Orlando Shigueo Ohashi Junior

Spatio-Temporal Prediction Methods

Doctoral Program in Computer Science
of the Universities of Minho, Aveiro and Porto

December 2012

Orlando Shigueo Ohashi Junior

Spatio-Temporal Prediction Methods

Thesis submitted to Faculty of Sciences of the University of Porto
for the Doctor Degree in Computer Science within the Joint Doctoral Program in
Computer Science of the Universities of Minho, Aveiro and Porto

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto

December 2012

Thesis Committee:

Prof. Donato Malerba, Universitá degli Studi di Bari "Aldo Moro"

Profa. Suzana Nascimento, Universidade Nova de Lisboa

Prof. José Luís Oliveira, Universidade de Aveiro

Prof. Paulo Jorge de Sousa Azevedo, Universidade do Minho

`Supervisor:` Prof. Luís Torgo, University of Porto

to my family

Acknowledgments

First of all, I would like to thank my supervisor Luís Torgo for his support and guidance.

I would also like to thank the researchers from LIAAD - INESC TEC for the good moments and friendship.

A special thanks to Rita and Jorge for all the funny moments and support in all these years.

I would like to thank the financial support of FCT through the PhD grant SFRH/BD/61795/2009 that allowed me to carry out this research project.

Thanks to my family, particularly my parents and brothers for the unconditional support, even from far away.

Finally, I cannot find words to express my gratitude to my fiancee Andréa for all the love and patience during all moments of this journey.

Abstract

The volume of data that is currently collected and stored was unthinkable a few years ago. This amount of information makes data and all phases of the process of collecting, storing and making sense of it, extremely important. Both academia and industry are working in this process. Data mining is a key component to help users to make sense of this huge amount of data. This research field includes a large set of tasks. This thesis addresses the problem of prediction using spatio-temporal data, i.e. data that are indexed both in time and in space.

The work presented in this thesis is driven by several real world applications: (i) monitoring and controlling water quality parameters within the water distribution network at Porto, Portugal; (ii) forecasting water consumption for a water distribution company in Spain; (iii) forecasting wind speed in some wind farm in the US; and (iv) filling in missing pixels of images.

Our work is organized in an incremental fashion by addressing different particularities of our applications. Concretely, we first address temporal prediction problems, then spatial prediction tasks and finally we focus on spatio-temporal data sets.

For temporal data we propose a new class of forecasting tasks that we name 2D-interval predictions, which consists on trying to obtain a forecast of the expected range of values for a future time interval. We formalize this task, propose a solution to it and establish the correct way of evaluating models for these tasks. Our extensive experimental tests show the advantage of our proposal for these tasks.

Regards, spatial data we address the problem of spatial interpolation by proposing a new methodology based on two key ideas: (i) transforming the problem into a regression task and (ii) describing the spatial dynamics by spatial indicators. This methodology

differentiates itself from the state of the art in that it allows the use of data from non-nearby regions to forecast the value for a certain location, thus somehow contradicting the first law of geography. We have extensively evaluated this methodology in problems of filling in missing pixels in photos. Our results show a clear advantage of our proposal when compared to the state of the art in spatial interpolation.

Finally, we proposed a new technique to improve the prediction accuracy in spatio-temporal data. Our technique differs from the most common approaches, in that is uses spatio-temporal properties of the data to improve the predictive accuracy. Namely, we propose a series of spatio-temporal indicators whose goal is to describe the spatio-temporal dynamics of the data for each location. We extensively test our technique using real world wind speed data, and we observed a clear advantage of our proposal when compared to several alternative methods that can be applied to these problems.

Resumo

O volume de dados recolhido e armazenado atualmente era inimaginável alguns anos atrás. Esta grande quantidade de dados faz com que o processo de recolha, armazenamento e extração de informação, seja fundamental. Tanto a academia como a indústria estão a trabalhar arduamente em todas as fases deste processo. A extração de conhecimento de dados é a componente chave para auxiliar os utilizadores na compreensão deste grande número de dados. Esta linha de investigação inclui um grande número de tarefas. Esta tese tem como foco o problema de previsão de dados espácio-temporais, ou seja, dados que são indexados tanto no tempo como no espaço.

O trabalho desenvolvido nesta tese foi guiado por várias aplicações reais: (i) a monitorização e controlo de parâmetros de qualidade de água, da companhia de distribuição de água do Porto, Portugal; (ii) a previsão do consumo de água, de uma companhia de distribuição de água em Espanha; (iii) a previsão da velocidade do vento de um parque eólico nos Estados Unidos; e (iv) a previsão de pixels ausentes em imagens.

Este trabalho foi organizado de maneira incremental, abordando particularidades das diferentes aplicações descritas. Concretamente, primeiro abordamos problemas de previsão de dados temporais; em seguida, previsão de dados espaciais; e finalmente, concentrámo-nos na previsão para dados espácio-temporais.

Na previsão de dados temporais propusemos uma nova classe de tarefas, denominada "2D-interval predictions", que consiste na tentativa de obter a previsão de um intervalo plausível de valores para uma janela temporal futura. Foi formalizada a tarefa, proposta uma solução para a mesma, e estabelecida uma maneira de avaliar os modelos para essa tarefa. Realizámos um extenso conjunto de testes experimentais, que mostram uma clara vantagem da nossa abordagem a este tipo de tarefas.

Em relação aos dados espaciais, propusemos uma solução para o problema de interpolação espacial através de uma nova metodologia baseada em duas idéias principais: (i) transformar o problema numa tarefa de regressão e (ii) descrever a dinâmica espacial dos dados através de indicadores espaciais. Esta metodologia diferencia-se do estado da arte, na medida em que permite a utilização de dados de regiões afastadas na previsão, contradizendo, dessa forma, a primeira lei da geografia. Nós testámos extensivamente esta metodologia em problemas de preenchimento de pixels ausentes em fotografias. Os resultados obtidos mostram uma clara vantagem da nossa abordagem, quando comparada com o estado da arte em interpolação espacial.

Finalmente, propusemos uma nova técnica que tem como objetivo melhorar a precisão da previsão em dados espácio-temporais. A técnica proposta difere das abordagens mais comuns, pois utiliza as características espácio-temporais dos dados para melhorar a precisão da previsão. Nomeadamente, propusemos uma série de indicadores espácio-temporais, cujo objetivo é descrever a dinâmica espácio-temporal dos dados para cada região. Testámos extensivamente o método proposto em dados reais de velocidade do vento. Observou-se uma clara vantagem da nossa proposta quando comparada com métodos alternativos que podem ser aplicados a este problema.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The volume of data that is generated every day is growing exponentially. More `data` (photos, videos, web traffic, etc.) are produced than ever before. The amount of data stored in the world doubles every 20 months [Witten and Frank, 2005]. This huge volume of data creates several challenges in terms of methods for the collection, storage and analysis of the data.

Data analysis, also known as `data mining` or `knowledge discovery in databases`, is the process of extracting useful information from data. Data mining is at the intersection of several research fields - statistics, artificial intelligence, computer science, machine learning, data visualization and databases. According to Witten and Frank [2005] "data mining is the extraction of implicit, previously unknown, and potentially useful information from data"; Rokach and Maimon [2008] refer that "the science and technology of exploring data in order to discover previously unknown patterns, is a part of the overall process of knowledge discovery in databases"; whilst for Torgo [2010], "data mining has to do with the discovery of useful, valid, unexpected, and understandable knowledge from data".

Kantardzic [2011] defines two general classes of data mining problems: description and prediction. Description has the goal of providing better insights on the data, by finding hidden patterns that describe interesting properties of the data. Prediction has two general goals: (i) forecasting the expected value of a variable of interest given a set of explanatory variables (predictors); and (ii) providing a better understanding of the relationship between the predictors and the target variable. Several data mining techniques were developed to

solve these two data mining problems. Fayyad et al. [1996] classified these data mining techniques in six main tasks: classification, regression, clustering, summarization, change and deviation detection, and dependency modeling.

Classification - discover a predictive function that can be used to forecast to which class (set of categories) a new observation belongs to based on a set of values of other variables (the predictors);

Regression - differs from classification on the type of variable of interest, which in regression is continuous;

Clustering - find the "natural" groupings (clusters) of the cases in the data;

Summarization - is used to describe key properties of a data set (e.g. variability, centrality, etc.);

Change and deviation detection - identify when significant changes in behavior have occured in the data;

Dependency modeling - search for dependencies between variables, identifying correlation between items (variables) in the data.

These and other data mining techniques are being applied to solve several real world problems: from spam classification [Benevenuto et al., 2010; Boykin and Roychowdhury, 2005; Drucker et al., 1999] to financial markets [Kim, 2003; Tay and Cao, 2001]. Several new data mining challenges are emerging, with new types of sensors (cheaper and more advanced) collecting more data and new domains being monitored.

In this thesis we focused on applying data mining techniques to solve real world problems described by data that has a temporal, spatial or spatio-temporal nature. The main particularity of these types of data sets is the fact that each data point is index by time, space or space-time. Forecasting methods for these types of data need to consider the temporal, spatial or spatio-temporal correlation between data points. Spatio-temporal forecasting methods, in particular, have received little attention from the scientific community. Spatio-temporal applications have as distinguishing characteristic the fact that it is expected that there is some form of unknown spatio-temporal correlation between data points and this fact should not be ignored by any method that is applied to these problems.

1.1 Problem Definition and Motivation

The work carried out in this thesis was driven by the requirements of several real world applications, namely:

Monitoring Water Quality Parameters: the company (AdDP - Águas do Douro e Paiva) managing the water distribution network of the city of Porto in Portugal, must have a tight control over several water quality parameters. Ensuring the quality of the water distributed to the population of the second largest city of Portugal is of key importance. For each water quality parameter the company has to guarantee that the values are acceptable, and inside the legal limits imposed by the government. If the company fails, severe penalties are applied.

Water Consumption Prediction: the water distribution network company of a city in south-eastern Spain must supply clean water at the right pressure to the consumers. To ensure that supply and demand are balanced, the company must known in advance the expected consumption. With that information the company can be more prepared to possible vulnerabilities in the system and can plan the best pumping scheme to minimize its costs.

Fill in Missing Pixels: automatically fill in the missing pixels in a image is important in many domains: surveillance, security, restoration, etc. An image is composed by several pixels indexed in a cartesian system (coordinates x and y). Several factors may lead to absence of information on some of these pixels. Filling in these missing pixels is an important task that tries to predict unknown pixels at several locations based on the known pixels of the image.

Wind Speed Forecasting: the wind farm in the eastern region of the US needs accurate predictions of the expected future wind speed. That information is crucial to negotiate in the electricity market. The electricity market is similar to an auction, where the participants buy and sell energy. The expected future power production is crucial to define the best bidding strategy to maximize the profit and avoids any penalties from missing delivering energy.

In the first application the AdDP company has to maintain a tight control of several water quality parameters, on the entire distribution network throughout the year. This necessity derives from the fact that there are legal limits that must not be crossed. As a cautious measure the AdDP company created internal limits that are tighter than the legal ones and serve as internal alarms that lead to inspection activities. These activities have costs to the company and thus setting these internal limits is a big challenge for the company because of the trade-off between these costs and the fines that need to be payed if the legal limits are crossed. If the limits are too narrow, too many false alarms are generated, while if they are nearer the legal limits, when alarms are generated it may be too late for a proper corrective action to be taken. This task can be classified as prediction problem, more specifically forecasting an interval where the values of certain water quality parameters are expected to be. According to the AdDP company this intervals of "normality" of the values will vary along the year as the parameters are influenced by several external factors to the water distribution network. Still, these changes are not on a daily basis and thus for the company it is important to have a forecast of the expected interval of values of each parameter for a certain future time window (e.g. the next month). Such intervals can then be used to drive their alarm generation (and thus inspection activities), with the goal of minimizing their operational costs without incurring the risk of paying heavy fines due to breaking the legal limits. In summary, the problem we are facing in this concrete application is that of providing a forecast of an interval of values for a certain future time interval.

A similar problem is faced in the second application - Water Consumption Forecasting. More important than having the prediction of the consumption distribution for a single point in the future, is to have the prediction of the consumption distribution for a certain future time interval, e.g. with high confidence the water consumption will be inside the interval $[X, Y]$ in the next 24 hours. The reason here is related with production planning. It is very bad for a water distribution company that they are not able to satisfy demand, so having the plausible distribution of the consumption values in a certain future time window is very useful as it allows for an timely preparation in terms of making sure this demand will be satisfied.

The third application consists in trying to fill in the missing pixels of an image using neighboring pixels. An image can be treated as spatial domain considering each pixel

location as a position in a cartesian system. Filling in the missing pixels of an image can be seen as a spatial interpolation problem.

The fourth application has to do with the electricity markets and wind power generation. These applications require accurate short term wind speed predictions. However, this problem differs from traditional time series forecasting tasks because wind farms are spread in space and thus the data that is collected is typically indexed in both time and space. The way wind travels provides clear evidence that there should exist some form of spatial correlation among the data that is collected, on top of the obvious temporal correlation. In this context, the problem we are facing in this application can be classified as spatio-temporal prediction task.

1.2 The Thesis Hypothesis and Main Contributions

The main connecting point between the different applications and problems tackled in this thesis is the fact that the observations in the used data sets are not independent. In all problems we address there is some form of correlation between the individual observations. This correlation is either temporal, spatial or spatio-temporal. Moreover, all the problems we tackle are numeric forecasting problems. Numeric predictions tasks are usually addressed using multiple regression approaches. However, standard regression techniques do not cope with the correlation among data observations.

In this context, the main driving hypothesis of the work in this thesis is that *it is possible to use existing out-of-the-box data mining modeling techniques to solve problems involving spatio-temporal data, provided carefully selected data pre-processing steps are carried out with the goal of providing the models with information on the spatio-temporal correlation between the data points.*

The main advantage and motivation for this approach lies on the fact that if this hypothesis holds as true we are able to apply a large, well-known and tested set of regression techniques on a range of relevant application domains. In this thesis we address a set of real world applications using this approach. Our goals are: (i) to develop pre-processing methods that allow incorporating relevant information on the spatio-temporal properties of the data, and (ii) to show that with the resulting data sets and standard out-of-the-box regression tools

we can obtain competitive performance with state of the art on these areas.

The work carried out in this thesis as lead to the following main contributions that can be categorized in three main topics:

Temporal prediction: we propose a new time series forecasting task. The two most common types of forecasting tasks in time series are point prediction and interval forecasting. Point prediction is by far the most frequent, and consists on the prediction of the future value of a time series variable for a given forecasting horizon (next h steps). The main limitation of point prediction is solved by interval forecast. Point prediction gives no information about the future variability of the prediction. Interval forecasting predicts an interval of values for the given forecasting horizon, where the future value is expected to be with a certain probability. However, interval forecasting does not solve all problems related with the future values of the variable of interest. For some domains, more important than having the expected interval for a future time point, is to have an expected interval for a future time interval (e.g. given a time series of the daily demand for a product, the production department may be interested to know that the value of the future demand for the product is expected to be between a and b in the next 15 days). We call this task "2D-Interval Prediction". We formalize this new forecasting task, propose a solution to it and describe means of evaluating solutions for these problems.

Spatial prediction: we proposed a new technique for spatial interpolation problems. Spatial interpolation is the process of filling the values of a variable at unsampled locations based on sampled ones. The research on spatial interpolation is based in some variation of the first law of geography that states that "everything is related to everything else, but near things are more related than distant things" [Tobler, 1970]. These techniques limit the influence of the sampled locations based on the neighborhood distance, and do not consider the values from far away regions. Our approach extends this notion by allowing the use of both types of data, the nearby values and values from far away regions. Our extensive set of experiments show that our proposed technique outperforms the compared techniques representing the state of the art on spatial interpolation.

Spatio-temporal prediction: the vast majority of the research carried out in spatio-

temporal applications generated by sensor networks consider only one dimension of the problem, usually the temporal dimension. In our opinion the use of only one dimension significantly limits the correct understanding of the problem. We proposed a new technique that is able to embed both dimensions, spatial and temporal. We carried out an extensive set of experiments using real data from wind farms in the US. The results of these experiments have show the advantages of our proposal.

1.3 Organization of the thesis

The organization of this thesis is driven by the three main contributions described in the previous section. The work carried out during the thesis is presented in five chapters:

Chapter 1 - Introduction

In this chapter, we contextualize the work developed in this thesis within the data mining research field. We also describe the organization, motivation and main contributions of this work.

Chapter 2 - 2D-Interval Predictions for Time Series

In the next chapter, we describe the work developed in the context of the analysis of temporal data. We propose a new data mining task motivated by a real world application. We also propose a new technique to solve these tasks and error metrics to evaluate approaches to these problems.

Chapter 3 - A Multiple Regression Approach for Spatial Interpolation

In Chapter 3, we present a new approach to the spatial interpolation problem. The proposed technique outperformed the state of the art in spatial imputation of missing pixels in photos.

Chapter 4 - Sensor Network Prediction through Spatio-Temporal Indicators

In Chapter 4, we describe a new technique to improve the prediction in spatio-temporal applications, that embeds both the spatial and the temporal characteristics of the data. We test our hypothesis using real world wind speed data measured on wind farms in US.

Chapter 5 - Conclusions and Future Directions

Chapter 5 summarizes the conclusions of this thesis and presents possible future research directions.

1.4 Publications

The work presented in this thesis was published at several international research conferences, namely:

Chapter 2

- Ohashi, O., Torgo, L., and Ribeiro, R. P. (2010). Interval forecast of water quality parameters. In 19th European Conference on Artificial Intelligence - ECAI'2010, pages 283-288. IOS Press.

- Torgo, L. and Ohashi, O. (2011). 2D-Interval Predictions for Time Series. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 787–794. ACM.

Chapter 3

- Ohashi, O. and Torgo, L. (2012). Spatial Interpolation using Multiple Regression. In 12th IEEE International Conference on Data Mining - ICDM'2012, pages 1044-1049. IEEE Computer Society.

Chapter 4

- Ohashi, O. and Torgo, L. (2012). Wind speed forecasting using spatio-temporal indicators. In 20th European Conference on Artificial Intelligence - ECAI'2012, pages 975-980. IOS Press.

Chapter 2

2D-Interval Predictions for Time Series

This chapter presents a new class of time series forecasting task - predicting the range of plausible values of a time series variable within a future time interval. The definition of this task was driven by a particular real world application - anticipating whether certain water quality parameters will be within certain acceptable ranges of values in the near future. We define this new class of forecasting tasks and describe a series of possible approaches to solve it. Finally, we carry out an extensive set of experiments that show that our proposals outperform the alternatives that were considered.

2.1 Introduction

The study of time series started in the 1800's, with the publication of Lexis "On the theory of the stability of statistical series" in 1879. This paper is the first study on the field of time series analysis [O'Connor and Robertson, 2012]. Other publication from the same period is the publication of Thiele in 1880, which applies time series techniques in astronomical geodesy, defining the distance between Copenhagen and Lund in Sweden [Lauritzen, 1981].

The notion of order among data observations distinguishes time series analysis from others. In time series the order (the time) that an event occurs is crucial for the analysis. A time series is a sequence of measurements of the variable of interest, usually collected at regular

time intervals,

$$X = \{x_t, x_{t+1}, x_{t+2}, \ldots \mid t \in \mathbb{N}\} \tag{2.1}$$

where x_t is the value of a variable measured at time t

The main goal of time series analysis is to identify patterns on the series. These patterns can be used to explain some phenomenon underlying the time series measurements or to anticipate future values of the series, i.e. forecasting. The work described in the chapter is focused on forecasting.

In Section 2.2 we formalize the task of time series forecasting. Section 2.3 describes the existing types of time series forecasting techniques. In Section 2.4 we present a new type of time series forecasting task, we describe some possible approaches to solve this task, and we propose a new approach as well as metrics to evaluate approaches to this task. In Section 2.5 we describe the experimental methodology used to extensively test our proposal in three different problems (two real world applications). We end this chapter in Section 2.6 with the conclusions.

2.2 Time Series Forecasting

Time series analysis is a solid research area with a vast number of contributions in many sub-fields: indexing, classification, clustering, segmentation and forecasting. In this thesis we are interested in the latter. Time series forecasting is a key modeling task with direct applicability in many real world domains: forecasting the future demand of electricity [Nogales et al., 2002], water consumption [Hipel and McLeod, 1994], wind speed [Brown et al., 1984], internet traffic [Basu et al., 1996], tourism demand [Lim and McAleer, 2002], the future value of stocks [Akgiray, 1989], etc. Time series forecasting is based on the assumption that the next observations of a series have some form of dependency on the previous values, and moreover that these dependency patterns repeat over time.

Time series forecasting is one of the most common forms of time series analysis and extremely used in several domains. According to Chatfield [2001] time series forecasting can be classified into three main groups: judgemental forecasters, univariate and multivariate

methods. Judgemental forecasting is a subjective technique, based on experts knowledge and/or intuition on the domain of interest. The general idea is to extract the experts knowledge. The most famous judgemental method is the *Delphi technique*, that proposes to identify a consensus between experts applying questionnaires [Chatfield, 2001]. Univariate methods forecast the expected future value of the time series looking only to the historical information of the series. The main motivation behind these techniques is that by looking at the past values of the series the model will be able to identify behavior patterns and use these patterns to forecast the expected future value of the series. Multivariate methods extend univariate methods by adding more variables in the analysis. Instead of just looking at the historical information of the target series, these techniques also look at other variables that are relevant for the domain. The key idea is to extend the search for patterns with one or more variables to improve the prediction of the variable of interest. These variables are usually named "predictor variables". In this thesis we focus on univariate methods.

The goal of time series forecasting is to discover the best patterns to describe the series. Chatfield [2001] classifies the behavior patterns of a time series in four groups:

Seasonal Variation - is a cyclic variation that occurs at regular periods of time, a year or less: quarterly, monthly or weekly;

Trend - this type of variation consists on a upward or downward behavior of the series;

Other cyclic variation - cyclic variations occurring at regular periods not related with the calendar, e.g.: business cycles or biological behaviors of living creatures;

Irregular fluctuation - random behavior that is not characterized by any of the previous variations.

The main goal of time series forecasting models is to be able to accurately forecast the future expected value of the series. In the analysis of a series X at the instant of time t (represented by x_t), the forecasting model makes a prediction of the expected future value x_{t+h}, where h is the forecast horizon (c.f. Equation 2.1). The difficulty of this task depends on the domain and on the type of patterns exhibited by the time series. One example of an extremely difficult domain are time series from financial markets [Cowles, 1933; Dokko

and Edelstein, 1989]. One famous quote from Box et al. [1976] - "All models are wrong but some are useful", describes the difficulty of modeling time series forecasters.

Box et al. [1976] in the 1970's presented what is considered one of the most traditional time series forecasting model, the Autoregressive Integrated Moving Average (ARIMA) or Box and Jenkins model. This work was pioneer in time series analysis, and served as the foundation to a vast number of scientific contributions in several domains, such as for instance: forecast the next day electricity prices [Contreras et al., 2003], forecasting the day-ahead wind speed [Kavasseri and Seetharaman, 2009], forecasting tourism travel demand for Australia [Lim and McAleer, 2002], forecasting network traffic [Zhou et al., 2005], etc.

Time series forecasting is not limited to the Box and Jenkins model and/or its variations. One of the most recent trends in time series forecasting is to apply machine learning techniques, or a combination of models to solve these tasks, e.g. Lu et al. [2009] applied support vector regression to forecast financial time series, Khashei et al. [2008] proposed a hybrid system using neural networks and fuzzy systems to forecast financial markets, Sapankevych and Sankar [2009] described several implementations of support vector machines applied in time series forecasting, Khashei and Bijari [2011] combined neural networks with ARIMA to forecast Canadian Lynx time series data, etc. Time series forecasting is a vast research area with many contributions, however the majority are focused on the analysis of point prediction, i.e. in forecasting the expected value of the series for a certain forecasting horizon (future time interval). In this chapter we will describe a novel and different type of forecasting task.

2.3 Types of Time Series Predictions

Chatfield [2001] classifies the univariate time series forecasting tasks in two main types: point predictors and interval forecasters.

2.3.1 Point Prediction

The most common form of prediction in times series analysis is point prediction. The goal of point prediction is to forecast the expected value of the series for a specific future instant of time; at the time instant t forecast the expected future value of the series for the instant of time $t+h$ (where h is the forecasting horizon). The most common approach (univariate time series models) uses the previous historical values of the series to obtain the models. At time t the previous $[t-n, t]$ values are then used as the quantity of information necessary to obtain a forecast for $t+h$ (c.f. Figure 2.1), i.e. the models make an assumption on the size of the past window of influence on the future values of the series. For instance, in a context of stock market forecasting, we could be interested in predicting the expected next day closing price of some stock to assist the decision concerning possible trading actions: sell, buy or keep the stock. The forecaster "A", using some previous historical information (previous n days) of the stock, makes a prediction of the expected next day stock price ($h = 1$ day). That information may help the analyst in the decision making process.

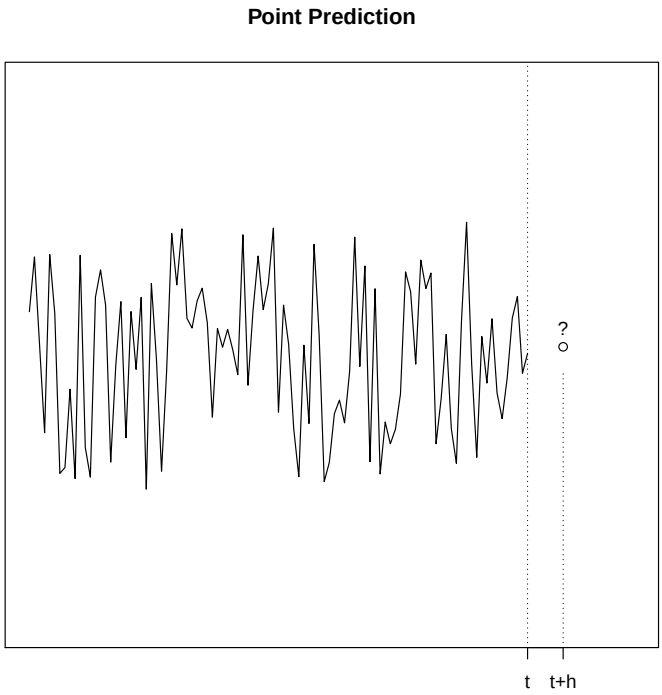


Figure 2.1: Point Prediction.

There is a vast amount of scientific contributions on point predictions in time series analysis.

We highlight some recent works in this research field. [Jain and Kumar, 2007] and Zhang [2003] proposed an hybrid approach using an ARIMA model and neural networks to forecast time series. This hybrid approach was used to forecast the next month streamflow data at Colorado river, USA. On the other hand Zhang [2003] tested the approach in three different datasets with three different forecasting horizons - Wolf's sunspot, Canadian Lynx and British pound US dollar exchange rate, with the forecasting horizons of 1, 6 and 12 months. Kim [2003] compared support vector machines and neural networks in the task of forecasting the next day return of the S&P 500 stock market index. Lu et al. [2009] proposed a support vector regression model to detect and remove noise from financial time series data - the Nikkei 225 opening cash index and TAIEX (Taiwan Stock Exchange Capitalization Weighted Stock Index) closing cash index. Lee and Tong [2011] and Shi et al. [2012] compared a combination of models in the task of forecasting time series. Lee and Tong [2011] compared a hybrid approach of ARIMA and genetic programming against two commonly used approaches ARIMA + neural networks and ARIMA + support vector machines. This work was carried out using two different time series datasets, the Canadian Lynx from 1821 to 1934 and the energy consumption, in China from 1957 to 2007. Shi et al. [2012] compared ARIMA + neural networks, ARIMA + support vector machines, ARIMA, neural networks and support vector machines in the task of forecasting multi-steps ahead (1, 3, 5, 7 and 9 hours) wind speed and power generation, using data from Colorado, USA, ranging from 2005 to 2007. Weron and Misiorek [2008] compared parametric and semi-parametric models in the task of forecasting 1-day ahead spot price in auction type electricity markets, in the California and Nord Pool markets. Bermolen and Rossi [2009] compared support vector regression against moving averages (MA) and auto-regressive models (AR) in the task of forecasting the load in network links. Matteson et al. [2011] analyzed Toronto's emergency calls (from january 2007 to december 2008), and proposed a forecaster based on GARCH models to predict the calls arrive volume. Assaad et al. [2008] proposed a boosting algorithm with recurrent neural networks to improve time series prediction in single step and multi steps ahead predictions, using a dataset concerning sunspots that describes the number of dark spots in the sun from 1700 to 1979.

