åska.
6	@telenaten Inget regn?

***Table 6.2:** Tweet text for keyword* regn *(rain) in Norrköping July 20, 2012.*

A total of 6 tweets contained the keyword *regn* (rain) on this day. Out of these 6, number 1 and 6 are most likely not relevant. This as the first one discusses hopes for how the weather will turn out and the last one is from a conversation where the user asks "No rain?". Tweet number 5 might also be irrelevant as it contains references to a football game played in Rosvalla, Nyköping which is far away from Norrköping. This would mean that potentially 50% of all tweets might be irrelevant and so can cause the irregularity in this test data. However, this problem is more difficult to handle as it might require some additional preprocessing in the form of classification or natural language processing.

6.3 Future Work

This thesis has illustrated how information can be gleaned from unstructured user generated content from social networks. Further work will focus on nowcasting ILI rates and the information retrieval process, refining the data used for regression as explained in Section 6.2. With better data, a more complex model might be necessary. Selection of keywords could also include possibly negatively correlated keywords as well as more nonlinear base functions like the logarithm and exponential function. It might also be interesting to take a completely Bayesian approach to model selection instead of the CV that is performed in this thesis. This would require, because of the high dimensionality of the problem, approximate methods like Markov Chain Monte Carlo (MCMC) methods. Another interesting approach would be to model the information contained in microposts using probabilistic topic models. For an introduction to, and review of topic models see [Blei, 2012]. Because of the strengths of the GP regression results, another interesting possibility would be to investigate sparse GP as described in Fox and Dunson [2012].

Because of the inherent properties of rainfall rate, discussed in Section 6.1, it would be interesting to try other methods that are more application specific. A first alternative would be alternate loss functions, for example the $\mathcal{L}^1$ norm. Thresholding could be another interesting addition to the models. Thresholding in this sense would be that when a model estimates below a certain value, that could be learned using CV, a zero is output as prediction. This would also solve the problem of negative predictions from the models very elegantly.

In summary, topics for future work are:

ILI rates Static model estimation of ILI rates using unstructured textual data from microblogging and other online social media.

Keywords Investigate keywords selction and potentially negatively correlated keywords.

Base functions Nonlinear base functions like the log function could potentiall improve nowcasting results.

Refine data Filter and pre-process the regression data, see Section 6.2.

Bayesian inference MCMC methods for model selection and validation.

Topic models A different approach to model the data using probabilistic topic models.

Sparse GP Using sparse GP regression.

Alternate loss functions Use the $\mathcal{L}^1$ norm for training.

Thresholding Prediction estimates under a certain value are set to zero.