As we have mentioned before, point prediction is the task that is most frequently addressed in time series forecasting research. However, point prediction has serious limitations for some types of real world problems. Namely, there are domains where on top of a particular

predicted value, it is of key relevance to have a confidence interval around this value, i.e. to forecast an interval where there is a high probability the future time series value will be.

2.3.2 Interval Forecasting

Interval forecasting, also known as, confidence intervals or confidence bounds, focus on the key limitation of point prediction, the variability/uncertainty around the point prediction. Point prediction, as the name suggest, produces a single value for the given forecasting horizon, giving no information about the uncertainty or variability of that prediction. Interval forecasting, instead of predicting a single value in the future produces an interval supported by a probability. This interval is defined by two values, lower and upper limits where the expected value is suppose to be in. Chatfield [2001] defines interval forecasts as "an upper and lower limit between which a future value is expected to lie with a prescribed probability". The goal of the interval forecasting is thus to obtain a prediction at time t of the interval where the expected value of the series for the future time $t + h$ (where h is the forecasting horizon) is supposed to lie with a certain probability. In univariate analysis the forecaster uses only the previous historical information of the time series $[t - n, t]$, where n is the size of the historical window used (c.f. Figure 2.2). For instance, in the area of manufacturing, a point forecaster would make predictions of the expected future consumption, whilst an interval forecaster would make predictions of the probable interval (upper and lower limits), where this expected future consumption is supposed to be in, with some probability.

The number of scientific contributions on interval forecasting is smaller when compared to point prediction. Still, there are several significative contributions for a considerable number of application domains. Khosravi et al. [2011] and Mazloumi et al. [2011] applied interval prediction in the analysis of bus travel time duration in Australia. Khosravi et al. [2011] applied neural networks to prediction of the bus travel time and applied two forecasting techniques, "the delta" and "Bayesian" in the prediction of the intervals. This methodology was applied in two datasets, six months of bus routes in Melbourne Australia and 95 days bus routes in the Netherlands. Mazloumi et al. [2011] also applied neural networks to forecast the bus travel time in Melbourne. Wu et al. [2006] proposed a quantile

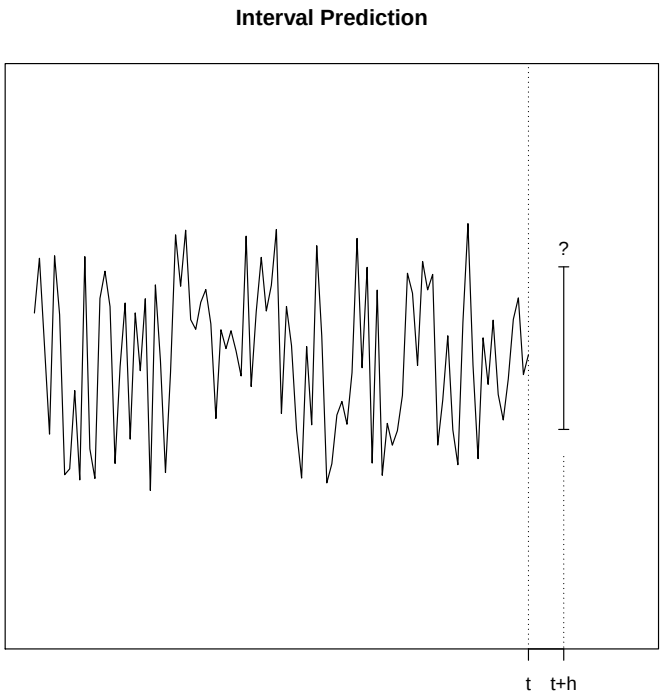


Figure 2.2: Interval Prediction.

regression approach to forecast the interval of transactions per hour in a network traffic domain. Nielsen et al. [2006] applied quantile regression, more specifically the quartiles Q_1 and Q_3 to forecast an interval of wind power production in the Nordpool market. Pinson and Kariniotakis [2010] also worked with the domain of wind power generation in Denmark (from March 2001 to May 2003). However, a different approach was used, as the intervals were calculated based on the past errors made by the point predictors. Isengildina-Massa et al. [2008] used quantile regression to forecast prices interval variation of corn and soybeans in the US. Within the financial market research area, Andersen et al. [1999] proposed to forecast one month ahead financial market volatility using GARCH models. Serguieva and Hunter [2004] proposed a hybrid approach applying a fuzzy method combined to neural networks in the prediction of price intervals for 35 UK companies in the London stock exchange market. Zhang and Luh [2005] applied a hybrid approach to forecast intervals of market closing prices, using neural networks combined to Kalman filter.

Density forecasting is considered an extension of interval forecasting, where the entire

probability distribution of the expected value is calculated [Chatfield, 2001]. Taylor et al. [2009a] proposed to apply ensemble models for density forecasting in the task of predicting 10 days ahead variability of wind power at five UK wind farms. Hall and Mitchell [2007] proposed an approach that combines different density forecasters to produce a more accurate density forecast (applying weights according to the accuracy of the forecasters) in the analysis one-year ahead RPIX inflation in UK.

2.4 2D-Interval Predictions

Although point prediction and interval forecasting (or density forecasting) may cover a considerable area in time series predictive modeling, for several domains these tasks are not sufficient. In these domains more important than having a prediction (point or interval) for a particular future instant of time, is to have a prediction of the expected variability/uncertainty for a future interval of time (a future time window). This means that instead of generating an interval for a single point in the future, the goal is to have an interval of values for a interval of time, e.g. for the next 30 days (assuming a daily time series). More specifically, the main goal at time t is to forecast the expected interval of values of the time series for a future time interval $[t+h, t+h+k]$, where h is the forecasting horizon and k is the size of the target time interval, c.f. Figure 2.3. We call this type of prediction tasks **2D-interval prediction**, as we have a two-dimensional (time and values) interval.

As mentioned before, existing work on time series forecasting is essentially focused on: (i) point predictions (e.g. [Chatfield, 2004]), (ii) interval predictions (e.g. Chatfield [2001]), or (iii) density forecasting [Tay and Wallis, 2000]. However, to the best of our knowledge there is no established methodology for forecasting an interval of values for an interval of time. Addressing this limitation is one of the main goals of this thesis. The lack of a methodology to handle this problem is rather surprising given the amount of relevant applications that could benefit from this type of prediction. For instance, any application requiring some form of production planning for a certain demand scenario, will find this type of prediction of high utility. This includes areas like manufacturing in general, energy production, water distribution, etc. For example, in wind power production it is important to predict the future wind variability in order to ensure that supply and

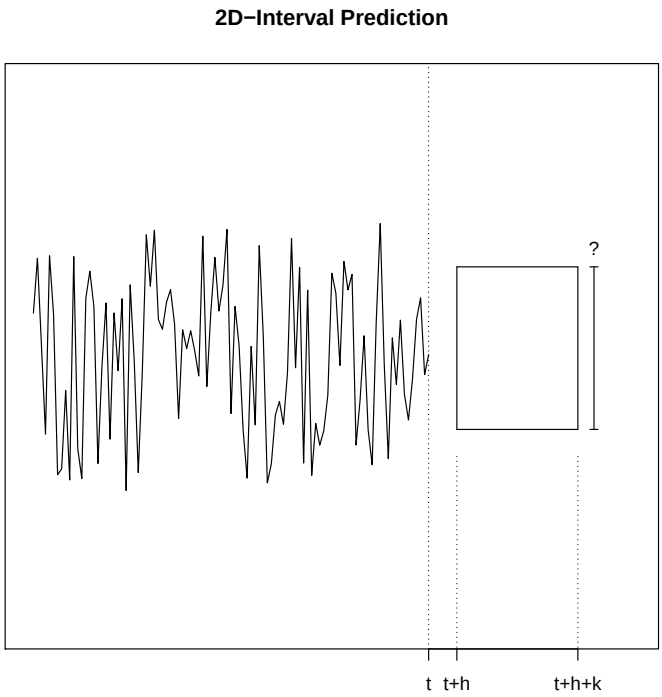


Figure 2.3: 2D-Interval Prediction.

demand are balanced [Bremnes, 2004; Taylor et al., 2009b]. In power production (electricity market [Taylor, 2006]) it is common to place bids for the future production. In this context, the prediction of the future wind variability must be accurate, to avoid penalties resulting from any deviation between production and demand. To make optimal decisions, a model that is able to predict the quantiles of the distribution gives much more information than single point predictors [Bremnes, 2004]. In inventory management, over-production may lead to inventory costs while under-production may originate unsatisfied demand and lost of profits [Chatfield, 1993]. Other relevant application areas include customer wallet estimation [Perlich et al., 2007] or computer network traffic analysis [Wu et al., 2006]. Several investment-related applications (e.g. financial markets) will also find this type of forecasts of great use.

From a theoretical perspective we are talking about forecasting the distribution of the values of the target time series for a future time interval. The key difference from existing work on interval prediction and density forecasting (e.g. [Tay and Wallis, 2000]) is the fact that we want to estimate this interval for a future interval of time and not for a single

point in time, see Figures 2.2 and 2.3. The applicability to real world scenarios motivates this difference. In this work we restrict this general scenario to the prediction of some descriptive statistics of this distribution. Namely, we will focus on forecasting a kind of interval of "normality" of the values of the series for a certain future time interval. We will represent this interval of normality by the 1st and 3rd quartiles of the variable distribution, though our proposal could be applied to any quantile.

The main contributions of the work presented in this chapter are: (i) to increase the awareness of the research community to a high-impact task that was not addressed with specific methods before; and (ii) to propose a general method for addressing this class of time series forecasting problems. In the context of the presentation of our proposal we will describe an extensive set of experiments that we have carried out to demonstrate the validity of our approach under a wide range of scenarios.

A 2D-interval was defined as an interval of values for a future interval of time (a *time window*). This interval is defined by two limits - upper and lower. The future values of the time series within the selected time interval are expected to be included in (c.f. Figure 2.3) within these limits. It is thus a form of summary statistic of the unknown future distribution of the values of the series for the target forecasting horizon. We will address this general problem by simplifying it into predicting two quantiles that will define the upper and lower boundaries of the interval, the 1st and 3rd quartiles of this unknown distribution. These two summary statistics provide an interval where a significant number of the values should be contained (assuming a near-normal distribution of the values of the target variable).

2.4.1 Possible Approaches to the Problem

As mentioned before, we are not aware of any existing approach to the previously described 2D-interval prediction problem. Nevertheless, we can try to adapt some existing time series forecasting techniques to obtain this type of forecasts.

Point prediction can be iterated to obtain predictions for more than one point in the future (known as iterated predictions e.g. [Chatfield, 2004]). At time t we obtain a prediction for time $t+1$. This prediction can be incorporated into the training data available to the model as if it was true, and a new step ahead prediction can be obtained, which in effect would

correspond to a prediction for time $t+2$. These iterations can continue until we get a point prediction for all the forecasting time interval $[t+h, t+h+k]$. Using these k predictions we can obtain the summary statistics we want to have a 2D-interval prediction. This is a simple approach that can be applied to any existing time series modeling technique. The main drawback of this strategy is that by incorporating the one-step ahead predictions into the training data we are potentially amplifying the prediction errors of the models as we move further in time. In the comparative experiments that will be described in this chapter we will call this approach the *iterated predictions* method.

A second plausible approach to the 2D-interval problem is decompose it into several prediction tasks (k-models). If we want a 2D-interval for a future time window of length k, then we can transform this into k prediction tasks with each being addressed by a different model designed to obtain a point prediction for $t+i$, where $i \in [1, k]$. Given these k models, to obtain a 2D-interval prediction at time t we use the k models predictions and calculate the necessary statistics to obtain the 2D-interval. The main drawback of this approach is the computational cost of obtaining k different models. This may be a critical issue with high-frequency data as it is often the case in time series problems. In our experiments we will call this approach the *k-models* method.

The two strategies described above, iterated predictions and k-models, can be easily used with interval forecasting models. The main difference is that instead of forecasting a single value for a time point, the models predict an interval. Then to get the 2D-interval prediction we can average the values of the limits (upper and lower) of the k predicted intervals.

In our experiments we will also consider a naive approach as a kind of baseline method. A 2D-interval forecast at time t for the interval $[t+h, t+h+k]$ will be obtained by using the summary statistics calculated with the most recent past values of the series in the interval $[t-k, t]$. This can be regarded as a kind of random walk approach to our problem and we will expect other alternatives to clearly outperform this baseline method.

2.4.2 Our Approach

The goal of 2D-interval predictions is to have an estimate of the way the values of the target time series will be distributed on a future time window. As we have mentioned before, we

will constrain this general goal to the task of obtaining some summary statistics of this future distribution of the values. More specifically, we will have as goal to estimate the future interval of "normal" values of the distribution and we will estimate this "normality" by means of two quantiles of the distribution that by definition form an envelop that includes the most frequent values of a variable that follows a near-normal distribution - more specifically an unimodal distribution.

The key idea of our proposal is that instead of obtaining these summary statistics from predictions of the values of the time series (as done in the approaches described in Section 2.4.1), we propose to directly forecast some summary statistics of the distribution of the time series values, in a future time window.

Let Q^k_α and Q^k_β be the α and β unknown quantiles of the time series values for a future time window of length k, respectively. These two values establish an interval where $|\beta-\alpha|\times 100\%$ of the series values are supposed to be included in. The interval between the 1st and 3rd quartiles ($Q_{0.25}$ and $Q_{0.75}$) will contain 50% of the values of the series. We will use the predicted values for these two distribution statistics as the source for obtaining 2D-interval predictions. This means that we will directly forecast these quantile statistics instead of calculating them from predictions of the target time series. Our proposal is thus to define the two following prediction problems:

Definition 2.1. The prediction tasks we will use to obtain 2D-interval predictions are defined by the following equation,

$$Q^k_\alpha = f(v_1, \cdots, v_a) \; and \; Q^k_\beta = f(v_1, \cdots, v_a) \tag{2.2}$$

where $v_1, \cdots, v_a$ are a set of descriptor variables, and Q^k_α (Q^k_β) are the target variables, and the α (β) are the quantiles of the time series variable for the next k time points, i.e. the estimated quantile for the time interval $[t+h, t+h+k]$, (c.f. Figure 2.3).

In order to be able to obtain models to forecast these quantiles we need to have a set of training data where these values are known. Thus for "preparing" the training sample at time t we will need information on both the selected predictor variables but also on the target time series values in the period $[t+h, t+h+k]$, where k is the time length of the target 2D-intervals.

Tables 2.1 and 2.2 present an illustrative example of the type of pre-processing we carry out. Table 2.1 shows an example time series, while in Table 2.2 we present the data set obtained to apply our proposed method. In this example we have used a future time window of size 5 ($k = 5$) and the 1st and 3rd quantiles $[Q_{.25}, Q_{.75}]$ to define the 2D-interval. The prediction tasks are represented by two functions (i.e. two target variables), one for each quantile: $Q^{k=5}_{\alpha=.25} = f(v_1, v_2, \cdots, v_p)$ and $Q^{k=5}_{\beta=.75} = f(v_1, v_2, \cdots, v_p)$. The pre-processing steps we need to carry out to obtain the data set consist of: (i) selecting a date with at least five observations in the future (in the table we illustrate with 2013-01-25); (ii) with the five observations in the future (represented in light blue) calculate the quantiles $Q_{.25}$ and $Q_{.75}$, thus obtaining the values of the target variables ($Q^5_{.25}$ and $Q^5_{.75}$, respectively); (iii) with the data in the past (represented in dark blue) calculate the predictor variables (columns $v_1, \cdots$); (iv) increment the date and repeat the process, until no more data is available.

time	$\mathcal{Y}$
2013-01-20	444.40
2013-01-21	410.03
2013-01-22	450.45
2013-01-23	400.07
2013-01-24	388.15
2013-01-25	390.89
2013-01-26	389.12
2013-01-27	413.34
2013-01-28	390.45
2013-01-29	400.07
2013-01-30	410.15
⋮	⋮

Table 2.1: Time Series

time	$Q^5_{.25}$	$Q^5_{.75}$	v_1	$\cdots$
2013-01-25	390.45	410.15	418.62	$\cdots$
⋮	⋮	⋮	⋮	$\ddots$

Table 2.2: Regression Data

Using this type of pre-processing process we obtain a data set that can be used to obtain models to address the two predictive tasks mentioned in Equation 2.2.

The methods presented in Section 2.4.1 are all based on predicting the values of the target time series for a future time interval. This is done either by iterated predictions or by obtaining k different models. Either way it is based on these predictions that we calculate the summary statistics to forecast the 2D-intervals. In our approach we directly predict these statistics. The main motivation for our proposal is that forecasting these statistics

directly is an easier prediction task as quantile statistics have a distribution that is smoother than the original variables from which they are calculated. In effect, quantiles are known to be robust to a few extreme values, thus smoothing-out these variations on the original time series. This means that the distribution of the quantile variables that we use as targets is clearly more "well behaved" than the distribution of the original series. In this context, we expect this prediction task to be easier to solve than the original of forecasting the time series values for a future time window. Our hypothesis is that by solving an easier prediction task we will be able to have more accurate predictions of the 2D-intervals. The experiments described in Section 2.5.1 were designed to test this hypothesis.

The choice of the most appropriated predictor variables $v_1, \cdots, v_a$ is not part of our proposal. They should be selected with the goal of optimizing the performance of the selected modeling tool in the task of forecasting the quantiles. This decision is not different from what is necessary on any other time series prediction task - it is a standard feature selection problem for which many existing approaches exist [Guyon and Elisseeff, 2003; Kira and Rendell, 1992]. Our proposal only changes the target variable. In our experiments we will include as predictors recent past values of the time series and also past values of some distribution statistics as we will see in Section 2.5.1.

2.5 Experimental Evaluation

In this section we evaluate our proposed method for obtaining 2D-interval predictions under a large set of experimental setups. In the previous sections we have formalized the 2D-interval prediction task, have presented possible approaches to this task using existing time series forecasting methodologies, and have proposed an alternative approach to this task. Our hypothesis is that our proposal is an easier forecasting task and thus the resulting 2D-interval predictions will be more accurate. The main goal of the experiments we describe in this section is to check the validity of this hypothesis.

We will test our hypothesis under different experimental conditions. Namely, we will try: i) different time series data sets; ii) different forecasting models and variants within these models; and iii) different evaluation metrics. The overall goal is to test the hypothesis under a diverse and large set of experimental settings. Obviously, these settings are far

from exhaustive but we are convinced that they are a good sample of real world prediction settings.

2.5.1 Experimental Methodology

At the core of any experimental comparison is the method used to obtain reliable estimates of the selected evaluation metrics. Our data set contains time series which means that there is a implicit order (time) among different observations in the data set. This fact precludes the use of any standard experimental methodology based on resampling strategies that involve changing the order among data points. In this context, we have used as experimental methodology for the three prediction problems we will consider (artificial data, water quality parameters and water consumption) a Monte Carlo simulation [Torgo, 2010]. Namely, we have randomly selected 10 time points within the available periods of time for each data set. For each of these 10 points we have used the previous m values of the series to obtain the models that were then used to make 2D-interval predictions for the next n points using a sliding window approach. The values of m and n are domain dependent. At each of these n prediction points the goal was to estimate the 2D-interval for a future time window of k time points. We will try more than one value of k. The results we present for each k value (the time length of the 2D-intervals) are averages over these test sets with n points, on the 10 Monte Carlo repetitions (i.e. at randomly selected points of the full time series).

For instance, for the water quality parameters at time t we use 365 days of training data to obtain a model that is used to forecast a 2D-interval ($[Q_{0.25}, Q_{0.75}]$) for the next 30 days, i.e. for the time interval $[t+1, t+1+30]$. After this prediction is made, the training window is slided one day forward and another model is obtained. This model, obtained at $t+1$, is again used to obtain a 2D-interval prediction for the next 30 days, i.e. a 2D-interval prediction for the time interval $[t+2, t+2+30]$. This sliding window process is repeated until we have made predictions at all time points in the next 90 days, the size of the test window. All model variants are evaluated using the same data.

The goal for all alternatives is to obtain an estimate of the 1st and 3rd quartiles for the target time interval k, i.e. a 2D-interval prediction for a future time window of length k. However, depending on the approach followed, these estimates are obtained using a

different method. This means that the target variable(s) will always be the same - $Q^k_{0.25,t}$ and $Q^k_{0.75,t}$; although the method used to reach predictions for these two targets will be different depending on the used approach. As we have seen (c.f. Section 2.4.1) some methods obtain these predictions by first forecasting the values of the time series, while our approach predicts these statistics directly.

Independently of the methodology used to obtain the predictions for the target variables, all compared approaches will use the the same predictor variables that were selected with the goal of trying to provide useful information on the recent dynamics of the time series and also past values of k-length descriptive statistics. Specifically, in our experiments we will obtain alternative models whose goal is to solve prediction tasks described by the following general equation,

$$TGT = f(Y_{t-1}, \cdots, Y_{t-p}, Q^{-k}_{\alpha,t}, Q^{-k}_{\beta,t}, \bar{Y}^{-k}, \sigma^{-k}_{Y}) \tag{2.3}$$

where $Q^{-k}_{\alpha,t}$ is the value of the α quantile calculated using the past k values of the series at time t, $\bar{Y}^{-k}$ is the average time series value on the same past window, σ^{-k}_{Y} the respective standard deviation, and $Y_{t-1}, \cdots, Y_{t-p}$ are the last p values of the series.

The target variable (TGT) will be different depending on the approach. For the iterated point predictions this will be the next value of the series, Y_{t+1}. For the iterated interval prediction will be $Q^{t+1}_{\alpha,t}$ ($Q^{t+1}_{\beta,t}$). For the k-models the targets will be Y_{t+i} where $i \in [1, k]$ for each of the k models for point prediction, and $Q^{i}_{\alpha,t}$ ($Q^{i}_{\beta,t}$) for interval prediction. For our proposal there will be two models, one predicting the 1st quartile on the 2D-interval, $Q^k_{0.25,t}$, and the other the 3rd quartile, $Q^k_{0.75,t}$.

To ensure a fair assessment of the hypothesis driving our experiments all alternatives that will be compared will be given exactly the same data in the context of the Monte Carlo simulation, with the exception of the target variable.

All data, code and extra results are provided in a web page [1] to ensure that our work is replicable.

[1] `http://goo.gl/hRBMd`

2.5.2 Models

For each of the considered prediction tasks we have tried a wide range of modeling approaches to test our hypothesis. The idea is to confirm its validity independently of the technique used to forecast the Q_γ^k quantiles. All the used tools are freely available in the R software environment [R Development Core Team, 2010], which ensures easy replication of our work. The following is a list of the methods used in our experiments and the variants of these models that were considered:

Random Walk (RW) - a simple baseline method that uses the quantiles estimated with the last k time series values as predictions for the quantiles of the next k time points;

Regression Trees (RT) - a regression tree (e.g. [Breiman, 1984]) based on the R package `rpart` [Therneau and port by B. Ripley., 2009]. In our experiments we have used an interface to the `rpart` function provided in package `DMwR` [Torgo, 2010] and have tried 4 different variants by using the parameter `se` that controls the level of pruning with values $0, 0.5, 1$ and 1.5.

Support Vector Machines (SVM) - an implementation of SVMs (e.g. [Cristianini and Shawe-Taylor, 2000]) available in the R package `e1071` [Dimitriadou et al., 2009]. We have tried 20 variants by using the parameter *cost* that represents the penalty associated with errors, with the values $1, 5, 10, 50, 100$ and the parameter *gamma* which the used radial based kernel, with the values $0.001, 0.01, 0.05$ and 0.1.

Random Forest (RF) - an implementation of random forests [Breiman, 2001] available in the R package `randomForest` [Liaw and Wiener, 2002]. We have used 3 variants of the parameter *ntree* that controls the number of trees in the forest (ensemble), with the values $500, 1000$ and 1500.

Quantile Regression Forests (QRF) - a random forest variant [Meinshausen, 2006] designed to optimize the prediction of quantiles (interval forecasting). We have used the implementation of these models available in the R package `quantregForest` [Meinshausen, 2007]. We have tried 3 variants of the parameter *ntree* that controls the number of trees in the forest (ensemble), with the values $500, 1000$ and 1500.

2.5.3 Evaluation Metrics

There is an extensive literature on evaluation metrics for single point prediction models. Most measures compare the true and predicted values and eventually contrast the performance of the model being evaluated against some baseline. Our prediction task is different as we have mentioned before. We are addressing the prediction of a 2D-interval by means of the 1st and 3rd quartiles, which means there are some similarities with the goals of quantile regression [Koenker, 2005]. However, in quantile regression the goal is to obtain point predictions of the quantiles. The evaluation of quantile regression models is usually carried out with the help of Equation 2.4. It can be easily shown that the value of $L_\alpha(y, \hat{y})$ is optimized by predicting the quantile Q_α (i.e. $\hat{y} = Q_\alpha$).

$$L_\alpha(y, \hat{y}) = \begin{cases} \alpha \cdot (y - \hat{y}) & \text{if} y \geq \hat{y} \\ (1 - \alpha) \cdot (\hat{y} - y) & \text{otherwise} \end{cases} \tag{2.4}$$

Table 2.3 presents an illustrative example of how $L_\alpha(y, \hat{y})$ is calculated. In the first column we have the true values of the series (y), in the second column we have the predicted values ($\hat{y}$), and in the third and fourth columns we have the loss functions $L_{0.25}$ and $L_{0.75}$ calculated using Equation 2.4 for the quantiles $Q_{0.25}$ and $Q_{0.75}$, respectively.

y	$\hat{y}$	$L_{0.25}$	$L_{0.75}$
2	2	0	0
2	3	0.75	0.25
3	2	0.25	0.75
4	2	0.5	1.5
2	4	1.5	0.5

Table 2.3: An illustrative example of calculating the $L_\alpha(y, \hat{y})$.

In this context, if we want to estimate the values of $Q_{0.25}$ and $Q_{0.75}$ for a certain period of time k, we can evaluate the predictions of our models ($\hat{Q}^k_{0.25}$ and $\hat{Q}^k_{0.75}$) using Equation 2.4, given the true target variable values $y_{t+1}, \cdots, y_{t+k}$. Moreover, if we are given a test set we can calculate the total quantile error (TQE) of our 2D-interval predictions as follows,

$$TQE = \sum_{i=1}^{n} \left[\sum_{j=i}^{i+k} L_{0.25}(y_j, \hat{Q}^k_{0.25,i}) + \sum_{j=i}^{i+k} L_{0.75}(y_j, \hat{Q}^k_{0.75,i}) \right] \tag{2.5}$$

where $\hat{Q}^k_{\alpha,t}$ is the α quantile prediction for the future k-length interval starting at time t.

We have also compared our alternative models using the mean absolute quantile (MAQ) deviation of the model predictions, i.e.

$$MAQ = \frac{1}{2n}\left[\sum_{i=1}^{n} |Q^k_{0.25,i} - \hat{Q}^k_{0.25,i}| + |Q^k_{0.75,i} - \hat{Q}^k_{0.75,i}|\right] \tag{2.6}$$

This evaluation metric can be easily obtained by calculating the observed 2D-interval quantiles and comparing these with the predictions of our models. This statistic will measure the average absolute error of our quantile predictions when compared to the true observed quantiles.

The real world applications that we target with this proposal of 2D-interval predictions have some particularities that are not completely captured by the statistics of Equations 2.5 and 2.6. Namely, there are usually costs and benefits associated with the predictions of the models. In effect, the predicted intervals are frequently used to carry out some actions (e.g. production planning) that may result in costs or benefits depending on the accuracy of the predictions. We will also evaluate the 2D-interval predictions taking these factors into account. The predicted intervals divide the values in three classes: unusually high or low values, and normal values that are within the interval limits. This means that given the predictions $\hat{Q}^k_{0.25}$ and $\hat{Q}^k_{0.75}$, we can discretize the series values into these three classes. Accordingly, we can look at the observed values in the k-length interval and calculate the true observed $Q^k_{0.25}$ and $Q^k_{0.75}$ values. Using these values we can calculate the true class labels of each value in the k period. Elkan [2001] has established the theoretical grounds for cost-sensitive learning. The proposed framework is based on the concept of benefit matrix. This matrix sets the benefits of all accurate predictions, as well as the costs (negative benefits) of the errors. We will also use this setup to evaluate the 2D-interval predictions of our models. We discretize the continuous values of the target time series using the method described above and then calculate the `Utility` of the predictions as the total sum of benefits using the matrix in Table 2.4.

Figure 2.4 describes an example of how the Utility is calculated for a concrete 2D-interval prediction. The figure shows the real time series values for a certain k length interval. The black lines (25% and 75%) represent the 2D-interval prediction of a hypothetical

Table 2.4: Benefit matrix.

	low	*normal*	*high*
$\hat{low}$	2	-1	-2
$\hat{normal}$	-1	1	-1
$\hat{high}$	-2	-1	2

model for this k length interval. According to this prediction any value of the series above the predicted upper limit (black line tagged with $\hat{Q_3}$), should be considered "unusually high". Accordingly, any value below the black line tagged with $\hat{Q_1}$, should be considered "unusually low", while values in between these two lines are considered "normal". If we calculate the effective true interval using the observed time series values in that k length interval, we may reach a different interval - the blue lines in the figure. Using these blue lines we can calculate the true class labels of each value in this time interval. Proceeding this way we will conclude that there were some labeling errors induced by the predicted interval, because the predicted interval is narrower. For instance, the green dot was tagged as "unusually high" by the model, but is considered "normal" in reality, while the red dot was tagged as "unusually low" but was also "normal". Using these 3 possible classifications originated by the intervals we can produce a benefit matrix as show in Table 2.4. In this table we can check that the costs associated with both the green and red dots would be -1.

2.5.4 Experimental Results

We now describe the experimental results of our experiments with three different time series tasks: i) a set of artificial time series; ii) time series data from a water quality monitoring application; and iii) data from a water consumption domain.

2.5.4.1 Experiments with Artificial Data

The first set of experiments that we will describe involves the use of artificially created data sets. The goal was to test our hypothesis on a diverse range of time series with different dynamic regimes. Figure 2.5 shows the seven artificial time series we have generated and used in our experiments, each with 3000 values. The gray lines on each graph try to describe the type of regimes in terms of trend and variability that have guided the generation process. As you can observe these series have rather different dynamic regimes

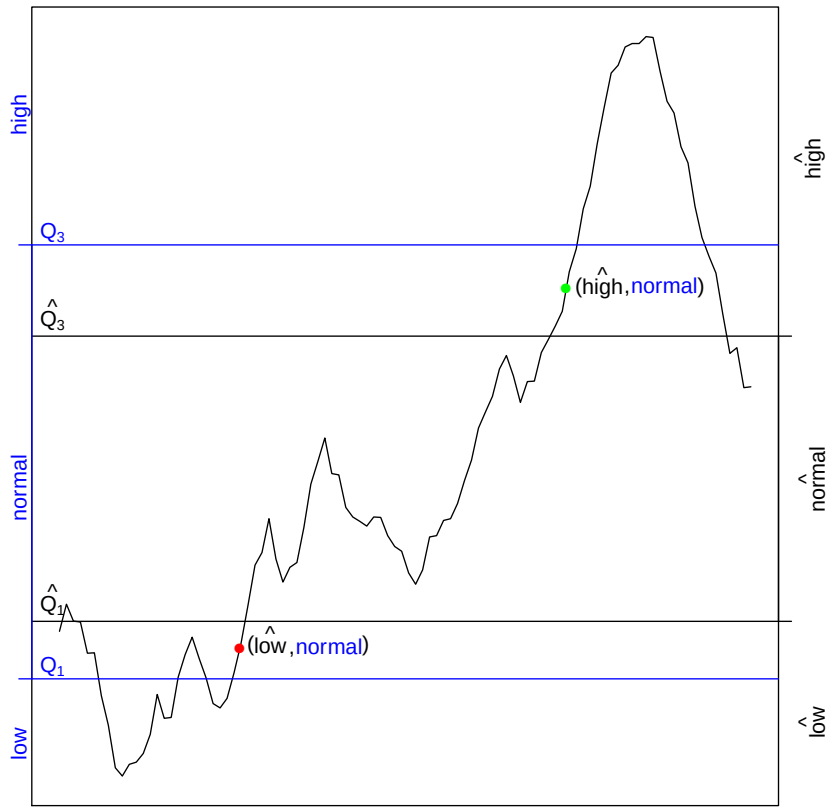


Figure 2.4: An illustrative example of calculating the Utility.

in terms of these two important properties.

We have applied the models described in Section 2.5.2 to these 7 problems, using 3 alternative methods for obtaining 2D-intervals: (i) iterating the model over the k window; (ii) obtaining k different models; or (iii) our proposed method of predicting directly the distribution statistics defining the intervals. Our goal is to check which is the best method for obtaining this type of predictions.

For each experimental setup, we have estimated the values of the 3 evaluation metrics (TQE, MAQ and Utility) described in Section 2.5.3 using 10 repetitions of the Monte Carlo simulation process described in Section 2.5.1. We have used the previous $m = 365$ values of the series to obtain the models that were then used to make 2D-interval predictions for the next $n = 90$ points using a sliding window approach. At each of these 90 prediction points the goal was to estimate the 2D-interval for a future time window of 10, 20 and 30

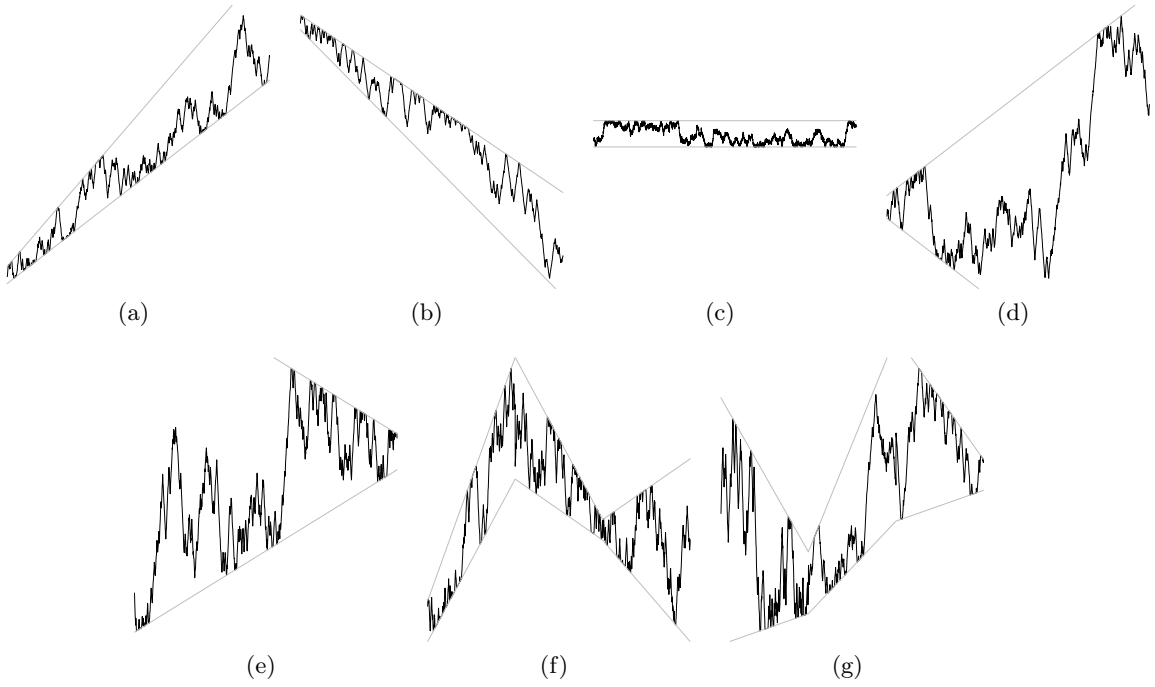


Figure 2.5: Seven artificial time series problems.

k time points, using $h = 1$.

Figures 2.6, 2.7 and 2.8 show the results for 3 different setups evaluated by 3 different measures. The complete list of the results is provided in the web page [2] in the file `ch2ArtificialExps.pdf`. We have selected one graph for each series, metric and interval length. The graph on the Figure 2.6 shows the TQE scores of all model variants on the first time series when predicting a 2D-interval of length 10. The results of the models are grouped in three batches, one for each different approach we are comparing: the iterated approach ("iterated" on the graphs), the use of several models ("k-models" on the graphs), and our proposal of directly predicting the quantiles ("quantiles" on the graphs). On each of the three batches we have one bar for each predictive model variant: 3 for Quantile Regression Forest (QRF); 3 for Regression Forests (RF); 20 variants of Support Vector Machines (SVM); and 4 variants of Regression Tress (RT). To facilitate distinguishing among each model, the bars of the respective parameter variants are represented by the same color. For the complete description of the model parameter variants, see Section 2.5.2.

A vertical dashed line marks the best score on each graph and the result of the baseline RW model is given as a sub-title of the graphs. The model QRF is the only interval forecasting

[2] `goo.gl/hRBMd`

model used in our experiments.

The graph in the Figure 2.7 shows the same type of results this time for the MAQ metric, 4th time series and $k = 20$, while the graph on the Figure 2.8 presents the results in terms of Utility on the 7th series for $k = 30$. Note that while for the first two metrics lower scores are better, for utility it is the opposite.

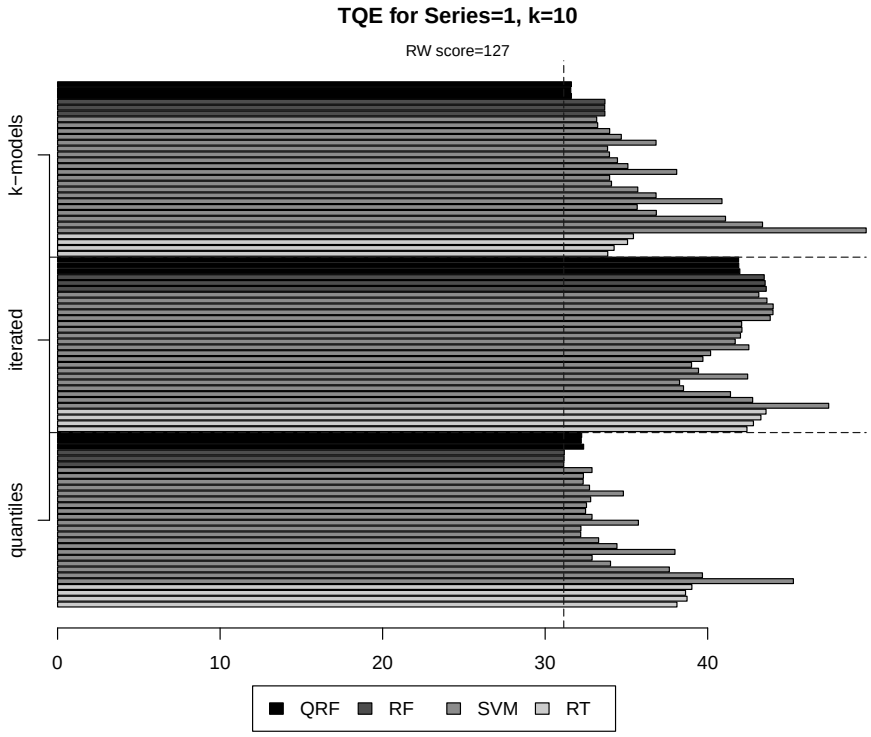


Figure 2.6: The results on stream 1 with $k = 10$.

We have observed that the "k-models" and "quantiles" approaches have rather similar results on these problems. Still, when there is some slight difference this is more frequently favorable to our approach. The "iterated" approach on the contrary is most of the times the worst in terms of scores, although all models typically outperform the baseline RW model, as expected. This pattern of results is similar over all experimental setups we have tried with these 7 problems. These results show that the prediction accuracy of our approach is highly competitive with the existing alternatives. Moreover, these scores are obtained with a significant advantage in terms of computational efficiency. In effect, while our approach requires two models to be obtained (one for each quantile), the "k-models" approach requires as many models as the length of the interval, i.e. k models. This is

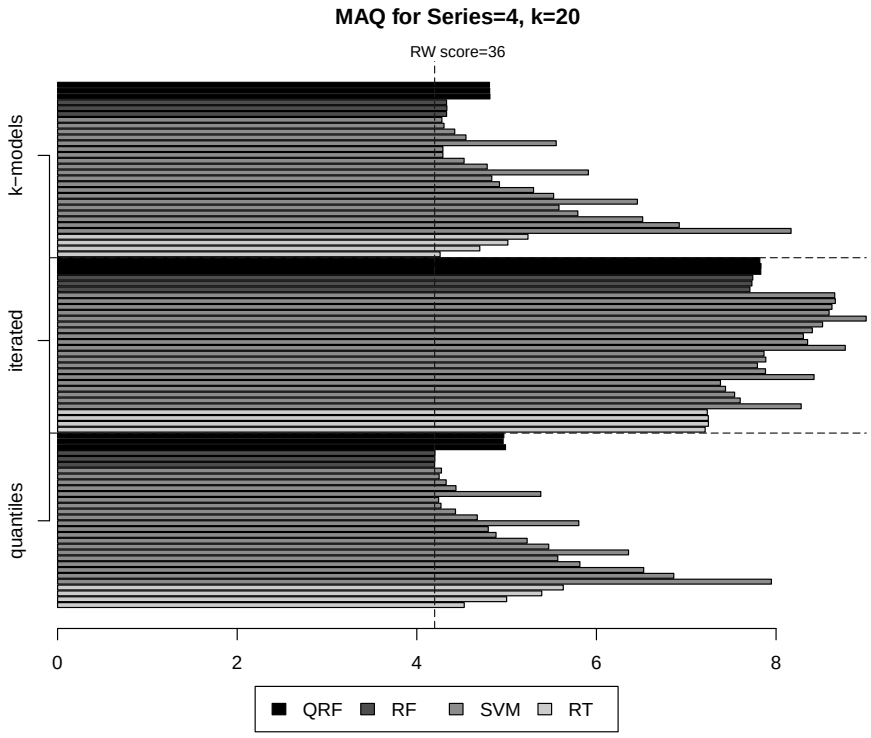


Figure 2.7: The results on stream 4 with $k = 20$.

a significant difference as shown in Figure 2.9. This figure shows the ratio between the computation time of "k-models" over our approach. We can observe that, depending on the size of the interval, the "k-models" approach can take from 5 to 30 times more time to be obtained. On dynamic environments where new data is constantly being collected, eventually requiring new models to be obtained, this margin can be rather significant. We do not show the results for the "iterated" approach as they are essentially similar to our proposal.

Summarizing, the experiments on these artificial problems show that our proposal is able to achieve a rather competitive prediction accuracy with a significantly smaller computational cost.

2.5.4.2 Experiments with Water Quality Parameters

The first real world application of 2D-intervals we describe is related with water quality control in the distribution network of the metropolitan region of Porto, Portugal (serving roughly 1 million people). The company (AdDP) that manages this network has legal

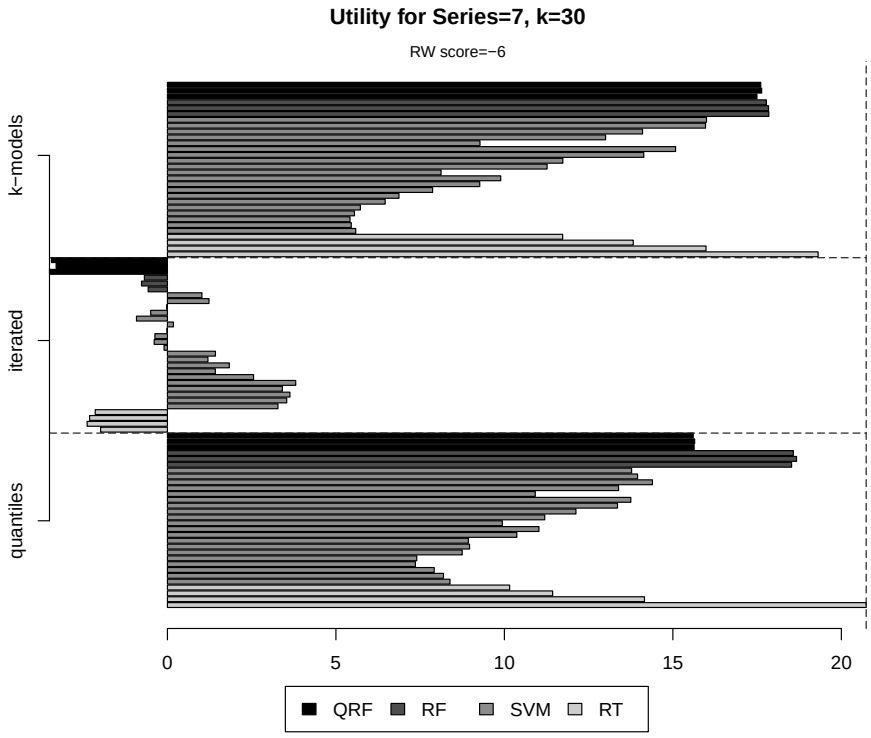


Figure 2.8: The results on stream 7 with $k = 30$.

limits that must not be crossed for several water quality parameters that are monitored. With the goal of avoiding the danger of crossing these limits (and paying the respective high fines) the company internally establishes tighter limits that if broken generate an alarm that leads to several control actions over the network. These actions have associated costs and thus having these internal limits too tight will lead to a high operational cost of the network. However, having wider internal limits, possibly too near the legal ones, will increase the risk of when the alarms are fired being too late for the control actions to have any effect to avoid breaking the legal limits. This means that establishing the intervals of acceptable (normal) values of a large set of water quality parameters is a problem with high socio-economical impact for the company and the region.

According to the company know-how of the problem these intervals that establish a range of normality for each parameter change along the year as the rivers from where the water is collected are very dynamic and change a lot during the seasons of the year. This means that this notion of normal parameter values is dynamic, though the legal limits are fixed. Still, the changes on the expected normal values for each parameter do not occur on a daily

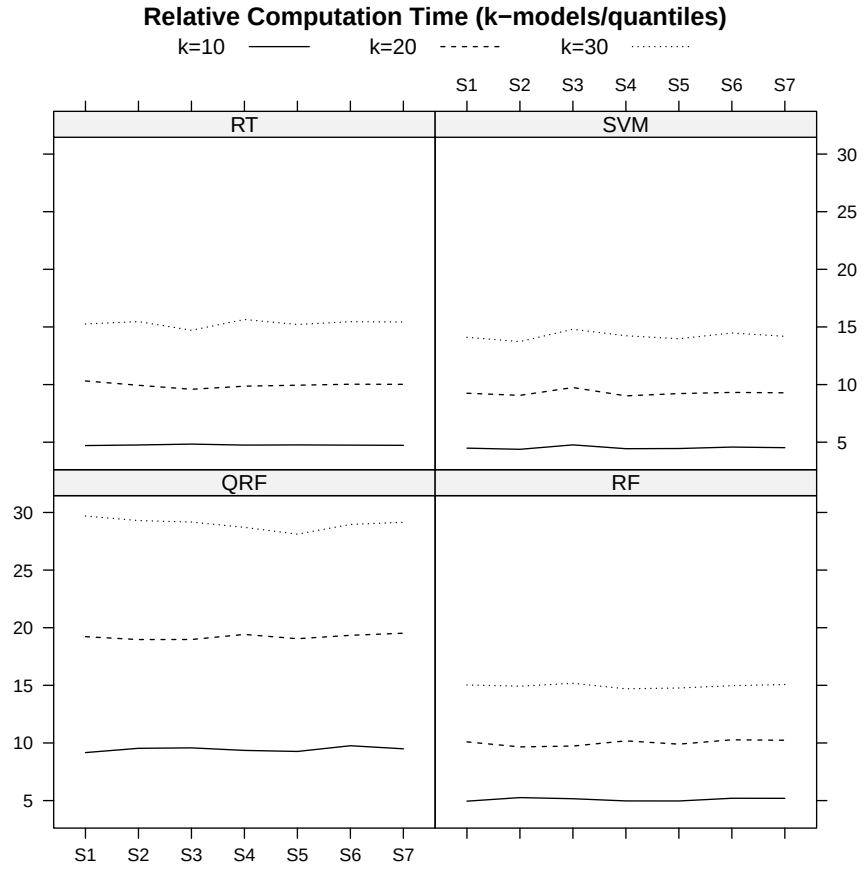


Figure 2.9: The relative computation times of "k-models" vs "quantiles" on the 7 artificial time series.

basis, but are expected to potentially change within a slower time frame. According to the company, it would be of the foremost usefulness to have information on the expected range of normality for each parameter on a monthly basis.

The AdDP company has provided us daily data concerning a large set of water quality parameters along several years (2000 till 2008). We will focus our analysis on the task of trying to forecast a 2D-interval of normal values for a small set of parameters (`pH`, `iron`, `turbidity` and `aluminium`) using this data set. This is a task similar to the ones described in Section 2.5.4.1. The interval of "normal" values can again be approximated by the expected 1st and 3rd quartiles on a future time window, i.e. a 2D-interval. We have used exactly the same prediction tasks and predictors as in the artificial problems (c.f. Equation 2.3). However, following the company requirements, we have only applied our method for a 2D-interval of the next 30 days (i.e. $k = 30$ and $h = 1$).

The results we report are again estimates of the 3 evaluation statistics using 10 repetitions of a Monte Carlo simulation. We have used the previous $m = 365$ values of each of the four time series and tested the models along a $n = 90$ days window, again using a sliding window approach.

The Figures 2.10, 2.11 and 2.12 show the results for 3 different setups using the same schema as before (the complete list is available in the web page [3] in the file `ch2WaterQualityExps.pdf`). On Figure 2.10 we have the TQE scores for the `Iron` time series. Figure 2.11 illustrates the results in terms of MAQ for the `pH` parameter, while Figure 2.12 shows the `Utility` results for `Turbidity`. The overall pattern is similar to the results on the artificial problems. The "iterated" approach is clearly the worst method, while our approach and "k-models" get similar scores. Still, compared to the results on the artificial data, we observe a more marked advantage of our proposal. The exception is the QRF model variants where the "k-models" approach achieves better results whilst not the best overall scores.

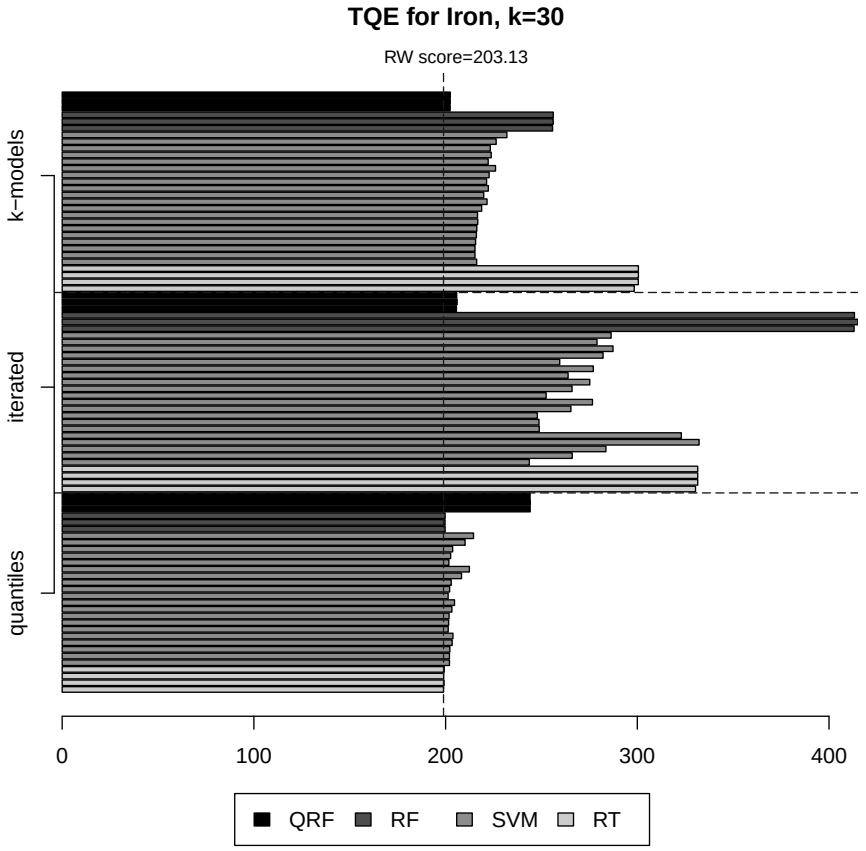


Figure 2.10: The results on Iron with $k = 30$.

[3] `goo.gl/hRBMd`

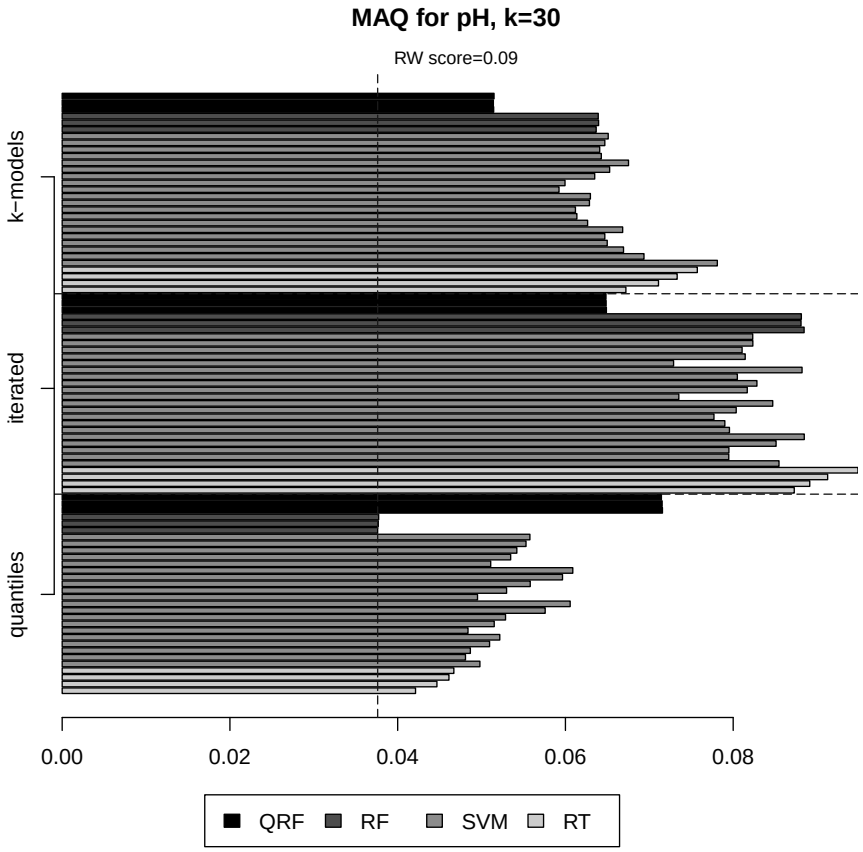


Figure 2.11: The results on pH with $k = 30$.

In terms of computation times the results are shown in Figure 2.13. Again we observe a significant overhead of the "k-models" approach when compared to our proposal.

The experiments on water quality parameters show similar results as the experiments with artificial data. Our proposal achieved competitive prediction accuracy with a significative smaller computational cost.