A

Code

A.1 PHP Script

This section contains the PHP script, Tweets crawler, written for the purpose of this thesis. Note that each city is queried for the 300 latest tweets at least once an hour, and Stockholm as many times as 18 times an hour.

```
<?php
$sthlm = "59.310768,18.061523,30km"; //Stockholm

$gbg = "57.663035,11.953125,25km"; //Goteborg

$malmo = "55.595419,13.287964,25km"; //Malmo

$uppsala = "59.864815,17.639923,20km"; //Uppsala
$vasteras = "59.616380,16.546783,15km"; //Vasteras
$orebro = "59.278511,15.222931,30km"; //Orebro
$linkoping = "58.406748,15.611572,15km"; //Linkoping
$helsingborg = "56.044048,12.700882,15km"; //Helsingborg
$jonkoping = "57.776348,14.162750,40km"; //Jonkoping

$nkpg = "58.590446,16.174622,15km"; //Norrkoping
$lund = "55.595419,13.287964,25km"; //Malmo
$umea = "63.826134,20.272522,50km"; //Umea
$gavle = "60.676542,17.142792,25km"; //Gavle
$boras = "57.714418,12.947388,10km"; //Boras
$eskilstuna = "59.373791,16.512451,15km"; //Eskilstuna
$sodertalje = "59.193516,17.626877,15km"; //Sodertalje
```

```
$karlstad = "59.383234,13.506317,50km"; //Karlstad
$vaxjo = "56.877123,14.809570,50km"; //Vaxjo
$halmstad = "56.670189,12.860870,30km"; //Halmstad
$sundsvall = "62.389733,17.303467,50km"; //Sundsvall
$lulea = "65.586288,22.167664,50km"; //Lulea

$cities = array($gbg , $sthlm , $malmo , $uppsala ,
   $sthlm , $gbg ,$vasteras ,$orebro ,$sthlm ,$linkoping
   ,$gbg ,$sthlm ,$malmo ,$helsingborg ,$sthlm ,$gbg ,
   $jonkoping ,$nkpg ,$sthlm ,$lund ,$gbg ,$sthlm ,$malmo
    ,$umea ,$sthlm ,$gbg ,$gavle ,$boras ,$sthlm ,
   $eskilstuna ,$gbg ,$sthlm ,$malmo ,$uppsala ,$sthlm ,
   $gbg ,$vasteras ,$orebro ,$sthlm ,$linkoping ,$gbg ,
   $sthlm ,$malmo ,$helsingborg ,$sthlm ,$gbg ,$jonkoping
    ,$sodertalje ,$sthlm ,$karlstad ,$gbg ,$sthlm ,$malmo
    ,$vaxjo ,$sthlm ,$gbg ,$halmstad ,$sundsvall ,$sthlm
   ,$lulea );
while (1){
for ($k = 0; $k <= 59; $k++){
  $count = 0;
  $current_time = microtime(true);

  for ($i = 1; $i <= 10; $i++){
    $request = 'http://search.twitter.com/search.json?rpp
       =100&geocode="' . $cities[$k] . '"&page=' . $i;
    $response = file_get_contents($request);
    $jsonobj = json_decode($response);

    if($jsonobj != null){
      $con = mysql_connect('localhost', 'DATABASE', '
         PASSWORD');
      if (!$con){
        die('Could not connect: ' . mysql_error());
      }
      foreach($jsonobj->results as $item){
      $id = $item->id;
      $created_at = $item->created_at;
      $created_at = strtotime($created_at);
      $mysqldate = date('Y-m-d H:i:s',$created_at);
      $from_user = mysql_real_escape_string($item->
         from_user);
      $from_user_id = $item->from_user_id;
      $text = mysql_real_escape_string($item->text);
      $geo = $item->geo;
      if (isset($geo)){
        $geom = $geo->coordinates[0] . ’,’ . $geo->
           coordinates[1];
        $locm = "";
      } else {
        $geom = "";
        $locm = mysql_real_escape_string($item->location)
           ;
      }
      $to_user_id = $item->to_user_id;
      if($to_user_id==""){ $to_user_id = 0; }

      mysql_select_db("tweets2", $con);

      mysql_query("SET NAMES ’utf8’") or die(mysql_error
         ());
      mysql_query("SET CHARACTER SET ’utf8’") or die(
         mysql_error());
      $query = mysql_real_escape_string($query);
      $query = "INSERT INTO tweets_1 VALUES ($id,’
         $mysqldate’,’$from_user’,$from_user_id,’$text’,’
         $locm’,’$geom’,$to_user_id) ON DUPLICATE KEY
         UPDATE id = id";
      $result = mysql_query($query);
      if (!$result){
        $message = ’Invalid query: ’ . mysql_error() . "\
           n";
        $message .= ’Whole query: ’ . $query;
        die($message);
      }
    }
  }
  mysql_close($con);
}
  var_dump(time_sleep_until($current_time + 60));
}
}
?>
```