2.5.4.3 Experiments with Water Consumption

The second real world problem concerns again a water distribution network, this time in the south of Spain. The problem here is related to production planning in order to face the varying demand in terms of water consumption. We have hourly data concerning the water consumption in a residential area of the water distribution network from January, 2005 till April, 2005. Our goal is to forecast a 2D-interval for the next 12 and 24 hours (i.e. $k = 12$ or $k = 24$, $h = 1$). The distribution of the water demand values has very marked

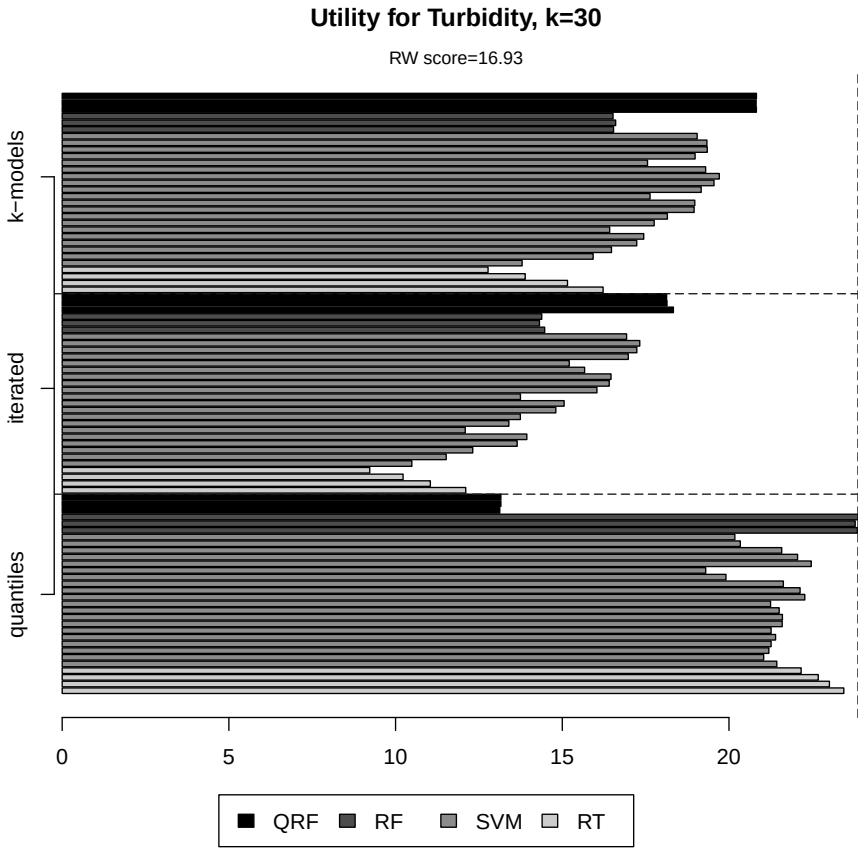


Figure 2.12: The results on Turbidity with $k = 30$.

seasonal properties not only along the different periods of the day, but also across similar weekdays. Because of this, we use a slightly different approach in terms of predictors with the goal of providing the models with information on this weekly seasonal effects, c.f. Equation 2.7. For $k = 12$ we have used the last 6 values of the demand ($p = 6$ on Equation 2.7), the same information as on previous problems regarding the quartiles, mean and standard deviation on the past k values, but also the quartiles that we want to forecast measured in the previous week (i.e. on the same weekday/hour) to provide information on this observed weekly seasonality ($Q_{\alpha,t}^{-k_{pw}}$ and $Q_{\beta,t}^{-k_{pw}}$). In the case of $k = 24$ we have increased the number of past values of the series from 6 to 12 ($p = 12$), while the other predictors stayed unchanged.

$$TGT = f(Y_{t-1}, \cdots, Y_{t-p}, Q_{\alpha,t}^{-k}, Q_{\beta,t}^{-k}, Q_{\alpha,t}^{-k_{pw}}, Q_{\beta,t}^{-k_{pw}}, \bar{Y}^{-k}, \sigma_Y^{-k}) \tag{2.7}$$

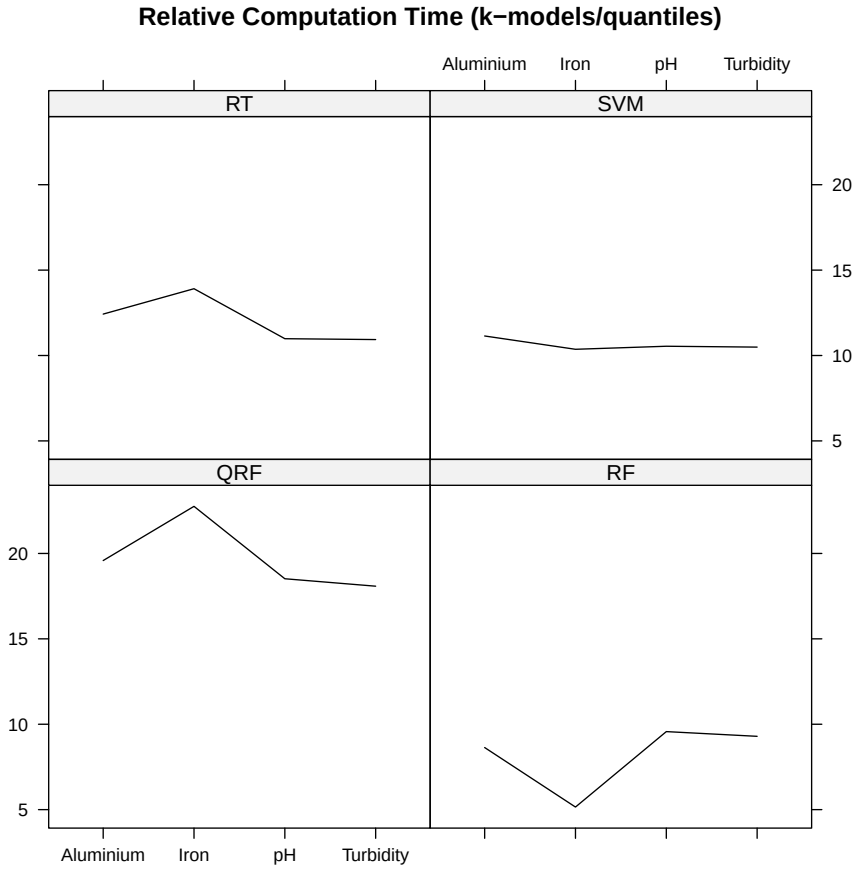


Figure 2.13: The relative computation times of "k-models" vs "quantiles" on the water quality problem.

The main difference from Equation 2.3 is the addition of two new predictor variables, $Q_{\alpha,t}^{-k_{pw}}$ and $Q_{\beta,t}^{-k_{pw}}$ that represent the quantiles (α and β) in the previous week.

Regards the experiments we have used again 10 random repetitions of a Monte Carlo simulation this time with a training window of $m = 1344$ values (8 weeks) and a test window of $n = 336$ values (2 weeks), for which predictions were obtained using a sliding window approach. The size of the 2D-intervals (k) was set to 12 and 24 hours (the complete list of results is available in the Appendix A).

The results of 3 different setups of these experiments are show in Figures 2.14, 2.15 and 2.16. The pattern of results is similar to the one observed in the water quality problems. Both "k-models" and our approach have similar results with a slight advantage of our method, while the "iterated" approach clearly lags behind. Once again we have observed that the QRF models achieve better results with the "k-models" approach. We should note

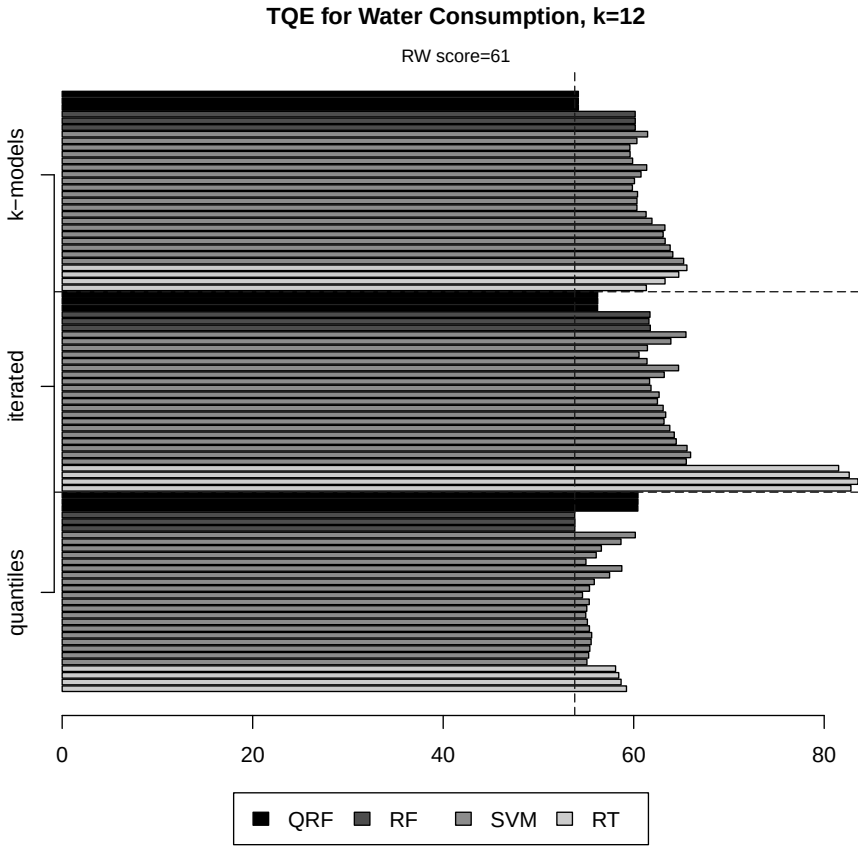


Figure 2.14: The TQE results on water consumption with $k = 12$.

that the `Utility` score is particularly relevant for this type of applications where serious mistakes may have significant financial costs.

In terms of computation times the scores are shown in Figure 2.17, and reveal the same type of advantage of our proposal over the "k-models" alternative.

This experiment showed a similar outcome as the previous two experiments. Our proposed approach achieved competitive results in terms of prediction accuracy together with reduced computational costs.

2.6 Conclusions

We have described a new type of time series forecasting task - 2D-interval predictions. This type of forecasting problems has high relevance for several application domains. To the best of our knowledge there is no established methodology to handle these problems across

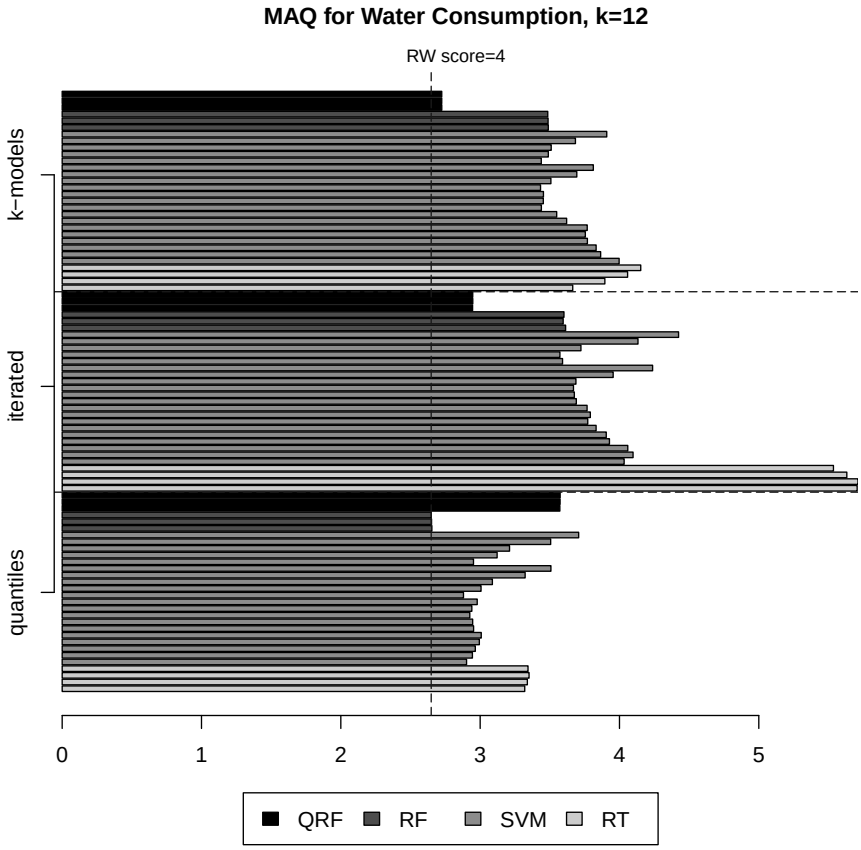


Figure 2.15: The MAQ results on water consumption with $k = 12$.

the several disciplines that address time-dependent data. In this context, our main goals were: (i) to raise the awareness of the data mining community to these relevant problems, and (ii) to propose a new methodology to address these tasks that can be used with any time series modeling technique. The key idea of the proposal is to directly predict the distribution statistics for the target time interval. As a concrete instance of this approach, we have focused on tasks of forecasting the range of plausible values for a future time period. We have approximated this range by means of an interval formed by two standard non-parametric statistics - the first and third quartiles.

We also proposed three evaluate measures adapted to this new class of problems, the prediction of 2D-intervals: (i) the TQE function which is an adaptation of existing measures on quantile regression, (ii) the MAQ which is based on the absolute distance between the predicted and real quantiles, and (iii) the utility metric motivated by the cost-sensitive learning, which maps the problem to a classification task and uses a benefit matrix to

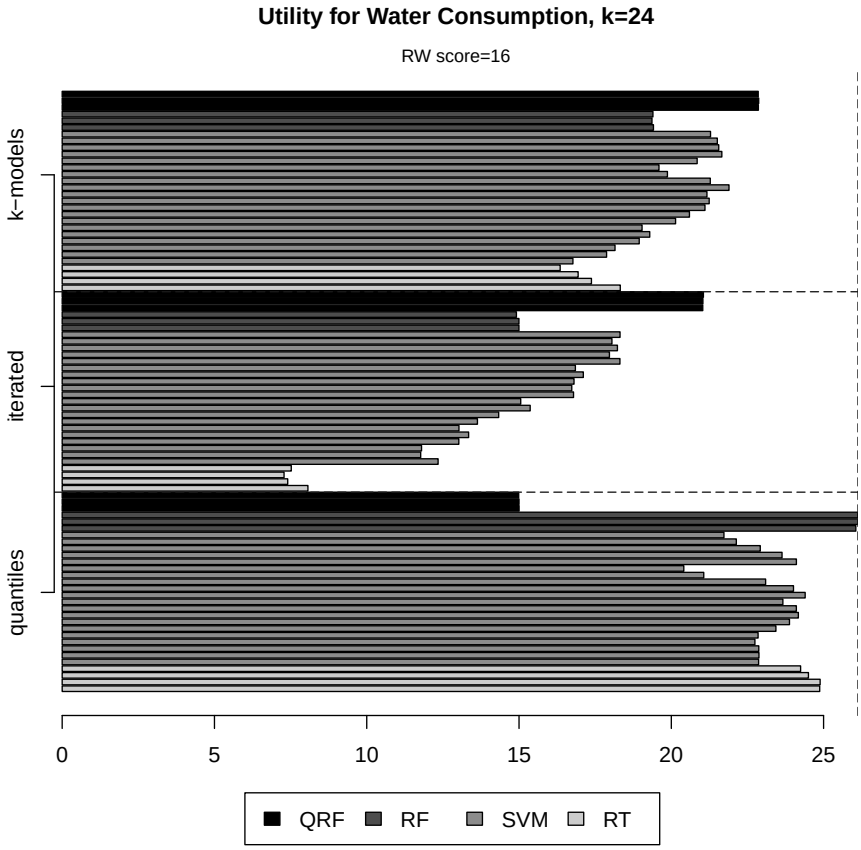


Figure 2.16: The Utility results on water consumption with $k = 24$.

calculate the total utility of the 2D-interval predictions.

We have described our proposal and have carried out an extensive set of experiments designed with the goal of checking the validity of the proposal when compared to existing alternatives. This comparison was carried out from two perspectives: (i) the perspective of prediction accuracy of the 2D-intervals, and (ii) the perspective of the computational cost of the alternatives. This latter issue may be particularly relevant in dynamic contexts where new data arrives at a high pace, like data streams. The question of the accuracy of the predictions was also addressed from different perspectives trying to capture characteristics that are important to this type of applications (e.g. the costs and benefits of the predictions).

The results of our experiments with several artificial time series and also two real world problems provide clear evidence on the validity of our proposal. It achieves a prediction accuracy that it is highly competitive with the best alternatives in the hundreds of different

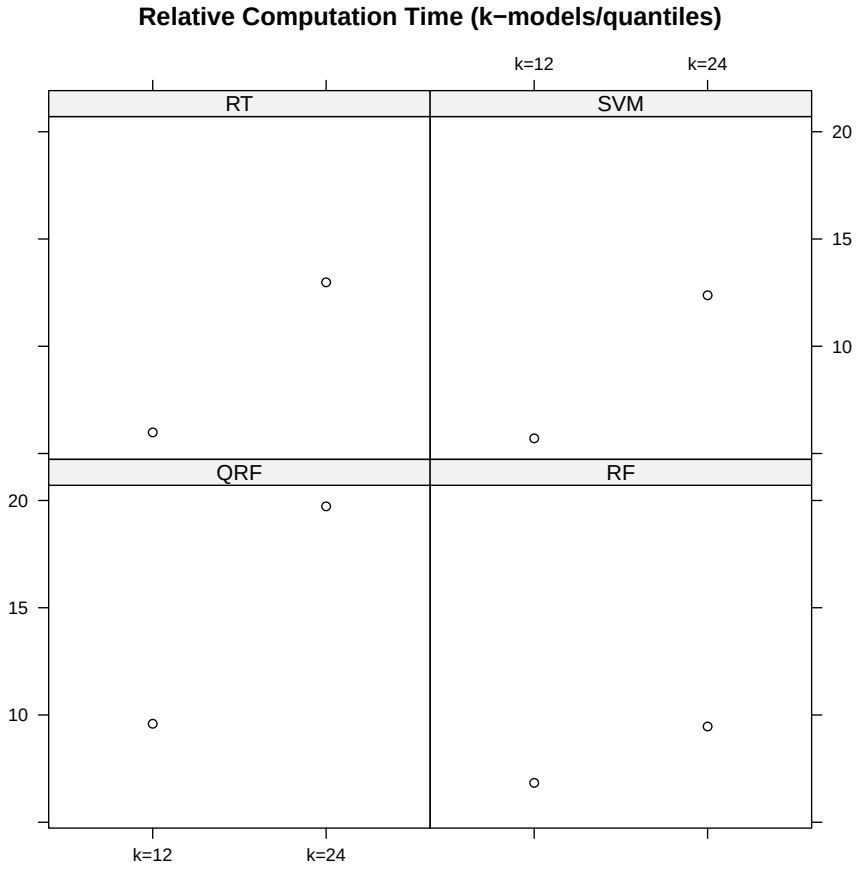


Figure 2.17: The relative computation times of "k-models" vs "quantiles" on the water consumption problem.

experimental setups that were considered, but with a significantly lower computational cost. This makes the proposal particularly adequate for high-frequency time series where 2D-interval predictions may be of use.

Chapter 3

A Multiple Regression Approach for Spatial Interpolation

In this chapter we propose a new spatial imputation method based on machine learning algorithms and a series of data pre-processing steps. The key distinguishing factor of this method is that allows the use of data from faraway regions, contrary to the state of the art on spatial data mining. We evaluate our methodology in the domain of image inpainting, by filling in unknown pixels on several images. We compare it to state of the art methods and provide strong experimental evidence of the advantages of our proposal. We also compared our approach against specific image inpainting algorithms, which resulted in observing a significant advantage of our method.

3.1 Introduction

Many real world data mining applications involve analysing geo-referenced data. These data are frequently obtained from measurements of real systems, e.g. wind speed, oil resources analysis, water quality assessment, satellite images, pictures and/or paintings repair, etc. The process of data collection is not fully controllable and it is prone to failures. Frequently, incomplete data sets are generated, in the sense that not all geographical coordinates have measured values of the variable(s) of interest. This incompleteness may be caused by poor data collection, measurement errors, costs associated with the collection

or with the analysis of the measurements and many other factors. These missing values may cause several difficulties in many applications based on these data, seriously impacting on the posterior data analysis. Moreover, other constraints, e.g. financial and human resources, may even increase the amount of missing data. Spatial imputation/interpolation methods try to fill in these unknown values in geo-referenced data sets. In this context, it is of key importance to have methods to accurately fill in these missing values, which is confirmed by the amount of literature and methods available for spatial interpolation. For a detailed comparison between different spatial interpolation methods see [Li and Heap, 2010].

In this chapter we propose a new spatial imputation method based on machine learning algorithms and a series of data pre-processing steps. The key distinguishing factor of this method is the possibility of using the data from faraway regions, contrary to the state of the art on spatial data mining. This technique also allows the use of advanced non-linear machine learning algorithms in the solution of this problem. Specifically, we may use any regression model with this solution, such as: regression trees, support vector machines, random forests, etc. These algorithms are considerable more advanced when compared with the state of the art in the spatial interpolation research area.

Pixels from images (e.g. from a satellite or video surveillance cameras) are an example of data that may also suffer from this incompleteness where some pixels are missing, which again may be caused by many factors. An image can be seen as a spatial data set in a Cartesian coordinates system, where each pixel (location) registers some value (e.g. degree of gray on a black and white image). Being able to recover the original image from a partial or incomplete version of the reality is a key application in many domains (e.g. surveillance, security, etc.). In this chapter we evaluate our general methodology for spatial interpolation on this type of problems. Namely, we check the ability of our method to fill in unknown pixels on several images. We compare it to state of the art methods and provide strong experimental evidence of the advantages of our proposal.

The main idea behind any approach to spatial imputation is the assumption that the value at any location has some form of dependence on the values on neighbouring locations. This is supported by the *First Law of the Geography* that says that "everything is related to everything else, but near things are more related than distant things" [Tobler, 1970].

Our work is also based on this assumption. However, the fundamental difference of our proposal when compared to the state of the art, is the fact that *we also allow the use of data from faraway regions provided these neighbourhoods have similar spatial dynamics to the target location* for which we want to fill in a value. The fact that two neighbourhoods are distant from each other does not preclude them from having similar spatial behaviour. Ignoring these similarities and data seems a waste. The main contribution of our work is to provide means to uncover these similarities and allowing the use of this extra data from distant apart regions. This means that our methods will tend to use much more data than existing methods to estimate the unknown values. The hypothesis driving our work is that this extra information will lead to gains in terms of the precision of the imputation.

We have tested and compared our proposal against a series of alternative state of the art methods on a particular task of filling in the missing pixels on pictures. Still, the approach is by no means restricted to this particular application and can actually be seen as a general approach to spatial imputation and even to the problem of formalising prediction tasks in the context of spatial data. Our experiments show that our approach significantly outperforms the most common techniques used for spatial interpolation - Inverse Distance Weighted (IDW) and Kriging according to Li and Shi [2010]). The particular application of filling in gaps in images is also handled within the research area of image processing. On this research area other approaches exist to this problem. One of the most used approaches is the Inpaint technique [Agrawal et al., 2010; Bertalmio et al., 2000]. We have also compared our approach against the Inpaint method and the results show a clear advantage of our technique.

In Section 3.2 we describe the most common techniques used for spatial interpolation. Section 3.3 describes our proposed approach and Section 3.4 the particular application used to test and compare our method. Section 3.5 presents the experimental evaluation of our proposal, including the data and experimental methodology that were used, as well as the presentation and discussion of the results. Finally, on Section 3.6 we draw the conclusions of this work and describe possible future research directions.

3.2 Spatial Interpolation

Forecasting the missing values in spatial data sets is not a new problem and it is usually known as spatial imputation or interpolation. Spatial interpolation methods address the problem of estimating unknown values of a variable of interest, Z, on certain geographical locations, based on a spatial data set $\mathcal{Z} = \{z_1, \cdots, z_n\}$, where z_i is the value of the variable Z at location i.

Many different approaches have been developed to solve the spatial interpolation problem. Existing approaches are usually motivated by the first law of the geography [Tobler, 1970] that prescribes that nearby points should have strongly correlated values. Li and Shi [2010] classifies spatial interpolators in three main classes: non-geostatistical interpolators, geostatistical interpolators and combined procedures that integrate approaches from the two former classes.

Non-geostatistical interpolators are based on the distance between the neighbours. The simplest method is the Distance Interpolator (DI) that consists on the use of the average value of the spatial neighbours as an approximation to the value at the missing location,

$$DI_{\beta}(o) = \frac{1}{|\mathcal{N}_o^{\beta}|} \sum_{z_i \in \mathcal{N}_o^{\beta}} z_i \tag{3.1}$$

where $\mathcal{N}_o^{\beta}$ are the values in the neighbourhood of the target location o defined as,

$$\mathcal{N}_o^{\beta} = \{z_i \in \mathcal{Z} : d(o, i) < \beta\} \tag{3.2}$$

where $d()$ a distance function between locations and β the threshold that limits the size of the neighbourhood region.

The Inverse Distance Weighted Interpolation (IDW) [Isaaks and Srivastava, 1989] is a simple improvement of the DI method. It is based on the assumption that the values that are farther apart within the neighbourhood of a point should contribute less to the average calculation. In this context, this method approximates the value at an unknown location as the weighted average of the known neighbourhood values, where the weights are inversely

proportional to the distance from the target location,

$$IDW_\beta(o) = \sum_{z_i \in \mathcal{N}_o^\beta} w_{o,i} \cdot z_i \tag{3.3}$$

where $w_{o,i} = \frac{1}{d(o,i)}$ and the weights must satisfy $\sum_{z_i \in \mathcal{N}_o^\beta} w_{o,i} = 1$.

The second class of existing methods are geostatistical interpolators that have origin in the work of Krige [Krige, 1951]. Kriging is a generic name for a family of generalised spatial interpolation models. According to Mitas and Mitasova [1999] kriging assumes that the spatial distribution of a geographical region can be modeled by the realisation of a random function, using a statistical technique to analyse the data. Kriging uses the same basic principal behind the inverse distance weighting technique - it approximates the unknown value at a location by interpolating the values at known locations given more importance to the closer neighbours. However, the way the weights are calculated is different as kriging uses the covariation between known data at various spatial locations [Krige, 1951]. There are several variants of kriging most of which differ on the way these weights are approximated. Frequently used variants include ordinary kriging and co-kriging. In this chapter we have only considered ordinary kriging because co-kriging requires an auxiliary variable (covariable) [Isaaks and Srivastava, 1989], which was not available in the domain considered in this chapter.

To compute the krige method, it is necessary to define a metric for the statistical distance. A common used distance is the empirical semivariogram γ_r^β [Isaaks and Srivastava, 1989],

$$\gamma_r^\beta = \frac{1}{2|\mathcal{N}_r^\beta|} \sum_{z_s \in \mathcal{N}_r^\beta} (z_r - z_s)^2 \tag{3.4}$$

To calculate the krige method it is also necessary to have information about the spatial correlation on locations where no samples are available. This is calculated by fitting a curve into the computed values of the empirical semivariogram (typical choices are: spherical, gaussian, exponential, etc.). The resulting semivariogram model is then used to find the correlation between the values at unsampled locations. Once a semivariogram model is defined, it can be applied for spatial interpolation using kriging. The ordinary kriging

interpolator for variable Z at the location o is given by,

$$OK_\beta(o) = \sum_{z_i \in \mathcal{N}_o^\beta} \gamma_i^\beta \cdot z_i \tag{3.5}$$

where the weights must fulfil the unbiasedness condition, i.e., $\sum_{z_i \in \mathcal{N}_o^\beta} \gamma_i^\beta = 1$ and the expected error is $E[\hat{z}_i - z_i] = 0$ [Isaaks and Srivastava, 1989].

On top of these standard approaches to the spatial interpolation problem, many other variants exist on the vast research literature on this topic. The following is a brief overview of some of the most representative.

Lu and Wong [2008] proposed a variation of the inverse distance weighting (IDW) method to forecast the missing precipitation values in the center region of Taiwan. The main contribution of this work was the development of an adaptive inverse distance weighting (AIDW) model. This model uses different neighbourhood sizes based on the density of the unsampled location. For unsampled locations with high concentration of points a smaller neighbourhood is used and a larger one is used in dispersed locations. This model was compared against standard IDW models with different neighbourhood size and ordinary kriging. The proposed method outperformed all variants of traditional IDW and was better than ordinary kriging in some experiments.

Goovaerts [2000] compares two classes of spatial interpolation methods, in the prediction of precipitation values in the region of Algarve, Portugal. Univariate interpolation methods (Thiessen polygon, inverse square distance and ordinary kriging) against multivariate methods (linear regression, simple kriging with varying local means, kriging with external drift and co-kriging). The external variable used by the multivariate methods was the elevation map of the region in analysis. In this comparison the multivariate methods outperformed the others interpolators. Linear regression was the best model. However, ordinary kriging outperformed linear regression when the correlation between rainfall and elevation is moderate (less than 0.75).

Rigol et al. [2001] propose a neural network approach for the spatial interpolation of the daily minimum air temperature in the region of Yorkshire, UK. The proposed methodology used terrain variables of the predicted location and temperature observations of the neigh-

boring locations as inputs for the neural network. The authors claim that the proposed technique achieves similar accuracy to the state of the art methods. However, they highlight the advantage of not requiring any linear characteristics in the data to achieve these results.

Umer et al. [2010] proposed a distributed kriging algorithm for spatial interpolation in wireless sensor networks. The main challenge of this domain was to accurately interpolate the unsampled locations respecting the limitation of energy consumption. The proposed technique was shown to be significantly more efficient in terms of energy consumption than a global implementation and more accurate than simple averaging.

Xie et al. [2011] applied spatial interpolation techniques in the task of forecasting the presence of heavy metals pollution in Beijing, China. The main motivation of this work was to reduce the costs in the analysis of soil samples. In this work several interpolators were compared: inverse distance weighting (IDW), ordinary kriging (OK), radial basis functions (RBF) and local polynomial interpolation (LP). All the methods tested had a high prediction rate, however OK and RBF had better performance in lower size polluted areas.

Lin and Chen [2004] proposed a hybrid model, which was a combination of a radial basis function network with a semivariogram model. This new model was denominated improved radial basis function network (IRBFN). The proposed model was compared against ordinary kriging (OK) and standard radial basis function network (RBFN), in the prediction of precipitation in Tanshui River Basin in northern Taiwan. The proposed technique outperformed the other two models. The authors also claim that the new model is considerably faster than ordinary kriging on large data sets.

Robinson and Metternicht [2006] tested four spatial interpolation techniques (inverse distance weighting, ordinary kriging, lognormal ordinary kriging and splines) in the analysis of three soil properties: pH, electric conductivity and organic matter. There was no single interpolation technique that outperformed others in all setups analysed. Ordinary kriging had the best results in the analysis of the property pH, splines outperformed kriging and inverse distance weighting in the analysis of organic matter, and lognormal kriging outperformed the others in the analysis of electric conductivity.

Malerba [2008] presents the area of spatial data mining and summarizes the challenges of this area in the context of relational approaches in database management systems.

Malerba et al. [2005] propose a new algorithm (Mrs-SMOTI) that creates a spatially adapted regression tree model that is able to integrate with a spatial database. They highlight three main characteristics of the proposed approach: (i) the proposed model is able to capture both global and local spatial effects from explanatory attributes; (ii) the model is not limited to a single layer analysis to find patterns between explanatory attributes and the response attribute; and (iii) the definition of the spatial relationships and attributes is influenced by the geometrical representation and relative positioning of the spatial objects. They evaluated the proposed algorithm using a real world spatial data set, the census data of Stockport metropolitan district in Greater Manchester (UK), showing a clear advantage of the proposed method.

Ceci et al. [2010] propose a transductive learning approach for spatial classification, transforming the spatial database into a relational database with multiple relations. The proposed solution is based on an iterative K-nearest neighbors algorithm for re-classification of labelled and un-labelled data. The method was evaluated with success using two real world datasets: census information from the Greater Manchester (UK) and Munich rental guide (DE).

3.3 Our Proposal - Multiple Regression Spatial Interpolation

Spatial interpolation aims at filling in the values of a variable of interest at geographical locations for which they are unknown. This problem is usually solved by assuming that the unknown values can be filled in by using the information of the known values in their vicinity. It is possible to look at this task as a prediction problem where the target variable is the variable of interest at a certain geographical location and the predictors are the values of this variable within the respective neighbourhood. We have taken this approach by mapping the spatial interpolation problem into a numeric forecasting task, i.e. a multiple regression problem. The main motivation for this transformation came from times series forecasting using machine learning models. Typically, these tasks are addressed as standard multiple regression problems using the technique of time delay embedding [Takens, 1981]. This technique consists on setting the future value of the time series at time $t + h$ as the target variable and then using a certain number of previous values (the embed size) of the time series as predictors. The key idea is to provide the modelling techniques with

information on the recent dynamics of that time series by means of the most recent values. Our proposal can be seen as applying the same idea in the spatial dimension by trying to forecast the value of a variable at a certain location as a function of the values in the spatial vicinity (a kind of spatial embedding).

Other authors have addressed the use of regression tools with spatial data (e.g. [Brunsdon et al., 1998]). Still, to the best of our knowledge all these works constrained the use of data to make predictions for a certain location to the neighbouring data (e.g. through kernels [Brunsdon et al., 1998]). These approaches are in accordance with the already mentioned first law of geography. The main distinguishing feature of our proposal is letting the models and their search criteria to select which observations are useful from the perspective of predictive accuracy. As we have mentioned before, the fact that an observation is faraway from the target location does not mean that it does not have the same type of spatial dynamics.

In financial time series forecasting it is common to extend the idea of time delay embedding by adding other predictors. Examples include "technical indicators", which provide information concerning the recent dynamics of the target time series. This information takes the form of summary statistics (the indicators) of several properties that are deemed interesting in terms of understanding the time evolution of the series. Examples include indicators of tendency, acceleration, momentum, volatility and so on. Adding these indicators as predictors to the "standard" time delay embedding values usually results in improvements on the predictive accuracy of the obtained models. We will import this idea into spatial data sets by proposing a series of *spatial indicators*.

Summarising, our proposal for the spatial interpolation problem consists of two key pre-processing ideas:

- Mapping the spatial interpolation problem into a multiple regression task;
- Propose a series of spatial indicators to better describe the spatial dynamics of the variable of interest.

The first idea has two main advantages: (i) allows the use of the large number of sophisticated function approximators that are available; and (ii) allows the use of data from faraway neighbourhoods if the models find them similar to the region being interpolated, in terms of

the predictor variables that are selected for a given task. Regards the second idea, we have considered three classes of properties to describe the spatial dynamics between the variable values in a neighbourhood: i) properties describing the typical value of the target variable; ii) properties describing the variability of the variable; and iii) properties describing the tendency (in spatial terms) of the variable. Among these, the third class is the one that differentiates more our work from the information used in standard approaches to spatial interpolation. Still, we should remark that standard approaches use these indicators to directly forecast the unknown values, while we are using them as predictors in a regression model, thus allowing for the discovery of possible non-linear interactions between the properties.

The typical value of the target variable within a neighbourhood can be captured by both the Distance Interpolator (Equation 3.1) and the Inverse Distance Weighted Interpolation (Equation 3.3), the difference being that the latter weights the contribution of the points by the distance to the target. In this context, we will use these values as predictors in our models. To simplify our notation we will use $\overline{z}(\mathcal{N}_o^\beta)$ for the standard averages ($= DI_o^\beta$), and $\widetilde{z}(\mathcal{N}_o^\beta)$ for the weighted averages ($= IDW_o^\beta$).

To capture the notion of spread of the values within a certain vicinity we have used the standard deviation calculated with the values in this neighbourhood,

$$\sigma_z(\mathcal{N}_o^\beta) = \sqrt{\frac{1}{|\mathcal{N}_o^\beta|} \sum_{z_i \in \mathcal{N}_o^\beta} (z_i - \overline{z}(\mathcal{N}_o^\beta))^2} \tag{3.6}$$

In financial forecasting it is common to describe the tendency of a time series by means of a ratio between two moving averages calculated using two different embed sizes. If the value of the moving average with shorter embed size surpasses the longer moving average we know that the time series is on an upwards tendency, while the opposite indicates a downwards direction. We have imported this idea into the spatial dimension. The ratio between two averages calculated on two spatial neighbourhoods with different sizes around the target location provides us with information on how the target variable values evolve in the vicinity of this location. If the shorter average is above the longer, then we know that values are increasing as we approach the target location, while the opposite occurs if the shorter average is smaller. This ratio can be defined as follows,

$$\overline{Z}_o^{\beta_1,\beta_2} = \frac{\overline{z}(\mathcal{N}_o^{\beta_1})}{\overline{z}(\mathcal{N}_o^{\beta_2})} \tag{3.7}$$

where β_1 and β_2 are two neighbourhood sizes ($\beta_1 < \beta_2$) and $\overline{z}()$ is the average of a set of points in the neighbourhood of o.

A variation of this indicator can be easily obtained by using weighted averages of the values within the spatial neighbourhood,

$$\widetilde{Z}_o^{\beta_1,\beta_2} = \frac{\widetilde{z}(\mathcal{N}_o^{\beta_1})}{\widetilde{z}(\mathcal{N}_o^{\beta_2})} \tag{3.8}$$

where $\tilde{z}()$ is the weighted average of a set of points in the neighbourhood of o.

Having defined a series of spatial indicators, we can proceed to map the spatial interpolation problem into a multiple regression task. The target variable of this task is the value of the variable Z at a geographical location. As predictors we propose to use several variants of the spatial indicators we have described above. Namely, we will estimate the value of Z at a target location o as a function of the following predictors,

$$\begin{aligned} \hat{z}_o = f(&\overline{z}(\mathcal{N}_o^{k_1}), \overline{z}(\mathcal{N}_o^{k_2}), \overline{z}(\mathcal{N}_o^{k_3}), \overline{Z}_o^{k_1,k_2}, \overline{Z}_o^{k_2,k_3}, \\ &\widetilde{z}(\mathcal{N}_o^{k_1}), \widetilde{z}(\mathcal{N}_o^{k_2}), \widetilde{z}(\mathcal{N}_o^{k_3}), \widetilde{Z}_o^{k_1,k_2}, \widetilde{Z}_o^{k_2,k_3}, \\ &\sigma_z(\mathcal{N}_o^{k_1}), \sigma_z(\mathcal{N}_o^{k_2}), \sigma_z(\mathcal{N}_o^{k_3})) \end{aligned} \tag{3.9}$$

where $f()$ is the unknown regression function we are trying to model using a set of training data $\mathcal{Z}$, and k_1, k_2 and k_3 (with $k_1 < k_2 < k_3$) are spatial neighbourhood sizes. In the experiments described in this chapter we have used the values $10, 30$ and 50, respectively, for these spatial neighbourhood sizes, but this is something that obviously is domain-dependent.

It is important to remark that several other indicators/predictors could have been used. The same can be said regards the sizes of the spatial neighbourhoods. Which predictors to use is a well-studied problem on predictive data mining. Several established methods exist to search and select the best predictors for a given data set and learning algorithm [Guyon

and Elisseeff, 2003; Kira and Rendell, 1992]. It is not the goal of this work to address this well-studied subject. Our contribution is the idea of mapping the problem of spatial interpolation into a multiple regression task and also to provide some new predictors that capture the spatial dynamics on a certain vicinity.

Tables 3.1 and 3.2 summarize the spatial pre-processing technique proposed in this thesis. Table 3.1 represents a spatial data set to be transformed to a regression data set (Table 3.2), applying our proposed technique. For instance, to obtain the data for the location $(44, 39)$ (in black on both tables), we will need to look for data points in its vicinity. In this illustrative example we defined two spatial neighborhoods, $\mathcal{N}_o^{k_1}$ and $\mathcal{N}_o^{k_2}$, with $k_1 < k_2$. The first spatial neighborhood of our location $\left(\mathcal{N}_{(44,39)}^{k_1}\right)$ is represented by the observations in light blue, while its second (larger) spatial neighborhood $\left(\mathcal{N}_{(44,39)}^{k_2}\right)$ is composed by the observations in light and dark blue. The pre-processing phase is the iterative process of: (i) selecting a location (e.g. $(44, 39)$); (ii) define its neighborhoods; (ii) calculate the values of predictor variables using the target variable values inside these neighborhoods; (iv) select a different location and repeat the process, until all locations are mapped.

location	$\mathcal{Z}_o$
(44, 39)	390.89
(45, 39)	410.15
(45, 38)	400.07
(55, 42)	780.45
(25, 32)	800.34
⋮	⋮

Table 3.1: Spatial Data

location	$\mathcal{Z}_o$	$\overline{z}(\mathcal{N}_o^{k_1})$	$\overline{z}(\mathcal{N}_o^{k_2})$	$\cdots$
(44, 39)	390.89	405.11	597.75	$\cdots$
⋮	⋮	⋮	⋮	$\ddots$

Table 3.2: Regression Data

Using this pre-processing method we are able to build a data set that can be used to obtain models that solve the predictive task show in Equation 3.9.

In summary, our proposal for the spatial imputation problem using a spatial data set $\mathcal{Z}$ consists on: (i) use these data to build a new multiple regression data set where the target variable is the value of Z on a location and the predictor variables are calculated using the values in the vicinity of this location (an example are the variables mentioned in Equation 3.9 but others could be used); (ii) use this new data set to build a regression model with some existing algorithm; (iii) apply this model to locations where the value of the target variable is unknown.

3.4 A Concrete Application - Image Inpainting

The hypothesis driving our proposal can be described by the following assertions:

- there are regions that are far from each other from a geographical perspective and yet may have similar spatial dynamics in terms of the variable of interest;
- this similarity can be exploited by standard multiple regression models provided good descriptors of the spatial dynamics within a neighborhood are provided to the models as predictor variables;
- the potential use of this extra information (from distant regions) by the models will increase their predictive accuracy on spatial interpolation tasks.

This section describes a real world application used to validate our hypothesis. The application consists on filling in missing pixels on a image. An image can be seen as a spatial data set, given that each pixel has a different location in a system of Cartesian coordinates. At each location one or more values may be measured. In our problem it is a single degree of gray (a value in the interval $[0, 255]$) that is measured. In the research area of image processing this type of problems are referred to as "image inpainting" [Agrawal et al., 2010; Bertalmio et al., 2000; Shih and Chang, 2005]. The term "inpainting" has its origin in the manual task of restoring damaged paintings and/or photos by professional restorers [Bertalmio et al., 2000]. Digital inpainting is a relatively new research area with the goal of developing tools that automatically restore damaged images. Examples of damages include: noise (missing pixels caused by some equipment failure), unwanted objects (persons, cars, red-eye, etc.), logos, stamps, scratches (old pictures), etc.

Hays and Efros [2007] classify the image inpainting techniques in two main groups. The first uses additional information besides the source image, in an attempt to accurately reconstruct the damaged region; e.g. using a sequence of frames in a video [Irani et al., 1995] or multiple photographs of the same physical scene [Hays and Efros, 2007; Snavely et al., 2006]. The second group, uses only the information provided by the source image to restore the damaged region. The most common techniques apply some form of interpolation of the surrounding pixels (e.g. [Bertalmio et al., 2000]). Another example of this latter approach is the work by Shih and Chang [2005] that proposed an algorithm to restore a

(a) Dog face (260x222) (b) Coliseum (320x240)

Figure 3.1: Original pictures.

damaged region in a Chinese painting using several layers (based on the colors difference algorithm). Our approach also belongs to the latter group that only uses information from the given image.

Since the target application of this work is the repairing of images, we have also compared our approach against one successful implementation of an image inpainting algorithm based on the "exemplar based approach" [Agrawal et al., 2010; Bertalmio et al., 2000]. In our experiments we have used an open source implementation[1] of this inpainting algorithm.

Figure 3.1 shows the two original images that were selected to evaluate and compare our proposal. The first picture (Figure 3.1a) is a dog face with 260x222 pixels, and the second picture (Figure 3.1b) is the Coliseum of Rome, with 320x240 pixels. The pictures are rather different in several aspects, though both are described by a degree of gray in the $[0, 255]$ interval. Based on these images we will generate several data sets with an increasing number of these original pixels removed for evaluating the alternative imputation methodologies.

[1]Publicly available at `http://sourceforge.net/projects/imageinpainting`

3.5 Experimental Evaluation

3.5.1 Experimental Methodology

The main goal of our experiments is to check the validity of our proposal for spatial interpolation. We have carried out an extensive set of experiments under different conditions. To ensure a fair comparison between all the spatial interpolation models we have considered several setups in terms of the amount of missing pixels. Namely, we have created 9 different data sets from each original image (Figure 3.1) with an increasing number of pixels being randomly removed[2]: 10%, 20%, $\cdots$, 90%. Moreover, to ensure the statistical significance of the results we have repeated this random selection 10 times for each of the 9 settings. This means that we have compared the models on 180 different data sets generated from the two original images.

For each of these 180 different data sets, formed by the known gray levels of a sub set of pixels, we have compared different alternatives in the task of forecasting the target variable (degree of gray) at the missing pixel locations. The predictions were compared against the true values (Figure 3.1) using the Mean Absolute Error metric,

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{z}_i - z_i| \tag{3.10}$$

where, n is the number of missing pixels, $\hat{z}_i$ is the level of gray predicted by the models, and z_i is the real value according to the pictures in Figure 3.1.

In all setups the alternative models were given the same exact pixel data to ensure a fair comparison. Still, for the different variants of our approach that were tried, we had to develop a regression training set from this pixel data. This regression training set was created using the formula given on Equation 3.9 (cf. Section 3.3).

All data, code and extra results are provided in a web page [3] to ensure that our work is replicable.

[2] We should remark that the values at these locations were actually removed, i.e. set as unknown, and not set as white pixels as the graphical representations of the data sets we will see later, may indicate.

[3] `goo.gl/hRBMd`

3.5.2 Models

Our methodology is based on the use of a regression algorithm to obtain the models that are then used to carry out the spatial imputation of unknown values. In order to fully test our ideas we have selected a diverse range of modeling approaches to test our hypothesis. The idea was to confirm its validity independently of the technique used to forecast. In this context, we have used the following regression algorithms:

Regression Trees (RT) - a regression tree (e.g. [Breiman, 1984]) based on the R package `rpart` [Therneau and port by B. Ripley., 2009]. In our experiments we have used an interface to the `rpart` function provided in package `DMwR` [Torgo, 2010] and have tried 4 different variants by using the parameter `se` that controls the level of pruning with values: 0, 0.5, 1 and 1.5.

Support Vector Machines (SVM) - an implementation of SVMs (e.g. [Cristianini and Shawe-Taylor, 2000]) available in the R package `e1071` [Dimitriadou et al., 2009]. We have used 6 variants of the parameters *cost* that represents the penalty associated with errors and the parameter *gamma* which the used radial based kernel. For the parameter *cost* we used the values: 1, 10, 100 and for the parameter *gamma* the values: 0.1 and 0.5.

Random Forest (RF) - an implementation of random forests [Breiman, 2001] available in the R package `randomForest` [Liaw and Wiener, 2002]. We have used 3 variants of the parameter *ntree* that controls the number of trees in the forest (ensemble), with the values: $500, 1000$ and 1500.

Regards the competitive approaches for spatial imputation we have selected a series of techniques that are a good representation of the state of the art on this area:

Distance Interpolator (DI) - a simple baseline method that uses the mean value of a circular neighborhood region. We have considered 3 neighborhood sizes: 10, 30 and 50.

Inverse Distance Weighted Interpolator (IDW) - a variation of the previous method that uses the weighted average value within the neighborhood region as the approximation for the unknown location. The weights are inversely proportional to the distance. We have considered the same neighborhood sizes as in DI.

Ordinary Kriging (OK) - we have used an implementation of this method available on

the R package `automap` [Hiemstra et al., 2008]. The implementation in this package automatically selects the best parameters for the kriging method, including the neighborhood size and the function used in the calculation of the semivariograms (it considers spherical, exponential, Gaussian and two variants of the Matern family). To limit the search space, in our experiments we have set the maximum neighborhood size to 90.

All the used tools are freely available in the R software environment [R Development Core Team, 2010], which ensures easy replication of our work.

3.5.3 Results

Figures 3.2 and 3.3 summarize the results obtained by all alternatives using the experimental settings described before. Each bar represents the estimated MAE value averaged over the 10 repetitions of a model variant on each of the 9 data sets. These 9 data sets are images with an increasing number of pixels randomly removed from the respective original image. The different approaches are presented in 6 groups. The first group is composed by the base line distance interpolator (DI) approach, obtained using the 3 selected neighborhood sizes: 10, 30 and 50 (the β in Equation 3.1). The second group contains 3 similar variants but this time using the IDW technique with the same spatial neighborhood sizes. Then we have all the parameter variants of regression trees (RT), SVMs and random forests (RF). The last group includes the ordinary kriging approach, whose parameters are automatically tuned by the used software package.

The results of Figures 3.2 and 3.3 show an overwhelming advantage of our approaches when compared to these state of the art methods. In particular, both the SVM and RF variants achieve remarkably good scores, although even with the simple RT approach the results are superior. The typical error of our approaches is around 5 in a scale of $[0, 255]$ gray levels. It is also remarkable that even at the highest level of noise (90% of the pixels removed), the scores of our approach are competitive with the existing methods when given a data set with only 10% of the pixels removed. These experiments provide clear evidence of the advantage of: (i) using more sophisticated function approximates; (ii) using more elaborated information concerning the spatial dynamics through spatial indicators; and (iii) allowing the use of data from distance points in space provided the regression models find this useful in terms of accuracy. Another noticeable observation

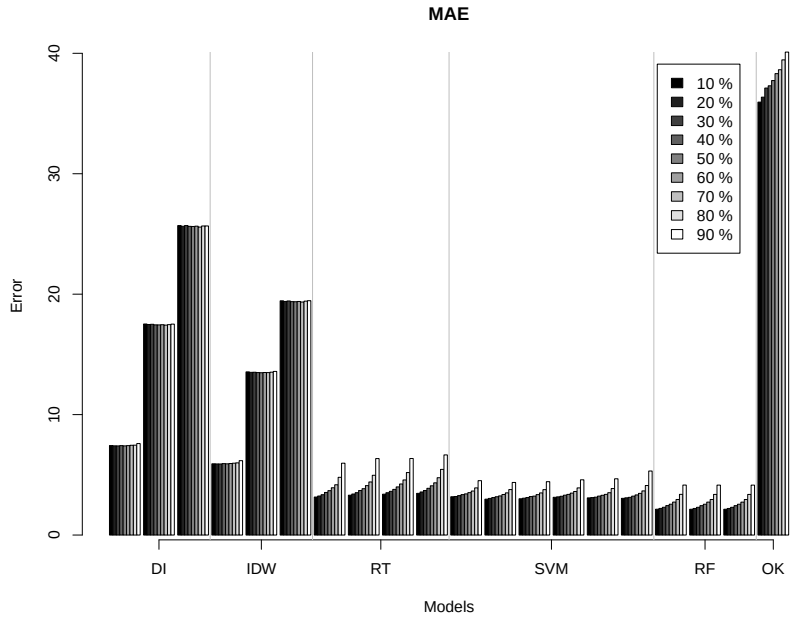


Figure 3.2: Estimated MAE of the different approaches for the Figure 3.1a.

is the surprisingly bad scores obtained by the used ordinary kriging method, which was unable to beat even the simple DI variants. This may indicate that the automatic tuning provided by the used software package may not be adequate for all situations and that these particular problems could require a more careful hand-tuning of the kriging parameters. Our approach, however, did not require any tuning at all, and it may even be the case that with different variants of our spatial indicators, for instance through the use of some feature selection algorithm, the performance could be further improved.

In order to better understand what the methods are doing in terms of approximating the original image, we have selected one of the ten repetitions and represented graphically both the original data and the approximations provided by the competing methods. These results are show on Figures 3.4 and 3.5, respectively. The first row of graphs shows the 9 original data sets with an increasing number of pixels removed. The remaining rows show the approximations provided by the predictions of each of the alternative approaches. Regards our approaches we have selected the best approximations for each algorithm, which were a regression tree model (RT) with $se = 0$, the SVM model with the $cost = 1$ and $gamma = 0.1$, and the random forest (RF) model with $ntree = 500$. These graphs illustrate the remarkable job that our approaches are able to achieve in terms of recovering the original image, even at very high levels of noise. The quality of the pixel imputation

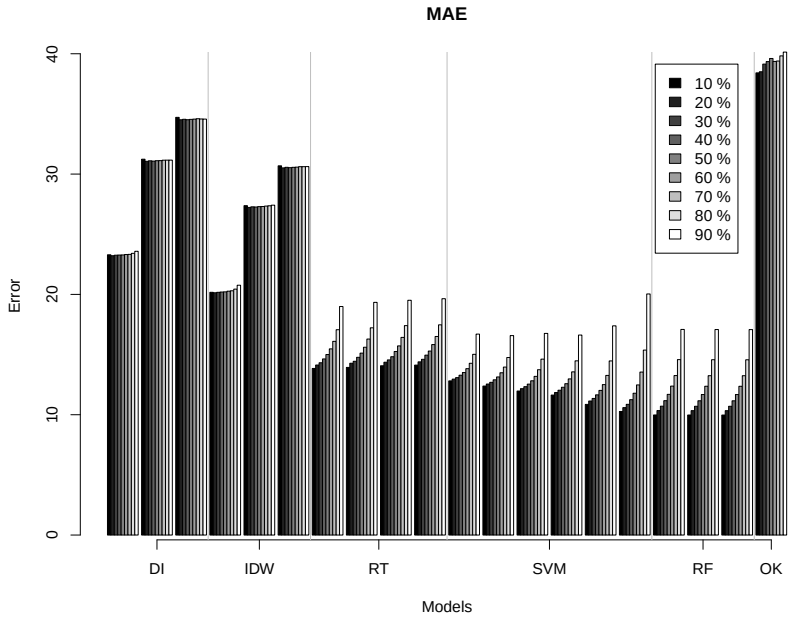


Figure 3.3: Estimated MAE of the different approaches for the Figure 3.1b.

even with 90% of the pixels removed is impressive.

With the goal of evaluating the performance of the models in different conditions we have carried out another experiment where we have artificially created larger areas with no pixels available. Namely, starting from an image with 60% of the pixels randomly removed ($DS_{60\%}$, Figure 3.4 first row sixth column), we have artificially added 24 circular holes with different radius (8, 10, 15 and 20). On these 24 regions all pixels were removed. This new data set was denominated DS_{holes}, and you can see the resulting image on Figure 3.7a. We have applied all competitive methods to this data set and the resulting MAE values are show on Figure 3.6. Once again we have observed a marked superiority of our approach. We should note that this is a situation where the ability to use data from faraway vicinities should give an extra advantage to our methods when trying to approximate pixels inside the circles. These experiments confirm this intuition as our models are able to provide reasonable approximations even on these areas where limited information on nearby pixels is available, as it can be confirmed in Figure 3.7 where we have the pictures recovered by each alternative method. We should remark that for some neighborhood sizes it may happen that no value is available in the vicinity of a target location. On these locations we have allowed the DI and IDW to dynamically increase the neighborhood size until a

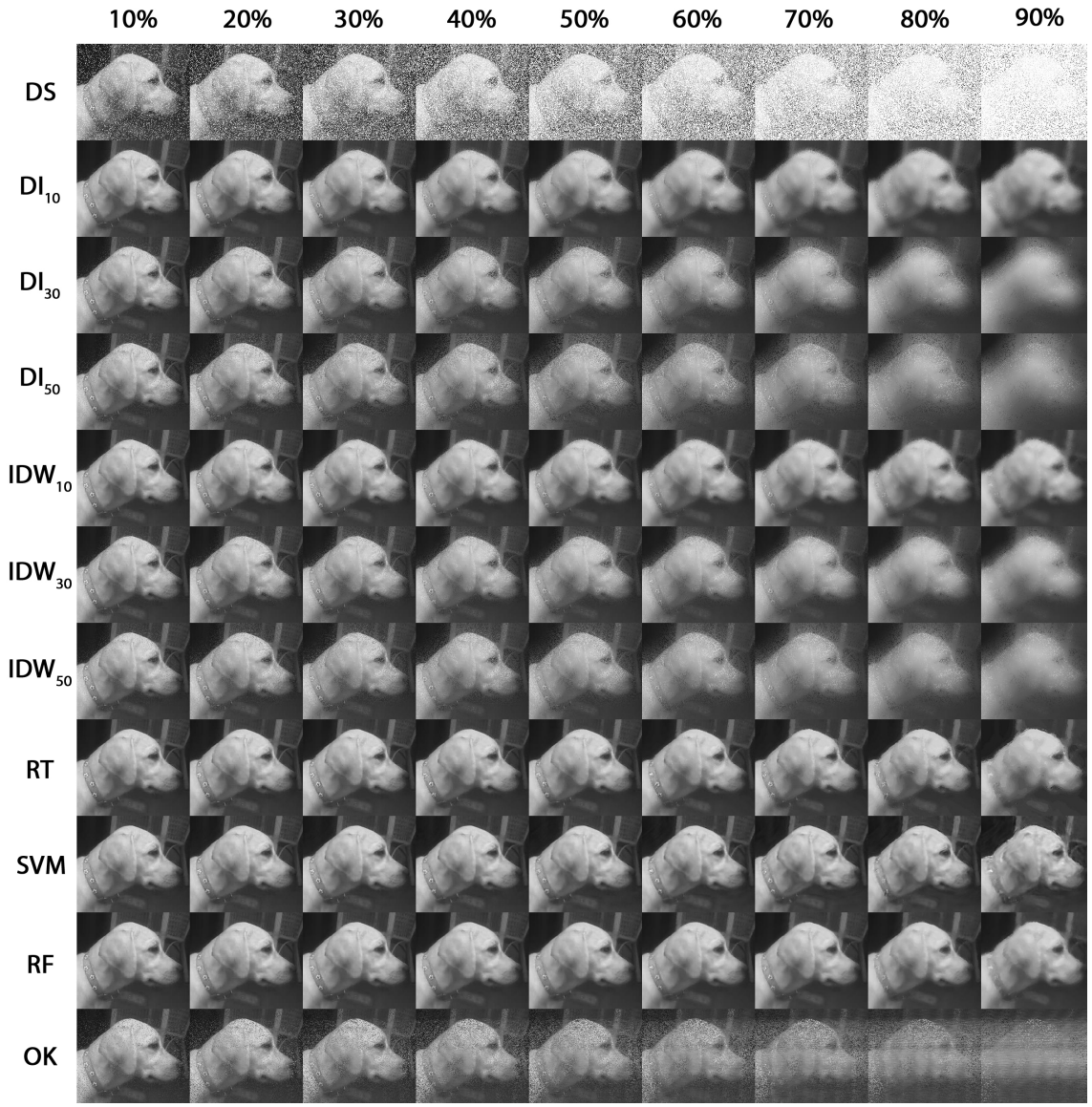


Figure 3.4: The used data sets and the resulting approximated images for Figure 3.1a.

reasonable amount of values exist to calculate their averages. On kriging this adaptation should be carried out automatically by the parameter tuning of the used function.