A.2 MySQL Queries

The MySQL queries performed for Göteborg to form input frequencies are shown below. Inputs are formed in the same way for all the other regions, just switch region-specific words.

```
/* Goteborg */
/* KW: REGN */
TRUNCATE data;

INSERT INTO data SELECT DATE(created_at) , COUNT(*)
FROM tweets
WHERE location LIKE  '%ögteborg%'
AND TEXT LIKE  '%regn%'
AND TEXT
RLIKE  '[[:<:]]regn[[:>:]]'
AND TEXT NOT LIKE  'RT%'
GROUP BY DATE(created_at);

INSERT IGNORE INTO data SELECT *,0 FROM date;

SELECT id,',',count FROM data INTO OUTFILE 'FILEPATH';


/* KW: REGNAR */
TRUNCATE data;

INSERT INTO data SELECT DATE(created_at), COUNT(*)
FROM tweets
WHERE location LIKE  '%ögteborg%'
AND TEXT LIKE  '%regnar%'
AND TEXT
RLIKE  '[[:<:]]regnar[[:>:]]'
AND TEXT NOT LIKE  'RT%'
GROUP BY DATE(created_at);

INSERT IGNORE INTO data SELECT *,0 FROM date;

SELECT id,',',count FROM data INTO OUTFILE 'FILEPATH';


/* KW: OSREGN */
TRUNCATE data;

INSERT INTO data SELECT DATE(created_at), COUNT(*)
FROM tweets
WHERE location LIKE  '%ögteborg%'
AND TEXT LIKE  'ö%sregn%'
AND TEXT
RLIKE  'ö[[:<:]]sregn[[:>:]]'
AND TEXT NOT LIKE  'RT%'
GROUP BY DATE(created_at);
```

```
INSERT IGNORE INTO data SELECT *,0 FROM date;

SELECT id,',',count FROM data INTO OUTFILE 'FILEPATH';


/* KW: BLOT */
TRUNCATE data;

INSERT INTO data SELECT DATE(created_at), COUNT(*)
FROM tweets
WHERE location LIKE  '%ögteborg%'
AND TEXT LIKE  '%öblt%'
AND TEXT
RLIKE  '[[:<:]]öblt[[:>:]]'
AND TEXT NOT LIKE  'RT%'
GROUP BY DATE(created_at);

INSERT IGNORE INTO data SELECT *,0 FROM date;

SELECT id,',',count FROM data INTO OUTFILE 'FILEPATH';


/* KW: MOLN */
TRUNCATE data;

INSERT INTO data SELECT DATE(created_at), COUNT(*)
FROM tweets
WHERE location LIKE  '%ögteborg%'
AND TEXT LIKE  '%moln%'
AND TEXT
RLIKE  '[[:<:]]moln[[:>:]]'
AND TEXT NOT LIKE  'RT%'
GROUP BY DATE(created_at);

INSERT IGNORE INTO data SELECT *,0 FROM date;

SELECT id,',',count FROM data INTO OUTFILE 'FILEPATH';



/* KW: PARAPLY */
TRUNCATE data;

INSERT INTO data SELECT DATE(created_at), COUNT(*)
FROM tweets
```

```
WHERE location LIKE '%ögteborg%'
AND TEXT LIKE '%paraply%'
AND TEXT
RLIKE '[[:<:]]paraply[[:>:]]'
AND TEXT NOT LIKE 'RT%'
GROUP BY DATE(created_at);

INSERT IGNORE INTO data SELECT *,0 FROM date;

SELECT id,',',count FROM data INTO OUTFILE 'FILEPATH';


/* KW: SKUR */
TRUNCATE data;

INSERT INTO data SELECT DATE(created_at), COUNT(*)
FROM tweets
WHERE location LIKE '%ögteborg%'
AND TEXT LIKE '%skur%'
AND TEXT
RLIKE '[[:<:]]skur[[:>:]]'
AND TEXT NOT LIKE 'RT%'
GROUP BY DATE(created_at);

INSERT IGNORE INTO data SELECT *,0 FROM date;

SELECT id,',',count FROM data INTO OUTFILE 'FILEPATH';



/* KW: ASKA */
TRUNCATE data;

INSERT INTO data SELECT DATE(created_at), COUNT(*)
FROM tweets
WHERE location LIKE '%ögteborg%'
AND TEXT LIKE 'å%ska%'
AND TEXT
RLIKE 'å[[:<:]]ska[[:>:]]'
AND TEXT NOT LIKE 'RT%'
GROUP BY DATE(created_at);

INSERT IGNORE INTO data SELECT *,0 FROM date;

SELECT id,',',count FROM data INTO OUTFILE 'FILEPATH';
```

```
/* KW: TOT */
SELECT DATE(created_at) ,',',COUNT(*)
FROM tweets
WHERE location LIKE  '%ögteborg%'
AND TEXT NOT LIKE  'RT%'
GROUP BY DATE(created_at) INTO OUTFILE 'FILEPATH';
```