3.5.4 Comparisons with Inpainting Algorithms

As mentioned before, the problem we are addressing is named image inpainting within the image processing research area. We have compared our best variant (RF $ntree = 500$) to one of the most common methods in image inpainting (see Section 3.4). We were not able to compare these two techniques on the 9 data sets with increasing percentage of removed

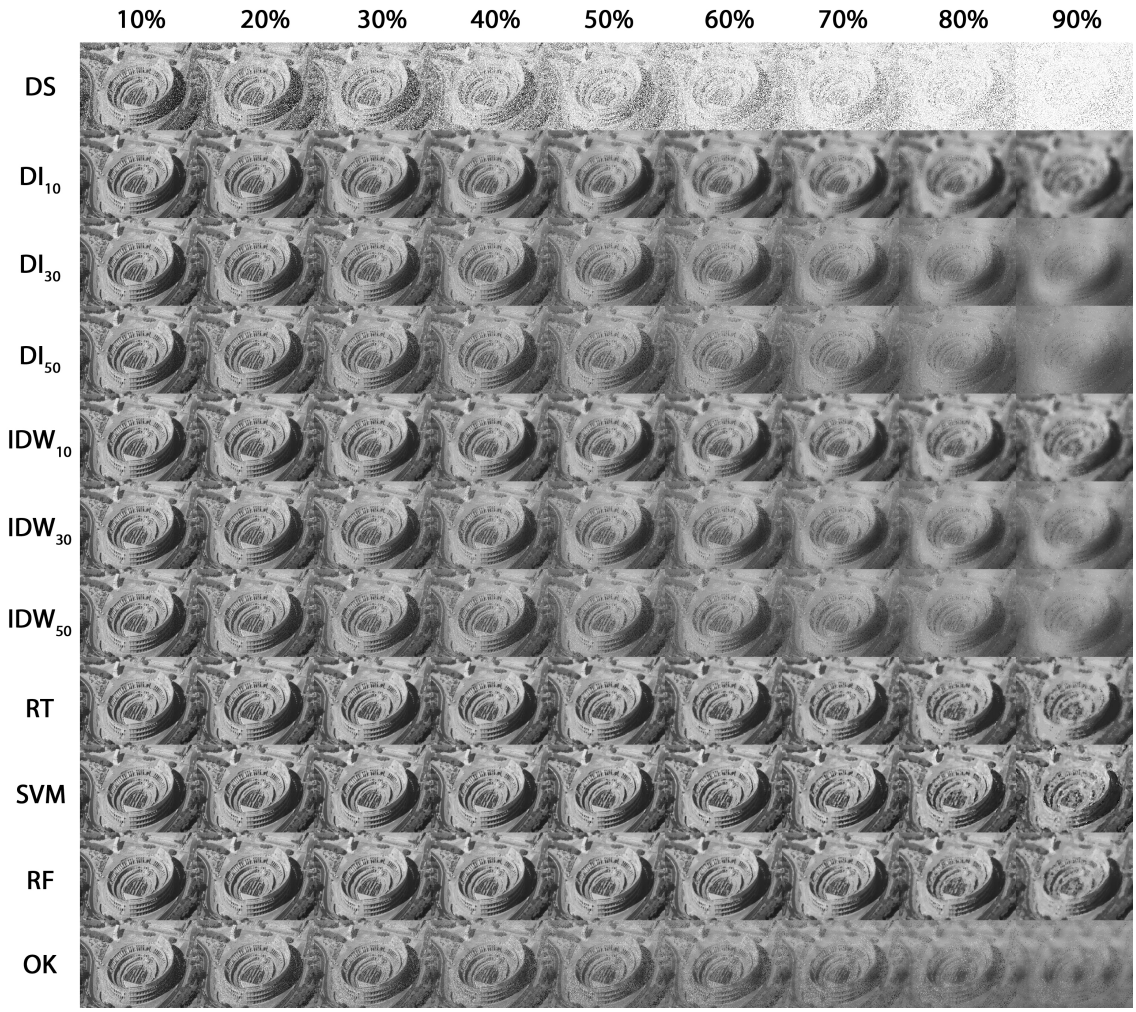


Figure 3.5: The used data sets and the resulting approximated images for Figure 3.1b.

pixels from Figure 3.1a, because the used inpaiting software crashed on data sets with too many unknown pixels. In this context, we were only able to collect results for the $DS_{10\%}$ and $DS_{20\%}$ data sets.

Figure 3.8 shows the results of this comparison. We show the original data sets; the approximations provided by two variants of the inpainting algorithm[4]: the fast implementation (Figures 3.8b and 3.8f) and the standard implementation (Figures 3.8c and 3.8g); and the results of the random forest in Figures 3.8d and 3.8h. Although the inpainting algorithm is able to achieve similar results on the data set with a lower level of unknowns (particularly in the standard implementation), in the data set with 20% of removed pixels we already see a marked advantage of our approach.

[4]To apply our missing dataset variant to the inpaint software we need to convert the missing pixels to the RGB green color.

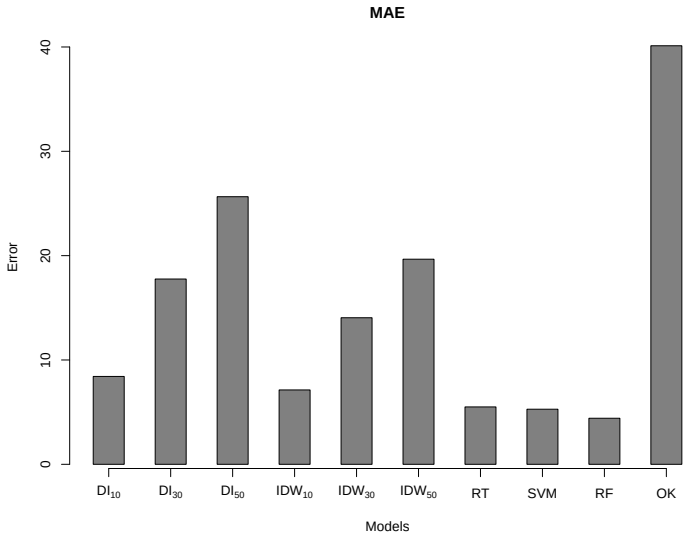
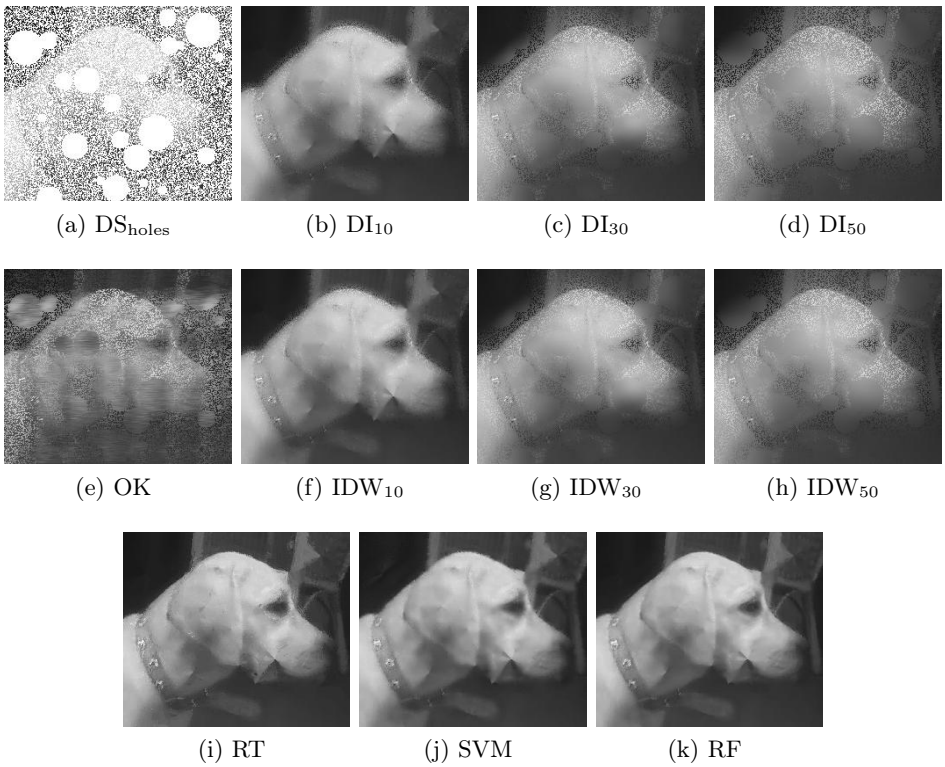


Figure 3.6: Mean Absolute Error on the DS_{holes} data set.

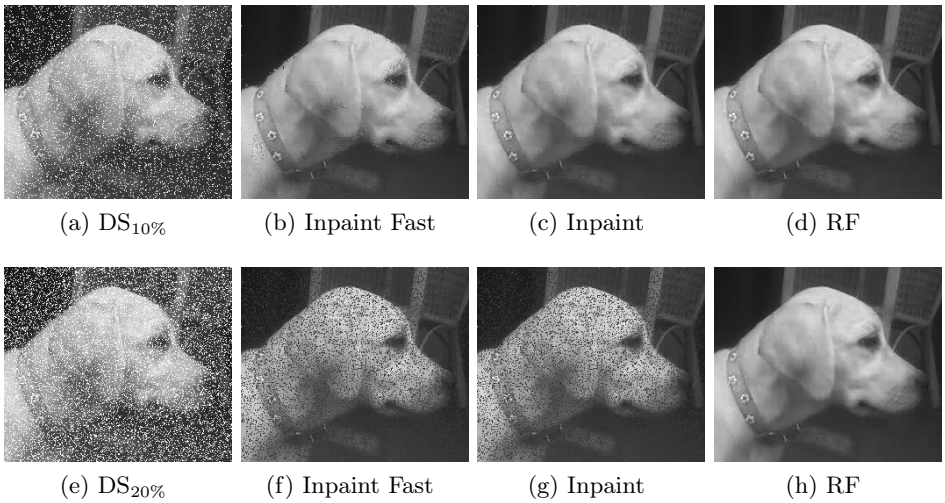


(a) DS_{holes} (b) DI_{10} (c) DI_{30} (d) DI_{50}

(e) OK (f) IDW_{10} (g) IDW_{30} (h) IDW_{50}

(i) RT (j) SVM (k) RF

Figure 3.7: The DS_{holes} data set and the approximated pictures of the methods.



(a) $DS_{10\%}$ (b) Inpaint Fast (c) Inpaint (d) RF

(e) $DS_{20\%}$ (f) Inpaint Fast (g) Inpaint (h) RF

Figure 3.8: Random Forest vs Inpaint Technique

3.5.5 The Usage of Data from Faraway Regions

One of the distinguish factors between our approach and the state of art spatial imputation methods, is the fact that our approach allows the use of data from faraway regions provided these regions have similar neighborhood dynamics. We believe that this additional information provides a significant advantage in favor of our proposed technique, for the task of spatial interpolation. To support this claim we selected our most interpretable model, regression trees, and learned two models, one for each picture in analysis (see Figures 3.1a and 3.1b). Regression trees are learned by trying to group training cases that have similar values on the target variable and which share some properties in terms of the predictor variables. The goal of this experiment was to analyze which training cases were put on each leaf of the learned trees. If our hypothesis is right we should expect to observe leaves with training cases that are distant apart from the perspective of the Cartesian coordinates of each picture. There are two possible outcomes for this experiment: in the first, i) the leafs have only cases from the same spatial neighborhood, which would be in accordance with the first law of geography and the approaches followed by the standard state-of-art methods; in the second ii) the leafs have cases from multiple spatial locations, supporting our claim that the model find useful to use data from faraway regions.

We have carried out this experiment by creating two regression data sets with the methodology described in Section 3.3, one for each photo. Using each of these regression data sets we have obtained a regression tree and have analyzed the cases that were put by the

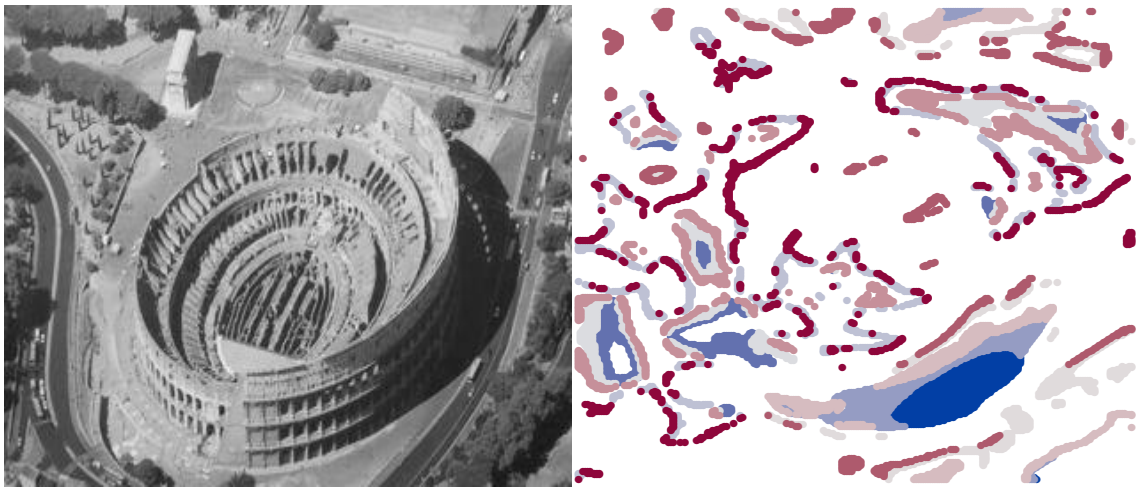
(a) Dog Face (b) Leafs Map

Figure 3.9: Leaves Map - Dog Face (see Figure 3.1a)

models on each leaf. Both trees were learned with the parameter $se = 1.5$, to generate small trees and avoid overfitting.

Figures 3.9 and 3.10 show the results of this experiment. On each figure we shows the original picture and a colored map where each color represents the pixels on a tree leaf. To facilitate the visualization of the results the maps only show the geographical location of the pixels in the 10 leaves with a higher number of cases. As we can see, in both cases the leaves contain cases from faraway regions. This provides clear evidence that these regression models find similarities, both in terms of the target variable values and in terms of the descriptors of the neighborhood spatial dynamics, between pixels that are distant from each other. Given the predictive accuracy of these models we claim that the usage of information from faraway regions is providing advantages to these models in terms of spatial interpolation.

In the Coliseum picture the usage of data from faraway regions is even more marked as can be observed in Figure 3.10. However, we can also observe a leaf (darker blue region) where all cases belong to a nearby vicinity. This means that our approach does not force the usage of data from distant regions. This decision is left to the criteria guiding the model construction which is typically related to some predictive accuracy optimization process.



(a) Coliseum (b) Leafs Map

Figure 3.10: Leaves Map - Coliseum (see Figure 3.1b)

3.6 Conclusions

This chapter describes a novel approach to the problem of spatial interpolation. Our general methodology is based on the idea of transforming this problem into a multiple regression task and then applying standard algorithms to a data set that is constructed from the original spatial data using a series of spatial indicators designed to better describe the spatial dynamics of the variable of interest. The key distinctive feature of this methodology is the data that is used to obtain the approximations of the unknown values of the variable of interest. Existing state of the art methods use only values within a certain neighborhood of the target location for which we want an estimate. Our proposal is based on the assumption that other distant vicinities may be used provided they show a similar spatial correlation pattern. The decision to use this extra data is left to the the optimization process of the regression models. With the goal of improving the discovery of similar neighborhoods we have also introduced the notion of spatial indicators. These are features constructed from the original data that try to provide useful information on the spatial correlation dynamics within a neighborhood. Their goal is to help the models in uncovering similarities among different regions of the space.

Although the described methodology is a general spatial imputation method, in this thesis we have tested it on a particular task with strong impact in several application domains:

image repairing. We have tested and compared our method under different setups in terms of missing information on the given images. On all setups we have observed a strong advantage of our approach that has achieved impressive results in terms of recovering an image even at high levels of noise. These results are very encouraging and provide strong empirical evidence towards the advantages of our approach to spatial imputation.

Chapter 4

Sensor Network Prediction through Spatio-Temporal Indicators

In this chapter we propose a new technique to improve short term prediction in sensor networks. The vast majority of the work developed in spatio-temporal sensor networks uses just one dimension of the problem. Our approach is based on the assumption that by incorporating both dimensions (spatial and temporal) will improve the predictive models accuracy. Our approach is based upon the definition of a spatio-temporal neighborhood. Within this neighborhood a series of spatio-temporal indicators are applied to provide information to the models on the spatio-temporal dynamics of the target variable. This approach is a natural follow-up of the approaches taken in the previous chapters, to both space and time. Once again our approach to tackle these problems is based on data pre-processing techniques that try to provide the models with useful information on the dynamic properties of the target variable. This approach was evaluated using real world wind speed data collected at wind farms in US. Our extensive experiments show that our proposal has clear advantages in most setups over a series of alternatives.

4.1 Introduction

Spatio-temporal data mining focus on spatial applications that evolve over time, e.g. traffic management, route planning, electric power systems, GPS based systems, water

distribution networks, etc. Most of the research effort on mining geo-referenced data has been based on segmenting the analysis of the problem; analyzing only the spatial information ignoring the temporal characteristics of the data, or otherwise, and treat the problem as a time series task. More elaborate approaches tries to combine two isolated approaches. However, in our opinion treating both dimensions separately significantly limits the understanding of the problem. The majority of the geographic phenomena evolve over time, so both spatial and temporal correlation information are key points in this analysis [Yao, 2003]. In this thesis we propose an approach that uses both dimensions for spatio-temporal forecasting tasks.

Spatio-temporal data mining is an emerging research area. It can be considered as a natural evolution of two established and well studied research areas: temporal data mining (time series) and spatial data mining. The main challenges in spatial domain are: the data representation (point, lines, and polygons), that are more complex than non-spatial representation; and the influence/correlation between spatial objects. In the temporal domain the challenge is to identify patterns on sequences of spatial objects, which is more complex given the complexity of these objects. The need to investigate both "spatial" and "temporal" relations at the same time complicates even further the data mining process. Most of the spatio-temporal application domains have the spatial and temporal characteristics explicitly defined. However, in some domains, like sensor networks, the spatial information is usually not embedded in the data. This information can be extracted from the domain, based on the sensor locations. This class of applications needs an extra step in the data preparation phase. In this chapter we address this sub-group of the spatio-temporal applications - data originated from sensor networks. This group of applications has been receiving substantially more attention from the scientific community, pushed by the growth of sensor networks applications in key areas of our society, like: water distribution network, power distribution network, power generation, traffic control, TV/Internet providers, etc.

The main goal of our work is to improve the ability to forecast sensor network data by using a spatio-temporal approach. Data mining research is frequently application-oriented. Our work was also be driven by a concrete application - forecasting wind speed at different locations of a wind farm, in US. Namely, our target was to forecast the next 2 hours wind speed. Although driven by a particular application, our proposed method is generic and

can be applied to a wide range of spatio-temporal forecasting scenarios.

In the Section 4.2 we describe the recent advances in spatio-temporal data mining research. In Section 4.3 we describe our proposed formalization of the prediction problem that includes the definition of spatio-temporal indicators. In Section 4.4 we describe the real world application focus of this chapter, wind speed forecasting. Section 4.5 describes the experiments, namely the data and the experimental methodology that were used; we also present and discuss the results of our experiments. In Section 4.5.4 we analyzed 20 parameters variants for the SVM model. And in the Section 4.5.5 we analyzed the performance of the best model (random forest with $ntree = 500$) with four new spatio-temporal neighborhood sizes, while on Section 4.6 we draw the conclusions of the work and describe our future research agenda.

4.2 Spatio-temporal Data Mining

According to Andrienko et al. [2006], "spatio-temporal data mining is an emerging research area, dedicated to the development and application of novel computational techniques for the analysis of large spatio-temporal databases".

For Koperski et al. [1998], "spatio-temporal data mining refers to the extraction of implicit knowledge, spatial and temporal relationships, or other not explicitly stored in spatio-temporal databases".

Spatio-temporal data mining is a sub-field of data mining addressing data analysis tasks related with spatio-temporal applications. Spatio-temporal applications are domains where data is collected on different spatial (geospatial) locations at different points in time.

Despite being a recent research area, there are already a considerable number of interesting contributions in spatio-temporal data mining. One of the application domains to which spatio-temporal data mining can be applied is on the identification of patterns on moving objects. The basic assumption is that objects follow the same approximate routes over regular time intervals, e.g. people wake up at the same time and follow approximately the same route to their work everyday. Cao et al. [2005] proposed an approach that transforms the objects trajectories (GPS movements) into a sequence of events using a line segment approach, and applied a mining algorithm based on a sub-string tree, looking for patterns

that identify interesting movement behavior. These patterns could be used to forecast the next location of objects. Mamoulis et al. [2004] proposed other approach to the same problem. They transformed the movement of the objects in a sequence of events using a grid approach, and proposed a fast mining algorithm to identify the maximum number of patterns and an index structure for efficient execution of spatio-temporal patterns queries.

Compieta et al. [2007] developed a system for exploratory spatio-temporal data analysis within in a visualization tool. The system has two main components, the spatio-temporal engine miner and the visualization engine. The spatio-temporal engine miner is based on a variation of the Apriori algorithm to generate spatio-temporal association rules. The visualization engine uses Google Earth to add a layer developed in Java3D for visualization of the patterns identified by the spatio-temporal engine miner. The system has a strong visual component, that enables high levels of interaction in the analysis of the domain. The case of study adopted in this work was the data set of the Hurricane Isabel, which struck the US east coast in September 2003.

Ciampi et al. [2010] proposed a new technique that combines spatial clustering with trend discovery in spatio-temporal sensor network data. The technique is composed of two phases: (i) in the first an online process continuously takes snapshots of the current status of the sensor network, with the aim of generating a spatial cluster of each snapshot, and then storing the cluster information; (ii) the second phase consists in discovering trends based on the generated clusters, using a time window approach (only the most recent clusters are analyzed). This technique was tested using synthetic and real world data streams: the temperature measurements collected at Intel Lab and South America climate streams.

Yao [2003] classifies spatio-temporal problems in five main tasks: segmentation, dependency analysis, deviation and outlier analysis, trend discovery, and generalization and characterization. `Segmentation` is the process of classifying or creating clusters using the spatio-temporal data; `dependency analysis` is the task of finding association rules between the spatio-temporal objects; `deviation and outlier analysis` identifies spatio-temporal objects that deviate from the normal behavior in the data set; `trend discovery` is related to the prediction of the expected future value (regression) in spatio-temporal context, discovery of current trends, and with the discover of correlation between events;

and `generalization and chracterization` of the data that transforms it in order to compact the information or better describe the properties of the data.

In this thesis we are interested in spatio-temporal prediction tasks. Spatio-temporal prediction consists in trying to forecast the future value of the target variable for some location, based on information on past values of this variable at that same location as well as on nearby locations. The existing approaches for forecasting spatio-temporal data usually analyze each dimension (space and time) individually. For instance Cheng and Wang [2008] proposed an integrated spatio-temporal framework for forest fire prediction in Canada. The proposed framework splits the problem in three prediction tasks: (i) use a time series arima model for the temporal prediction; then (ii) apply a recurrent neural network for the spatial prediction; and finally (iii) combine these two predictions with a linear regression model. Contrary to these approaches, our proposal in this thesis is to integrate in a single modeling task both the space and time dimensions.

4.3 Our Proposal - Spatio-temporal Indicators

The task being addressed in this chapter consists on trying to forecast the future value of a variable evolving in time on a certain geographical location, based on historical data of this variable collected on both this and other geographical locations, across a past time period.

If we forget the fact that we have measurements of the target variable on nearby locations and that there may exist spatial correlation between the values on these locations and the current target location, we would be facing a time series forecasting problem. The most common approach to time series forecasting using machine learning models consists in transforming the original problem into a multiple regression task, where the target variable is the future value of the series, while the predictors are previous past values of the series up to a certain p-length time window. For instance, using the time series shown in the Figure 4.1, and assuming we want to forecast its future value, we can transform this time series into a regression data set using the schema outlined in Table 4.1. This consists on iteratively applying the rule: `the target variable value is the next value of the series and the predictors are the previous` p `values of the series`. This trans-

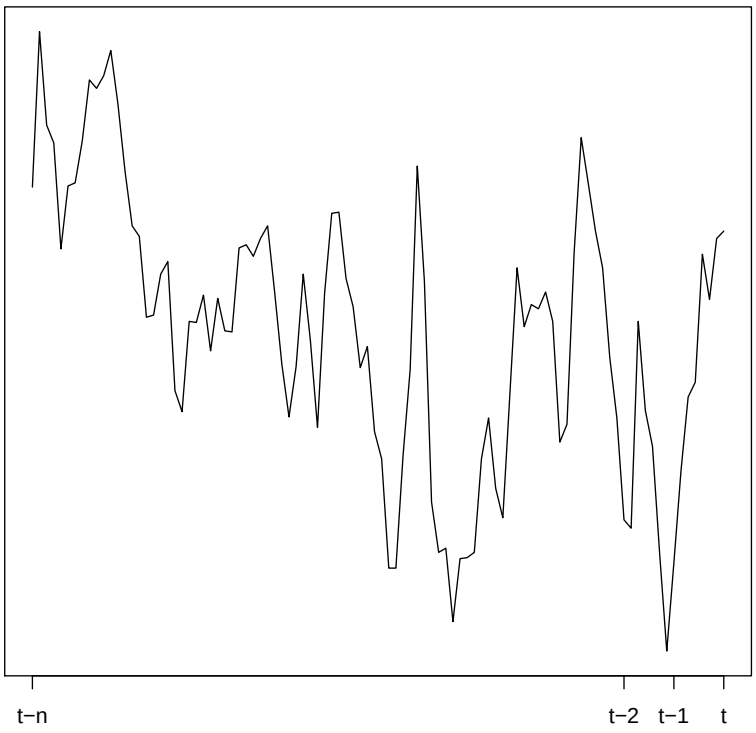


Figure 4.1: Time Series - time delay embedding

formation technique is usually known as time delay embedding [Takens, 1981]. The motivation for time delay embedding is the assumption that future values of the series depend at most on the past p values, with p being known as the embed size. The value of p needs to be determined and is clearly domain-dependent but Takens has shown [Takens, 1981] that with the correct embed size determined, we can approximate any dynamic system using this strategy. The idea behind time delay embedding is actually simple - we should provide the models with sufficient information for them to be able to uncover the mechanism generating the time series values. This mechanism is assumed to be non-random in the sense that future values of the series depend (in an unknown way) on the previous values of the series. Moreover, it is sensibly assumed that the length of this window of dependency on the past is limited in size.

The simplest form of using the idea of time delay embedding is to provide the models with the most recent values of the target time series as illustrated in Table 4.1. An improvement

target	predictors (e.g. last 10)		
W_t	W_{t-1}	$\cdots$	W_{t-10}
W_{t-1}	W_{t-2}	$\cdots$	W_{t-11}
W_{t-2}	W_{t-3}	$\cdots$	W_{t-12}

Table 4.1: Regression - time delay embedding

over this simple strategy is frequently used within financial forecasting. In this field it is frequent to also use as predictors what are known as `technical indicators`. According with the financial dictionary[1] technical indicators are "statistical information that is used to determine future trends in security prices and to make or recommend investment decisions based on those trends." These variables are nothing more than statistical summaries of certain properties of the time series. These properties include effects like tendency, acceleration, momentum and so on. Different indicators were developed to express these features of a time series. These indicators can be regarded as "sophisticated" descriptors of the recent dynamics of the time series we want to forecast. In a way they provide a synthesis of important properties of the recently observed values of a time series. Including them as predictors should provide the models with more information on the recent behavior of the time series.

The assumption behind our proposal is that the future value of the target variable at a location i depends not only on the previous past values of the variable at this same location, but also on the past values on nearby locations. This means we assume and try to model both the temporal and spatial correlation among the values of the target variable. Ignoring any of these two important forms of correlation seems limiting in terms of predictive accuracy. This is the case in the domain that has driven our work - wind speed forecasting. The future values of the wind speed depend not only on the recent past values at the same location but also on the neighboring locations. This spatio-temporal dependency is not particular to wind speed forecasting. Several real world domains have similar forecasting problems with the same type of spatio-temporal correlation. In effect, with the profusion of mobile computing devices with GPS capabilities, the demand for the analysis of spatio-temporal data is increasing at a very high rate.

Our approach is essentially a form of data pre-processing methodology. The key idea behind

[1] http://financial-dictionary.thefreedictionary.com/

our proposal is to try to develop predictors that are able to describe the spatio-temporal dynamics of the time series we aim to forecast. More precisely, we plan on mapping the concept of technical indicators used in financial forecasting to a spatio-temporal context. With this purpose we derive a series of spatio-temporal indicators that can be used as predictors in the task of developing forecasting models. Our assumption is that these extra information will provide the models with important characteristics on the recent spatio-temporal dynamics of the time series, which in turn will improve the model prediction accuracy. In this context, we plan to formalize the prediction problem in such a way that the future values of the target time series are predicted using not only previous values of the series and summaries of its temporal dynamics, but also with spatio-temporal indicators that summarize the dynamics of the series within the neighborhood.