Bibliography

H. Achrekar, A. Gandhe, R. Lazarus, Ssu-Hsin Yu, and Benyuan Liu. Predicting flu trends using twitter data. In *IEEE INFOCOM 2011 - IEEE Conference on Computer Communications Workshops*, pages 702 –707, april 2011. doi: 10.1109/INFCOMW.2011.5928903. Cited on page 2.

Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. Twitter improves seasonal influenza prediction. In *HEALTHINF*, pages 61–70, Vilamoura, Algarve, Portugal, 2012. SciTePress. Cited on page 2.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC, 2006. Cited on pages 9 and 10.

David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133826. URL `http://doi.acm.org/10.1145/2133806.2133826`. Cited on page 44.

J. Bollen and Huina Mao. Twitter mood as a stock market predictor. *Computer*, 44(10):91 –94, oct. 2011. Cited on page 2.

J. Bollen, H. Mao, and X.-J. Zeng. Twitter mood predicts the stock market. *ArXiv e-prints*, October 2010. Cited on page 2.

Lingji Chen, Harshavardhan Achrekar, Benyuan Liu, and Ross Lazarus. Vision: towards real time epidemic vigilance through online social networks: introducing sneft – social network enabled flu trends. In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing; Services: Social Networks and Beyond*, MCS '10, pages 4:1–4:5, San Francisco, California, 2010. ACM. Cited on page 2.

Cynthia Chew and Gunther Eysenbach. Pandemics in the age of twitter: Content analysis of tweets during the 2009 h1n1 outbreak. *PLoS ONE*, 5(11), 11 2010. Cited on page 2.

Neil M. Ferguson, Derek A.T. Cummings, Simon Cauchemez, Christophe Fraser, Steven Riley, Aronrag Meeyai, Sopon Iamsirithaworn, and Donald S. Burke.

Strategies for containing an emerging influenza pandemic in southeast asia. *Nature*, 437(7056):209–214, 2005. Cited on page 2.

E. B. Fox and D. B. Dunson. Multiresolution Gaussian Processes. *ArXiv e-prints*, September 2012. Cited on page 44.

Jeremy Ginsberg, Matthew H. Mohebbi, Rajan S. Patel, Lynnette Brammer, Mark S. Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457:1012–1014, 2009. Cited on page 2.

Fredrik Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur AB, 2010. Cited on page 2.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Elements of Statistical Learning*. Springer Science+Business Media, LLC, second edition, 2009. Cited on pages 7, 8, 10, and 22.

Vasileios Lampos and Nello Cristianini. Nowcasting events from the social web with statistical learning. *ACM Transactions on Intelligent Systems and Technology*, 3(3), 2011. Cited on page 2.

Thomas Lansdall-Welfare, Vasileios Lampos, and Nello Cristianini. Effects of the recession on public mood in the UK. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 1221–1226, New York, NY, USA, 2012. ACM. Cited on page 2.

Ira M. Longini Jr., Azhar Nizam, Shufu Xu, Kumnuan Ungchusak, Wanna Hanshaoworakui, Derek A.T. Cummings, and M. Elizabeth Halloran. Containing pandemic influenza at the source. *Science*, 309(5737):1083–1087, 2005. Cited on page 2.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. Cited on pages 13, 15, 16, 17, and 19.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. Cited on pages 2 and 42.

George A. F. Seber and Alan J. Lee. *Linear Regression Analysis*. John Wiley & Sons, Inc., second edition, 2003. Cited on page 7.

Linköpings universitet

Upphovsrätt

Copyright