The first question we need to address is how to describe the behavior of the time series within the neighborhood of the target location. Our proposal is based on the notion of spatio-temporal neighborhood. In this context, we need to define a function to calculate the distance between any two points in the space-time dimension.

In this work a point in space-time is the value of a variable (in our application the wind speed) at a time t in a certain geographical location. In Chapter 3 we have proposed the notation z_i to represent the value of a variable Z for a geographical location i. Here we will extend this notation by including the time dimension as an exponent. This means that z_i^k will denote the value of the variable Z on location i at time k. Let x and y be two points in space-time (i.e. two measurements z_i^k and z_j^l of the variable under study). We define the spatio-temporal distance between these two points in a similar way to Ming-yao et al. [2009], namely,

$$D_{x,y} = d_{i,j} \times \alpha + t_{k,l} \times (1 - \alpha) \tag{4.1}$$

where $d_{i,j}$ is the spatial distance between the locations of the objects (z_i^k and z_j^l), $t_{k,l}$ is the time distance between the objects, both the $d_{i,j}$ and the $t_{k,l}$ are normalized (divided by the respective maximum value) to have the values between $[0, 1]$, and α is a weighing factor between time and geographical distances.

The spatial distance can be calculated using a standard metric, like for instance the

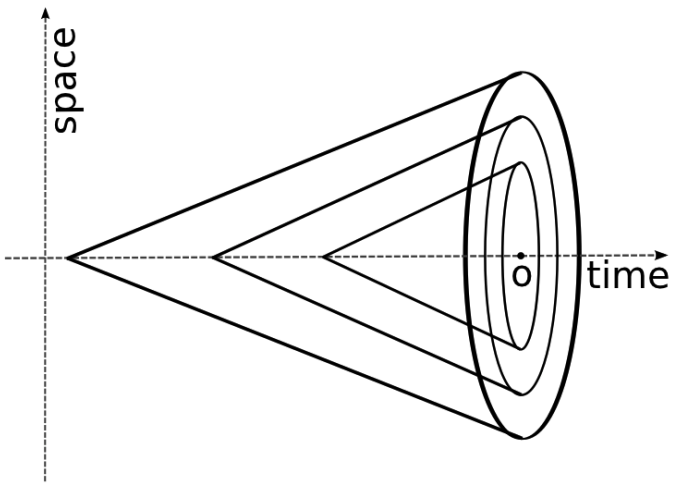


Figure 4.2: Defining spatio-temporal neighborhoods with different sizes.

Euclidean distance, or more sophisticated versions for geographical data like the great-circle distance [Bridson and Haefliger, 1999]. In our experiments we have use this latter alternative given that our data is geographically indexed. The time distance is simply the absolute difference between the two time tags in some adequate time unit (e.g. hours).

Having defined the spatio-temporal distance between two objects we can define the spatio-temporal neighborhood of a point o as the set of points within a certain spatio-temporal distance,

$$\mathcal{N}_o^\beta = \{k \in \mathcal{D} : D_{o,k} < \beta\} \tag{4.2}$$

where $\mathcal{D}$ is the available spatio-temporal data set.

Given the above definitions we can look at the spatio-temporal neighborhood of a point as a kind of cone within space-time. Different settings for α and β lead to cones of difference sizes as shown in Figure 4.2.

Each cone defines a neighborhood around a central location. These cones can be seen as defining the points in space-time that most influence the values of the target variable at that location. The cones can be regarded as the spatio-temporal equivalents of the idea of time-delay embedding. Increasing the size of the cone will increase the spatio-temporal embed size.

As we have mentioned before, in financial time series forecasting it is common to summarize the dynamics of a time series by means of technical indicators. Different indicators capture different properties of the dynamics of the time series. In this thesis we propose three examples of spatio-temporal technical indicators capturing three different and important properties of the spatio-temporal dynamics of the target variable: (i) the typical value; (ii) the spread; and (iii) the tendency.

The typical value is a centrality statistic that tries to estimate the most common value of a variable. In this case we are using the average value of target variable within the spatio-temporal neighborhood of the target location as the value of this spatio-temporal indicator, namely,

$$\overline{w}(\mathcal{N}_o^\beta) = \frac{1}{|\mathcal{N}_o^\beta|} \sum_{x \in \mathcal{N}_o^\beta} x \tag{4.3}$$

where $\mathcal{N}_o^\beta$ was defined in the Equation 4.2.

Other variation of a typical value is the weighted version, where the weights are inversely proportional to the spatio-temporal distance, more weight is given the "closer" location, namely,

$$\widetilde{w}(\mathcal{N}_o^\beta) = \frac{1}{|\mathcal{N}_o^\beta|} \sum_{x \in \mathcal{N}_o^\beta} w_{o,x} \cdot x \tag{4.4}$$

where $w_{o,x} = \frac{1}{D_{o,x}}$, $D_{o,x}$ was defined in the Equation 4.1 and the weights must satisfy $\sum_{x \in \mathcal{N}_o^\beta} w_{o,x} = 1$.

The second spatio-temporal indicator we propose tries to capture the notion of spread of the values within the spatio-temporal vicinity of the target location. We have used the standard deviation calculated within this neighborhood as the value of the indicator,

$$\sigma_z(\mathcal{N}_o^\beta) = \sqrt{\frac{1}{|\mathcal{N}_o^\beta|} \sum_{x \in \mathcal{N}_o^\beta} (x - \overline{w}(\mathcal{N}_o^\beta))^2} \tag{4.5}$$

where $\mathcal{N}_o^\beta$ was defined in the Equation 4.2 and $\overline{w}(\mathcal{N}_o^\beta)$ was defined in the Equation 4.3.

The third proposed spatio-temporal indicator was developed to describe the tendency of the target variable as it approaches the target location, e.g. is it increasing its values or decreasing them? The notion of tendency can be captured by the ratio between two averages calculated with different neighborhood sizes. If the value of the average with shorter neighborhood surpasses the longer average we know that the variable is on an upwards tendency as it approaches the target location, while the opposite indicates a downwards tendency. While originally this idea was developed for time series (i.e only the temporal dimension) we have imported this idea into the spatio-temporal dimension. The ratio between two spatio-temporal averages provides us with information on how the target variable values evolve in the space-time dimension. This ratio can be defined as follows,

$$\overline{W}_o^{\beta_1,\beta_2} = \frac{\overline{w}(\mathcal{N}_o^{\beta_1})}{\overline{w}(\mathcal{N}_o^{\beta_2})} \tag{4.6}$$

where β_1 and β_2 are two neighborhood sizes and $\overline{w}()$ is the average of the target time series values for a set of points in the neighborhood of o, defined in Equation 4.3.

A variation of this indicator can be easily obtained by using weighted averages of the values within the spatio-temporal neighborhood. If we set the weights to the inverse of the spatio-temporal distance to the point o we have the effect that "closer" (in spatio-temporal terms) points are given more importance within the averages,

$$\widetilde{W}_o^{\beta_1,\beta_2} = \frac{\widetilde{w}(\mathcal{N}_o^{\beta_1})}{\widetilde{w}(\mathcal{N}_o^{\beta_2})} \tag{4.7}$$

where $\tilde{w}()$ is the weighed average of target time series for a set of points in the neighborhood of o, defined in Equation 4.4.

Having defined a series of spatio-temporal indicators, our hypothesis is that they provide useful information for the target prediction task. In this context, given the goal of forecasting the value of the target variable for k time steps ahead at location o, we propose to tackle this problem using the following formalization,

$$
\begin{aligned}
W_o^{t+k} = f(&W_o^t, W_o^{t-1}, \cdots, W_o^{t-m}, \\
&\overline{w}(\mathcal{N}_o^{k_1}), \overline{w}(\mathcal{N}_o^{k_2}), \overline{w}(\mathcal{N}_o^{k_3}), \overline{W}_o^{k_1,k_2}, \overline{W}_o^{k_2,k_3}, \\
&\widetilde{w}(\mathcal{N}_o^{k_1}), \widetilde{w}(\mathcal{N}_o^{k_2}), \widetilde{w}(\mathcal{N}_o^{k_3}), \widetilde{W}_o^{k_1,k_2}, \widetilde{W}_o^{k_2,k_3}, \\
&\sigma_z(\mathcal{N}_o^{k_1}), \sigma_z(\mathcal{N}_o^{k_2}), \sigma_z(\mathcal{N}_o^{k_3}))
\end{aligned}
\tag{4.8}
$$

where $f()$ is the unknown regression function we are trying to model using a set of training data $\mathcal{D}$, m is the size of a temporal embed, while k_1, k_2 and k_3 (with $k_1 < k_2 < k_3$) are spatio-temporal neighborhood sizes.

We should note that this is simply one among many possible setups including spatio-temporal indicators as predictors. The decision of using 3 spatio-temporal neighborhood sizes was arbitrary and other setups could make more sense depending on the application. Still, this was the setup used in our experiments with wind speed forecasting.

Figure 4.3 and Tables 3.1 and 3.2 present an illustrative example of the spatio-temporal pre-processing technique proposed in this thesis. This example was applied in the context of wind speed forecasting, with the goal of forecasting the next step ahead wind speed for a certain location. Figure 4.3 and Table 4.2 represent a spatio-temporal data set to be transformed to a regression data set (Table 4.3). In this example we used two spatio-temporal neighborhoods defined by $\mathcal{N}_C^{k_1}$ and $\mathcal{N}_C^{k_2}$, where $k_1 < k_2$ (cones defined by the red lines in Figure 4.3). The spatio-temporal neighborhood $\mathcal{N}_C^{k_1}$ for the location C includes the values of wind speed measured in the turbines B, C and D, that are inside the spatio-temporal neighborhood defined by the Equation 4.1 (the smaller cone). The (larger) spatio-temporal neighborhood $\mathcal{N}_C^{k_2}$ also includes locations A and F, as well as values further back in time as shown by the larger cone in the figure. In our example where the goal is to forecast the next value of the wind speed for location C (i.e. W_C^{t+1}), the necessary pre-processing steps are: (i) for each time step t, check the next value of the wind speed at location C to fill in the column W_C^{t+1} in Table 4.3; (ii) define the neighborhoods, $\mathcal{N}_C^{k_1}$ and $\mathcal{N}_C^{k_2}$, of this point in space-time; (iii) calculate the predictor variables using the values of wind speed inside these neighborhoods and fill in the corresponding columns in Table 4.3; (iv) iterate to different values of t.

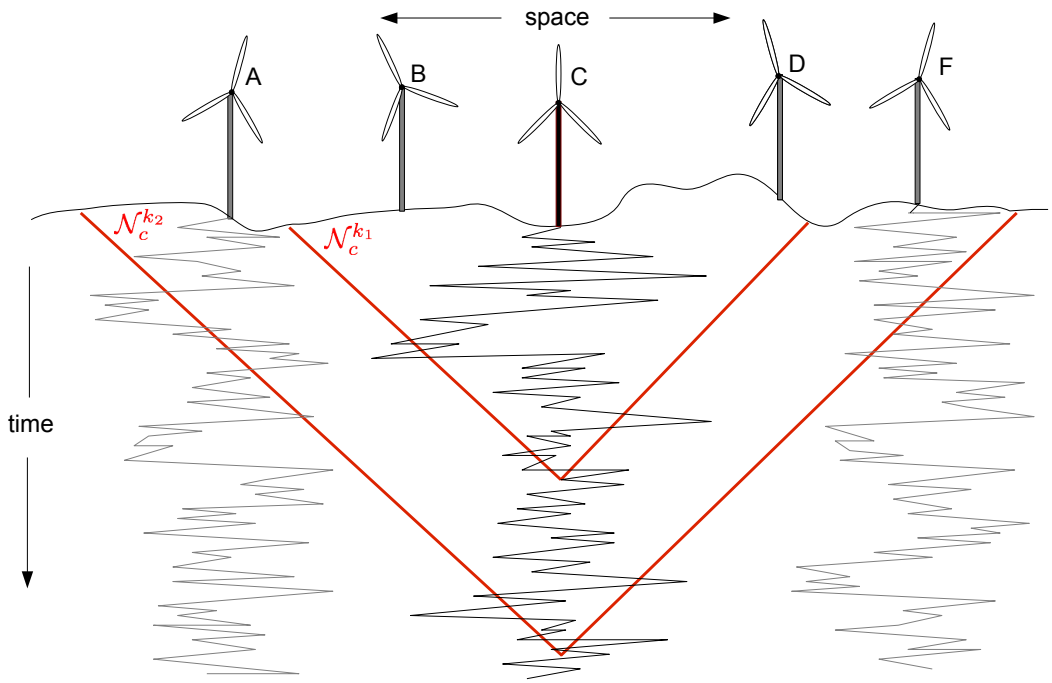


Figure 4.3: A spatio-temporal neighborhood for the wind farm problem.

time	A	B	C	D	F
1	700.00	390.89	410.67	400.32	800.23
2	⋮	⋮	390.89	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮

Table 4.2: The original spatio-temporal data.

4.4 Concrete Application - Wind Speed Forecast

The importance of wind power production is continuously increasing, as countries are looking for more sustainable alternatives for their power grid. Wind power generation is an excellent option given that it is a continuous resource of clean energy. The main drawback of this technology is the large variability in production, which makes almost impossible to rely solely in the wind energy. Generally, wind energy is used in conjunction with other types of technologies, like: thermal, hydraulic, natural-gas, etc. Wind power generation is also crucial in small remote autonomous locations, where it can be used as a fuel saver to reduce the operational costs. Some countries like the US [Joskow and Kahn,

time	W_C^{t+1}	$\overline{z}(\mathcal{N}_C^{k_1})$	$\overline{z}(\mathcal{N}_C^{k_2})$	$\cdots$
1	390.89	405.11	597.75	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Table 4.3: The generated regression data set.

2001], China [Zhao et al., 2011] and UK [Barthelmie et al., 2008] have electricity markets, which work similarly to an auction. Market participants rely on the expected future power production and on the market price to decide their bidding strategy. These expectations are usually considered for a short period of time, from a couple of hours to a day ahead. All these factors contribute to the crucial importance of having accurate prediction models of future power production. For wind energy this is even more relevant given its dependency on other sources of energy when the wind speed is low. Having an accurate forecast of the wind speed in the next hours is of key importance to estimate wind power production and define the best bidding strategy that maximizes the profit and avoids the penalties from missing delivering energy.

According to Alexiadis et al. [1999] wind power production is a function of the wind speed. This means that the accurate forecast of wind speed allows a better estimate of future wind power production. The wind is considered one of the most difficult meteorological parameters to forecast [Sfetsos, 2000]. The wind speed behavior is influenced by several factors like: the topographical properties of the land, the rotation of the earth, temperature, pressure, obstacles, the height of the anemometer, etc. [Kusiak et al., 2009; Sfetsos, 2000]. Lei et al. [2009] classify wind speed prediction models in four classes: physical models, conventional statistical models, spatial correlation models and artificial intelligence models. The physical models consider only characteristics like: terrain, obstacles, pressure and temperature to estimate the future wind speed. They generally have poor results in short term prediction. Conventional statistical models are based on time series techniques (ARMA, ARIMA, etc.) to forecast the future wind speed. Spatial models use the neighborhood information as predictors of the wind speed, usually applied to locations where the wind speed measurement is not available. Artificial intelligent models use historical data to obtain machine learning models that can be used to forecast the future wind speed. The method proposed in this chapter is an artificial intelligence approach that incorporates spatio-temporal predictors to forecast the future wind speed on any location.

The most frequent approach uses machine learning models to predict the expected wind speed considering as predictors the previously observed values of this wind speed [Kusiak et al., 2009; Mohandes et al., 2004; Sfetsos, 2000; Zhao et al., 2011]. Similar approaches are adopted by time series models [Kavasseri and Seetharaman, 2009]. All these approaches assume that the future wind speed depends on the recently observed wind speed on the same location. Given the fact that wind travels through the landscape this might be limiting for the models as they are being feed only with values from the same location for which a future prediction is required. These models ignore the spatial dependency that exists on this domain, where the wind speed at a certain location is clearly correlated with the wind speed at neighboring locations. There are some attempts to use the spatial information of the domain. In the work of Bilgili et al. [2007], they propose to use the monthly average wind speed at 4 neighboring locations as inputs for a neural network model to forecast the monthly average at the target location; and in the work of Alexiadis et al. [1999] that tries to identify the temporal relationship of the wind speed between spatial locations. They try to identify a pattern of the wind speed measured in two different locations, based on the travel time of the wind from one location to the other. The authors use this relationship to forecast the wind speed in a sub-sequent location. The main drawback of this approach is that it limits the neighbors used in the analysis and requires the information of the wind direction between the locations. In situations where this information is not available or is unreliable we can not use this technique.

In this chapter all the experiments were carried out using real world data publicly provided by the DOE/NREL/ALLIANCE[2]. The data consist in wind speed measurements from 1326 different locations at 80m of height in the eastern region of the US. The data were collected in 10 minutes intervals during the year of 2004. This wind farm is able to produce 580 GW, and each site produces between 100 MW and 600 MW. For our experiments we have selected two locations (A and B) as our targets in terms of forecasting the future wind speed. This selection was guided by the availability of a larger number of neighboring sites at these places. Figure 4.4 shows the geographical location of the data collection sites.

In the Figure 4.5 we can visualize the wind speed variation during the year of 2004 for one of the 1326 locations.

[2]http://www.nrel.gov/

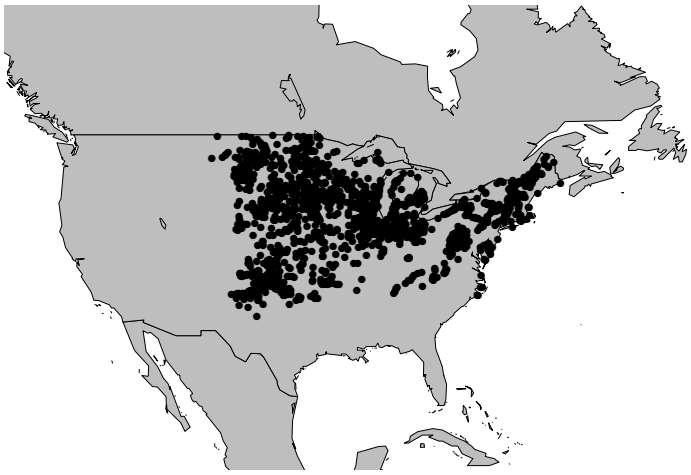


Figure 4.4: Wind Farm at Eastern US.

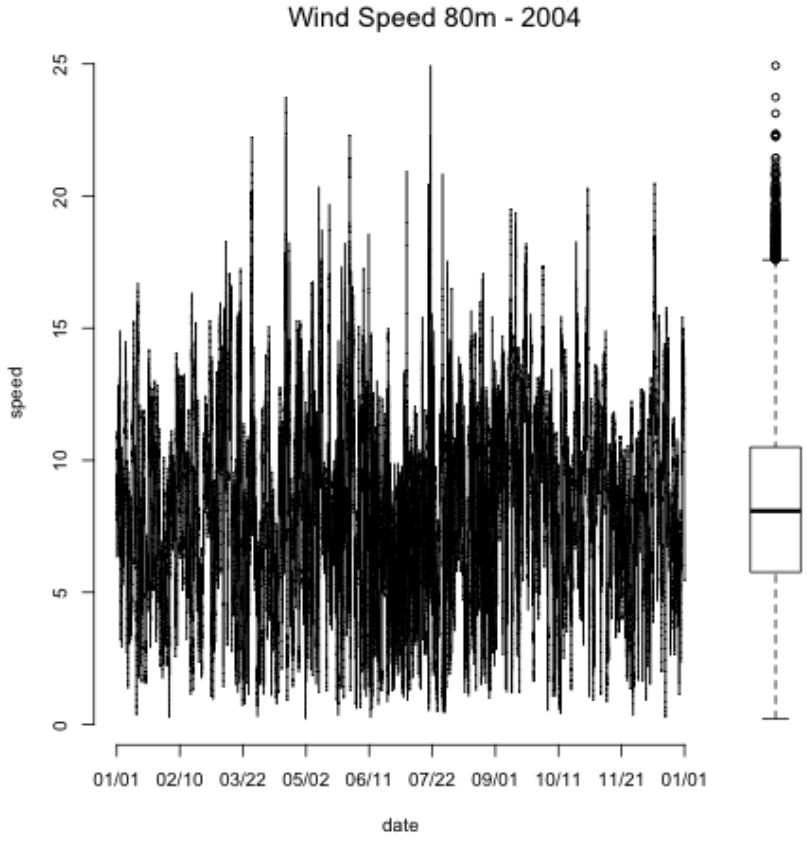


Figure 4.5: Wind Speed Variation

4.5 Experimental Evaluation

In this section we evaluate our proposed approach. The main goal of our experiments is to test the hypothesis that motivates this work: using information on the wind speed of nearby locations in recent time will improve the predictive accuracy of our models when forecasting the future wind speed at a certain location.

4.5.1 Experimental Methodology

With the goal of collecting experimental evidence towards this hypothesis we have designed an experiment where we have compared different models that tackle this prediction task using different predictors. Namely, we have compared our approach that includes spatio-temporal indicators as shown in Equation 4.8, with other approaches where the predictors do not include data from this spatio-temporal vicinity. In order to exclude eventual dependencies of the outcome of the experiments on the used modeling tools, we have repeated the comparisons using several learning algorithms with different parameter settings.

Each of the model variants that we will describe in Section 4.5.2 was applied to 6 different prediction tasks. These tasks have exactly the same target variable (the wind speed at time $t+2h$), but differ in the way they use the available past data to obtain the predictors used in the modeling task. One of these 6 tasks only uses data from the same spatial location, i.e. it only uses information from the past values of the wind speed measured on the site for which we want a forecast. This means that this task only considers the eventual time correlation among the values of the target variable, completely ignoring the spatial correlation. The other 5 variants use the formalization we have proposed in Equation 4.8, with different configurations for the sizes of the 3 neighborhoods. As we have seen these neighborhoods are cones defined by Equation 4.2. The formalization in Equation 4.8 uses three of these cones. An alternative way of defining a cone is by its maximum radius and its height from the base. This equivalent specification of the neighborhood is more intuitive in our application. For instance, the cone with maximum radius of 10km and height of 10 days, defines a neighborhood that for the current time uses data points that are at most 10km away from the target location, and goes back in time at most 10 days. Using this alternative specification of neighborhoods (cones) we can describe the remaining five

variants of the problem specification as follows: i) [50km, 10 days], [100km, 20 days] and [150km, 30 days]; ii) [140km, 10 days], [350km, 20 days] and [730km, 30 days]; iii) [75km, 10 days], [150km, 20 days] and [300km, 30 days]; iv) [100km, 10 days], [500km, 20 days] and [900km, 30 days]; and v) [150km, 10 days], [675km, 20 days] and [1200km, 30 days]. Regards the first variant using only data from the same location we have used exactly the same predictors as in Equation 4.8. However, all indicators are calculated using only the wind speed values of the target location, i.e. the spatial neighbors are ignored. It is like we were using a cylinder of spatial radius near zero, instead of the cones.

The predictions of the different model trials that we will describe were evaluated using the mean absolute error (MAE),

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{4.9}$$

where $\hat{y}_i$ is the predicted wind speed value (for $t + 2h$) for a true value of y_i.

With the goal of obtaining statistically reliable estimates of this error measure we have used a Monte Carlo simulation. The simulation was designed to provide estimates of the MAE at predicting the wind speed for two hours ahead of the different alternatives considered in our experiments. To increase the statistical reliability of the experiments we have repeated the process 10 times at randomly selected time points within the available data interval (10 minutes measurements throughout all 2004). For each of these 10 randomly selected time points, and for each of the two sites, the alternatives were learned using the data from the previous month and the respective MAE calculated using the respective predictions for the following day (144 predictions given that the periodicity of the data is 10 minutes and the test window is a full day). The predictions for the next day were obtained using a sliding window approach. For instance, at time t and site A we use the available training data to obtain a model that is used to forecast the wind speed at time $t+2h$. After this prediction is obtained, the training window is slided one time step (i.e. 10 mins) and another model is obtained to forecast the value of wind speed at time $t+2h+10mins$. This sliding window process is repeated until we have predictions for all time points in the next day. All model variants are evaluated using the same data.

All data, code and extra results are provided in a web page [3] to ensure that our work is replicable.

4.5.2 Models

We have tried to select a wide range of modeling approaches to test our hypothesis. The idea is to confirm its validity independently of the technique used to forecast. All used tools are freely available in the R software environment R Development Core Team [2010], which ensures easy replication of our work. The following is a list of the methods used in our experiments as well as the considered parameter variants:

Random Walk - a simple baseline method that uses the last wind speed measurement as prediction for the 2 hours ahead wind speed;

Arima - a time series Box-Jenkins model [Pankratz, 1983] based on the R package `forecast` [Hyndman, 2011]. The function `auto.arima` automatically selects the best parameters for the algorithm;

Regression Trees (RT) - a regression tree (e.g. [Breiman, 1984]) based on the R package `rpart` [Therneau and port by B. Ripley., 2009]. In our experiments we have used an interface to the `rpart` function provided in package `DMwR` [Torgo, 2010] and have tried 4 different variants by using the parameter `se` that controls the level of pruning with values: 0, 0.5, 1 and 1.5.

Support Vector Machines (SVM) - an implementation of SVMs (e.g. [Cristianini and Shawe-Taylor, 2000]) available in the R package `e1071`. Six variants were tried by using the parameter *cost* that represents the penalty associated with errors, with the values 10 and 100, and the parameter *epsilon* determines the level of accuracy of the approximated function, we used the values 0.1, 0.3 and 0.5.

Random Forest (RF) - an implementation of random forests [Breiman, 2001] available in the R package `randomForest` [Liaw and Wiener, 2002]. We have used 3 variants of the parameter *ntree* that controls the number of trees in the forest (ensemble), with the values $500, 1000$ and 1500.

[3] `goo.gl/hRBMd`

4.5.3 Results of All Model Variants

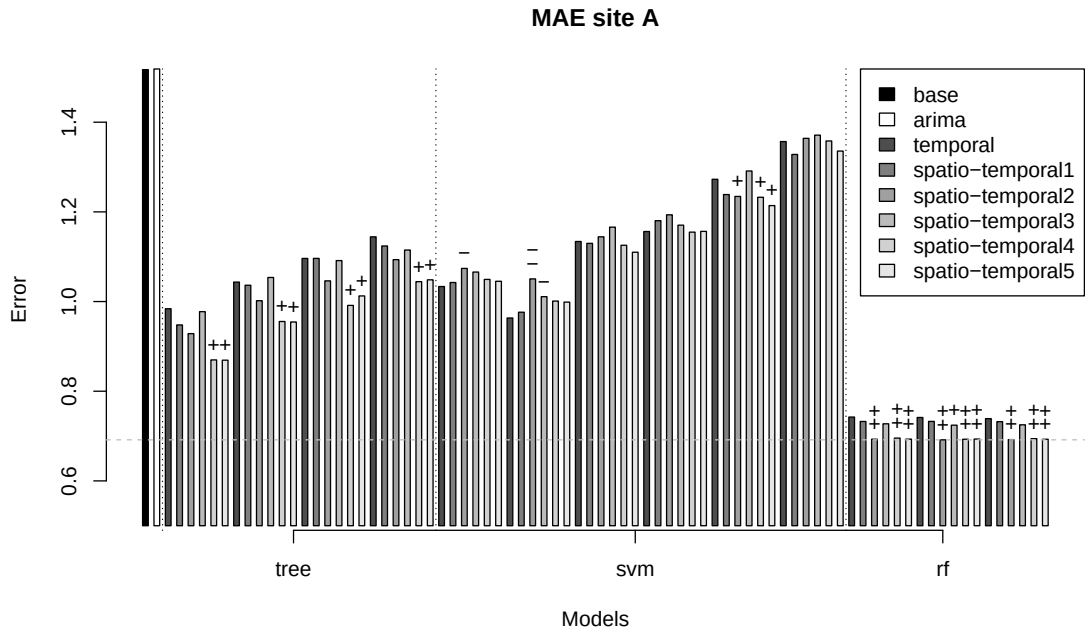


Figure 4.6: Results for site A.

Figures 4.6 and 4.7 summarize the results of all experiments describe in the Section 4.5.1. They present the Monte Carlo estimates of the MAE of all considered variants for the sites A and B, respectively. Each bar is the MAE estimate of a variant. There are four groups of model variants. The first group includes the baseline approaches: the random walk and the arima model. Then we have all variants of the regression trees, SVMs and random forests. For each of the parameter settings we have considered (c.f. Section 4.5.2) we show 6 bars, corresponding to each of the 6 alternative problem formulations we have described in Section 4.5.2. Recall that the main goal of our experiments is to compare the use of the spatio-temporal indicators as predictors against the use of indicators built with data from the same location only. This means we want to compare the 5 last bars of each variant against the first bar (darkest bar of the six). On top of the last five bars we may have one or two symbols (+ or −). They represent the statistical significance of the difference in performance against the first bar according to a paired t-test. A single + (−) means that the respective bar is better (worse) than the first bar with 95% confidence. Two symbols increase the confidence to 99%.

In general, with the exception of some SVM variants, we can say that these experiments

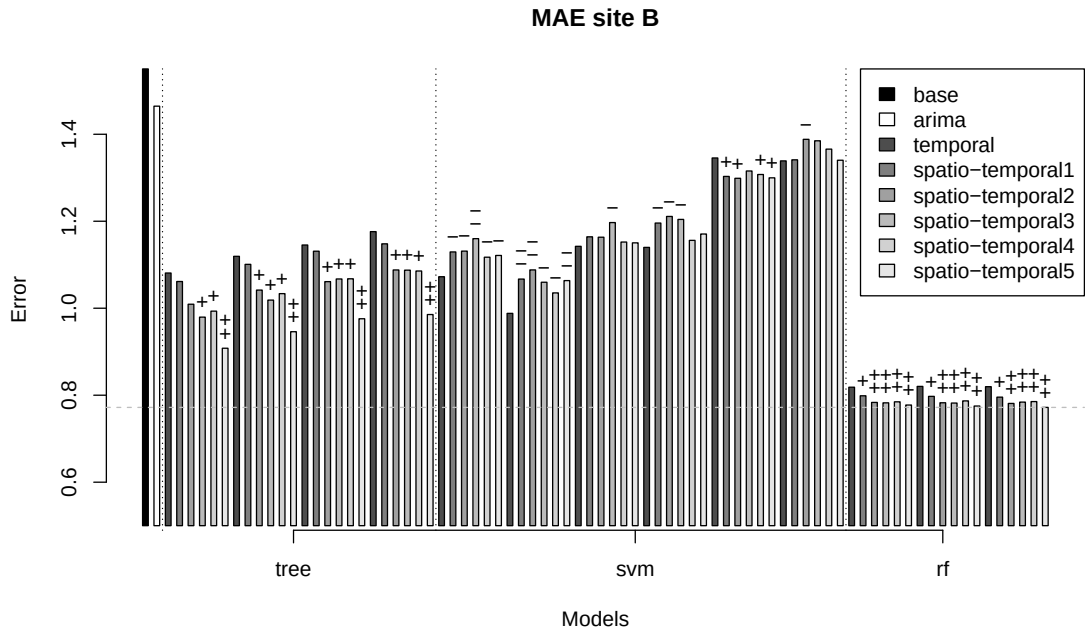


Figure 4.7: Results for site B.

confirm our hypothesis that the use of predictors based on data from a spatio-temporal neighborhood is advantageous in terms of predictive performance. Moreover, for the best models in the set we have considered (Random Forests), this advantage is even more marked. As shown in the graphs the best overall predictive performance is always obtained by some random forest variant using our spatio-temporal indicators. Regression trees have achieved a performance surprisingly competitive with SVMs, and they have also taken advantage of the use of our indicators. The results with SVMs are a bit contradictory and their generally poor performance may provide indications that further parameter tuning may be required for improving their performance, which we will check in Section 4.5.4.

Figure 4.8 summarizes the results on the number of significant paired differences between the spatio-temporal neighborhood variants and the strategy of using only the temporal information. Each bar represents the number of significant wins (+'s) or losses (−'s) of the spatio-temporal variants for the different experimental configurations, one symbol (+ or −) represents 95% of confidence and two symbols represent 99%. We have six combinations of experimental configurations, three models (RT, SVM and RF) on two locations (A and B). The rows of the graph represent the locations A and B, and the columns the models. Looking at the results we can observe that for both the RT (Regression

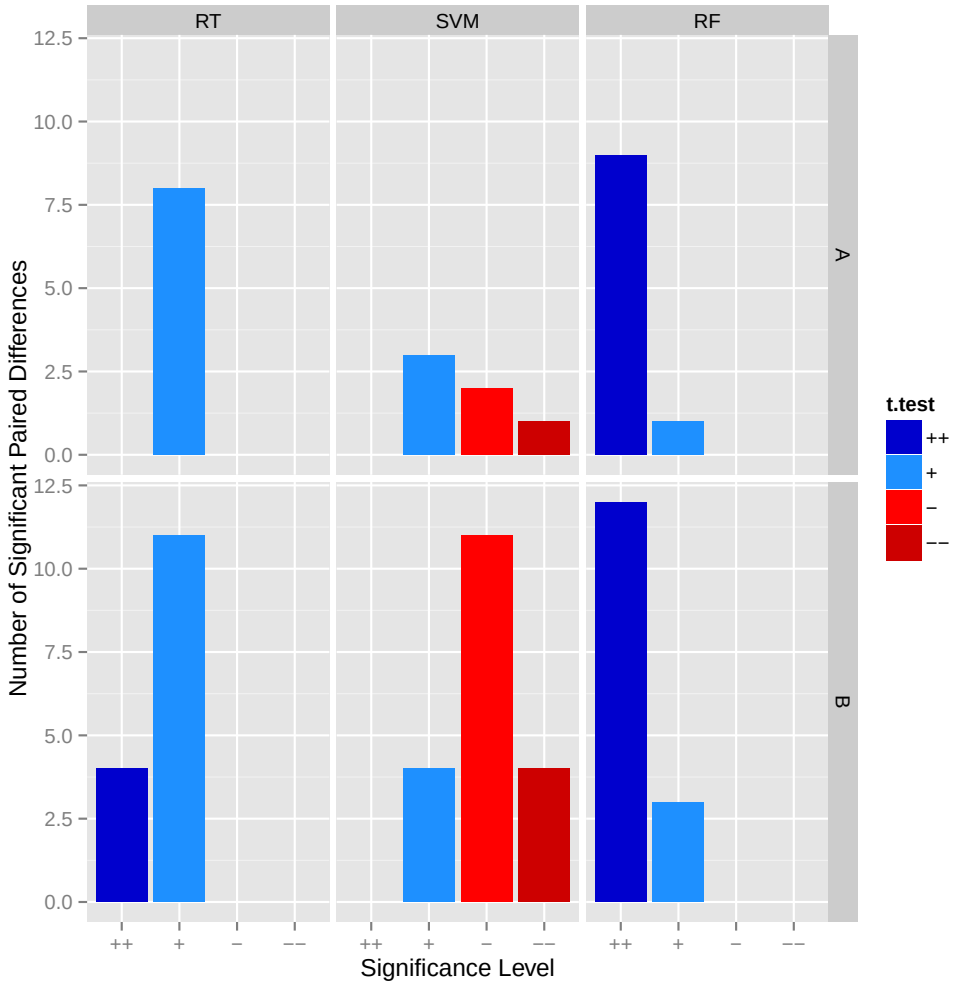


Figure 4.8: Significant Wins vs Losses

Tree) and RF (Random Forests) models there is a significant advantage in using spatio-temporal indicators as predictors. For the SVM model the results are contradictory with this observation. One possible explanation was the bad choice of parameters for the SVM model. In Section 4.5.4 we explore this hypothesis.

4.5.4 Variation of the SVM Model Parameters

SVMs are known for frequently requiring a lot of parameter tuning before their performance is optimal. In our initial set of experiments reported on the previous section, we have observed that the performance of the variants of SVMs that were considered was rather disappointing. In this context, we have decided to check if these results were a consequence of insufficient parameter tuning or of a fundamental flaw of our spatio-temporal

approach. With this goal we have tried 20 extra variants of two SVM parameters, the parameter *cost* with values: $1, 5, 10, 50, 100$ and the parameter *gamma* with the values: $0.001, 0.01, 0.05, 0.1$. To evaluate these new variants we applied the same methodology described for the previous experiment (c.f. Section 4.5.1). The main difference was the fact that we limited the analysis to one site, the location A.

Figures 4.9 and 4.10 summarize the results of this experiment. They present the Monte Carlo estimates of the MAE for these 20 variants of the SVM model at location A. Each group of 6 bars represents the mean absolute error (MAE) of one model variant in all 10 Monte Carlo runs, where the first bar (the darkest) represents the temporal approach and the remaining are the 5 spatio-temporal variants. Based on these results we can draw three main conclusions:

1. A significant improvement on the prediction accuracy of the SVM models in comparison with the previous experiment. In the previous analysis, all the SVM variants had terrible results, with larger errors than Regression Trees. In this new experiment some of the variants (the majority presented in the Figure 4.10), had a substantial improvement in the prediction accuracy, clearly outperforming Regression Trees and getting closer to the best model on the previous experiment, the Random Forest.

2. Generally SVMs take advantage of the spatio-temporal neighborhood strategy. A considerable number of SVM variants had improvement in the prediction error when using spatio-temporal neighborhood information.

3. However, the best result was achieved using the parameters *cost*: 50 and *gamma*: 0.1, with the temporal only neighborhood variant. Still, this score was followed closely by some of the respective spatio-temporal variants.

4.5.5 Sensitivity Analysis for the Best Model Configurations

In Section 4.5.3 we compared the use of different spatio-temporal neighborhoods against the use of only temporal neighborhoods, in the task of forecasting the next 2 hours ahead wind speed. The goal of that experiment was to validate the advantages of using predictors that try to capture the spatio-temporal correlation between locations against using predictors

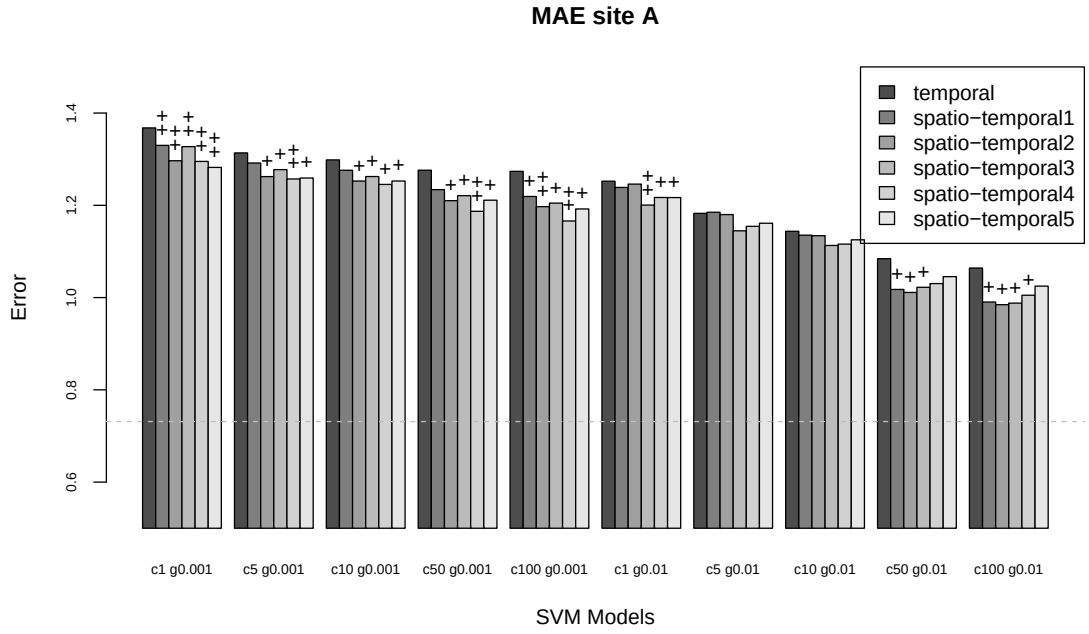


Figure 4.9: Results of Further Variants of SVMs (first 10 extra variants).

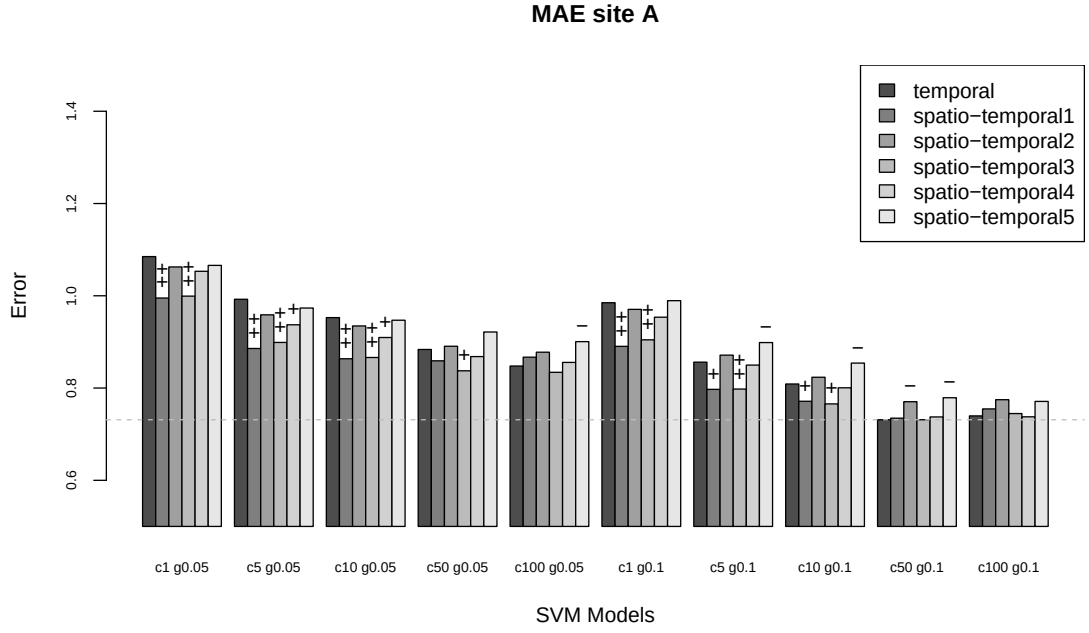


Figure 4.10: Results of Further Variants of SVMs (second 10 extra variants).

that capture only the temporal correlation. To allow for fair paired comparisons all the spatio-temporal configurations were based on neighborhoods with the same temporal size (10, 20 and 30 days).

In this section we peek the two best configurations in terms of spatial neighborhood sizes: (1) 100, 500 and 900 km (config 4); and (2) 150, 675 and 1200km (config 5); and checked the impact on their performance when we vary the temporal size (i.e. the height of the cones). More specifically, we have used the spatial configurations of the configs 4 and 5: (i) (100, 500 and 900 km) and (ii) (150, 675 and 1200km), and combined them with the following temporal dimensions: (i) 10, 30 and 60 days; and (ii) 7, 15 and 30 days. This leads to the following four new spatio-temporal configurations:

config 4_i: [100 km, 10 days], [500 km, 30 days] and [900 km, 60 days];

config 4_ii: [100 km, 7 days], [500 km, 15 days] and [900 km, 30 days];

config 5_i: [150 km, 10 days], [675 km, 30 days] and [1200 km, 60 days];

config 5_ii: [150 km, 7 days], [675 km, 15 days] and [1200 km, 30 days].

We have used the same experimental methodology as before to evaluate these four spatio-temporal configurations. However, this new set of experiments was limited to the location A. Once again the results we present are obtained with 10 repetitions of a Monte Carlo simulation, with a train size of one month and test size of the next day. As modeling technique we have selected the model that achieved the best results on the previous experiments: the Random Forest with `ntree` = 500.

Figure 4.11 summarizes the results of this experiment. Each bar represents the Mean Absolute Error estimated by the 10 repetitions of the Monte Carlo simulation. The black bar represents the results of `config 4`, while the dark grey bars the new variants (`config 4_i` and `config 4_ii`). The white bar represents the results of the variant `config 5`, and the light grey bars the new variants `config 5_i` and `config 5_ii`.

The results of these experiments reveal some degree of variability when we change the temporal neighborhoods of the two best variants of the previous experiments. This indicates that there is potentially some sensitivity to the correct choice of this neighborhood. Still, in general we have confirmed the good results obtained by these two variants. In effect,

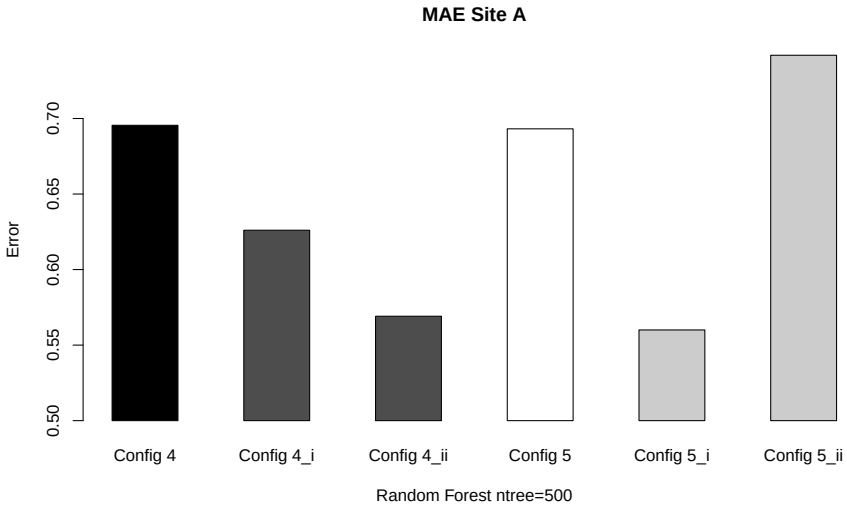


Figure 4.11: Results of the Random Forest for the new configurations at site A.

some of the new variants where even able to outperform the previous scores, although for `config 5` one of the new variants lead to a slightly worse MAE score.

4.6 Conclusions

This chapter has described a new methodology for short-term wind speed prediction, a class of problems with extreme relevance for electricity markets and wind power production. We proposed a new formalization of this spatio-temporal prediction problem, which includes the definition of spatio-temporal indicators. These predictors provide information on the spatio-temporal dynamics of the target time series. Our proposal is general and can be applied to any spatio-temporal prediction task. These type of prediction problems are becoming more and more relevant with the prevalence of mobile computing devices with localization features. In this chapter we have tested our proposal on the task of forecasting the wind speed for a two hours ahead horizon in the eastern region of the US. Our experimental results confirm the advantages of the use of spatio-temporal information on this prediction task. Models using our spatio-temporal indicators have generally obtained superior performance.

Chapter 5

Conclusions and Future Directions

In this chapter we summarize the main contributions and results achieved in this thesis, and we describe some future research topics.

5.1 Summary

This thesis was driven by several concrete real world applications. The connecting link between these problems is that they require handling forecasting tasks with data that has a temporal, spatial or spatio-temporal nature. This means that for all applications that were tackled there was some form of correlation between the available data points. The main hypothesis driving our thesis was that it is possible to transform these prediction tasks into standard regression tasks, provided we find means of giving the models information on the temporal, spatial or spatio-temporal correlation between data points. The main advantage of proceeding this way is on the fact that we are able to take advantage of the wide range of existing modelling tools without having the need to develop special-purpose tools. In this context, our work was mainly centred on the question of how to properly convey this spatio-temporal correlation information to the models. This has lead to the development of a series of pre-processing steps that are able to provide this information to the models.

The first tasks we have addressed have a temporal nature, i.e. they belong to the field of time series analysis. The concrete tasks of water quality monitoring and water demand

forecasting, have lead us to propose a new type of time series prediction tasks: 2D interval predictions. This task addresses problems where one wants to forecast a range of plausible values of the target time series for a future time interval. To the best of our knowledge this task was never formally defined/proposed and applications requiring this type of predictions would typically be solved using an indirect approach (e.g. iterated predictions). Our main goal was to alert the data mining community to this new task, given the large number of important potential applications. We proposed a new methodology to address this task. This methodology consists in transforming the problem into the problem of forecasting the future value of some descriptive statistics. We have also proposed a series of error metrics to properly evaluate models in these tasks. We have tested our proposal on an extensive set of experiments, considering both its predictive accuracy and its computational costs. The results of these experiments have show that our proposal is highly competitive with the best alternatives in the different experimental set-ups considered, but with a significantly lower computational cost. This makes the proposal particularly adequate for high-frequency time series.

The second task we have addressed has to do with spatial imputation or interpolation. Spatial interpolation techniques are typically based on some variation of *first law of geography*, that gives more importance to neighbors in the prediction of unsampled locations. Once again our approach was to transform the spatial interpolation problem into a standard multiple regression task. This transformation was accomplished by deriving a series of spatial indicators with the goal of incorporating the information on the spatial correlation between neighboring locations into the predictors of the regression task. This procedure has two main advantages: (i) firstly we are able to use the wide range of regression tools that are available; (ii) secondly, and more importantly, this allows the use of data from far away regions when predicting the value for a certain location provided the models judge these locations as having similar spatial dynamics (as described by our proposed indicators). We have extensively tested our approach against state of the art spatial interpolation method on a real world task: filling in the missing pixels on several images. On all set-ups we have observed a strong advantage of our approach in recovering the missing pixels of the images even at high levels of noise. These results are very encouraging and provide strong empirical evidence towards the advantages of our approach to spatial interpolation.

The final application domain we have addressed is wind speed forecasting. This is a very

relevant application for wind-based energy production. The task in this case consisted on forecasting the near future (2 hours ahead) wind speed on some location, having information on historical wind speed values on both this location and nearby locations. This means that it is a spatio-temporal forecasting task. Given the spatio-temporal correlation among data points we have tried to solve this problem again using a pre-processing approach that allows its transformation into a standard regression problem. Our strategy consisted in developing a series of spatio-temporal indicators that convey relevant information on the spatial and temporal dynamics of the target variable in the nearby regions to the target location. We tested our approach against the most common approach that considers only the temporal dimension, thus handling the problem as a time series task. The experimental evaluation was carried out using a real world wind speed data set collected at wind farms in the US. On the majority of the experimental set-ups we have considered our technique has shown significant advantages. These results provide strong empirical evidence to support the claim that by combining information on both the spatial and temporal correlation of data points better predictive accuracy is achievable.

The work carried out on this thesis has shown that complex problems can often be handled by carefully chosen pre-processing steps, allowing the use of standard of-the-shelf tools with given proofs. Our contributions and the results achieved on several concrete domains of application provided strong evidence on the validity of these approaches on the case of data sets with temporal, spatial or spatio-temporal correlation among data points.

Although the approaches presented on this thesis were driven by concrete real world applications, they are general methodologies that can be applicable to other domains. In this context, the work carried out in the thesis can be regarded as a general methodology for addressing spatio-temporal prediction tasks, which are key applications given the current trends on data collection devices.

5.2 Main Contributions

The thesis has described a series of work on several real world applications. The following is a list of what we think are the main contributions of our work:

The definition of a new task in time series forecasting: motivated by real world ap-

plications, we have defined the task of 2D-interval prediction. This task consist on forecasting the range of plausible values of a time series variable for a future time window.

A new method for handling 2D-interval predictions: we have described a pre-processing method that transforms this task into two standard regression tasks where the target variables are statistics of the the distribution of the target time series for a certain time window. We have used the 1st and 3rd quartiles as good representative statistics of the plausible range of values.

Error metrics for 2D-interval prediction tasks: we have proposed three new error metrics (TQE, MAQ, Benefit Matrix) for 2D-interval prediction tasks.

A new methodology for spatial interpolation: we have described a new method for spatial interpolation based on a series of pre-processing steps designed to provide information on the spatial dynamics of the neighborhood of the target locations, which allows handling this task as a standard regression problem.

A new methodology for spatio-temporal prediction: we have proposed a new method for solving spatio-temporal forecasting problems. This method is again based on a series of pre-processing steps that produce indicators of the spatio-temporal correlation between data points, allowing the use of standard regression tools.

5.3 Future Research Directions

This thesis has presented several new approaches that were designed to address concrete real world data mining tasks. In spite of this, these new approaches are general methods for some classes of spatio-temporal prediction tasks. In the context of these proposed methodologies, several paths for future research exist. Still, given that all methodologies are strongly based on pre-processing steps designed to create new variables used in the modeling tasks, a common topic that deserves further research is that of analyzing more extensively further alternatives in terms of these created variables.

Regards 2D-interval predictions one obvious extension of our work is to study our approach for a more theoretical perspective. Namely, the key issue in our proposed method is that we

are directly forecasting distribution statistics instead of calculating them from predictions of actual values. Our experiments have shown advantages in terms of the used predictive metrics, however it should be interesting to study whether there is some theoretical reason justifying these results. We have suggested the hypothesis that forecasting robust statistics like the quartiles is easier than forecasting the variable values, as the distribution of the quantiles is smoother than the distribution of the actual variables. Still, this requires a deeper study to be confirmed and/or proved.

With respect to spatial interpolation the results of our method were extremely interesting when compared to state of the art methods. However, future work should extend these comparisons to other domains, namely outside of image analysis. Additionally, we plan to study more deeply the reasons for the success of the proposed methodology and its eventual impact on the way spatial data analysis is usually carried out, which is mostly based on the use of data from a certain vicinity supported by Tobler's first law of geography.

Finally, regards spatio-temporal prediction tasks further work should also extend the experiments to other domains. Moreover, we should also explore other possibilities regards spatio-temporal indicators that describe the spatio-temporal dynamics within each region and also further tests regards the neighborhood sizes.

Appendices

Appendix A

Water Consumption Results

A.1 Window Size 12

This section presents the results for the water consumption problem for the window size of 12 values, $k = 12$.

A.1.1 MAQ - Mean Absolute Quantile Deviation

Figure A.1 shows the results using the MAQ error measure for all different setups.

Table A.1 shows all setups order by the MAQ error.

	MAQ.Q1	MAQ.Q3	MAQ
Quantile RF ntree 1500	2.15	3.15	2.65
Quantile RF ntree 1000	2.15	3.15	2.65
Quantile RF ntree 500	2.15	3.15	2.65
Nmodels QRF ntree 1500	2.33	3.12	2.72
Nmodels QRF ntree 1000	2.33	3.12	2.72
Nmodels QRF ntree 500	2.33	3.12	2.72
Quantile SVM cost 1 gamma 0.05	2.39	3.37	2.88
Quantile SVM cost 1 gamma 0.001	2.45	3.36	2.9

Quantile SVM cost 10 gamma 0.01	2.39	3.46	2.93
Quantile SVM cost 50 gamma 0.01	2.42	3.46	2.94
Quantile SVM cost 5 gamma 0.001	2.46	3.43	2.94
Iterated QRF ntree 500	2.73	3.16	2.95
Quantile SVM cost 5 gamma 0.01	2.4	3.49	2.95
Iterated QRF ntree 1500	2.72	3.17	2.95
Iterated QRF ntree 1000	2.73	3.17	2.95
Quantile SVM cost 1 gamma 0.1	2.42	3.48	2.95
Quantile SVM cost 1 gamma 0.01	2.4	3.5	2.95
Quantile SVM cost 10 gamma 0.001	2.47	3.47	2.97
Quantile SVM cost 100 gamma 0.01	2.44	3.52	2.98
Quantile SVM cost 50 gamma 0.001	2.46	3.52	2.99
Quantile SVM cost 5 gamma 0.05	2.49	3.52	3.01
Quantile SVM cost 100 gamma 0.001	2.46	3.55	3.01
Quantile SVM cost 10 gamma 0.05	2.56	3.62	3.09
Quantile SVM cost 5 gamma 0.1	2.52	3.73	3.12
Quantile SVM cost 10 gamma 0.1	2.58	3.84	3.21
Quantile RT se 0	2.77	3.87	3.32
Quantile SVM cost 50 gamma 0.05	2.69	3.96	3.32
Quantile RT se 0.5	2.72	3.96	3.34
Quantile RT se 1.5	2.63	4.06	3.34
Quantile RT se 1	2.66	4.04	3.35
Nmodels SVM cost 5 gamma 0.05	2.97	3.89	3.43
Nmodels SVM cost 1 gamma 0.1	3.02	3.86	3.44
Nmodels SVM cost 50 gamma 0.01	3.06	3.82	3.44
Nmodels SVM cost 100 gamma 0.01	3.06	3.85	3.45
Nmodels SVM cost 1 gamma 0.05	3.02	3.89	3.45
Nmodels RF ntree 1500	3.54	3.43	3.49
Nmodels RF ntree 1000	3.55	3.43	3.49
Nmodels RF ntree 500	3.54	3.44	3.49
Nmodels SVM cost 5 gamma 0.1	3.01	3.97	3.49

Quantile SVM cost 50 gamma 0.1	2.82	4.2	3.51
Nmodels SVM cost 10 gamma 0.05	3.01	4	3.51
Quantile SVM cost 100 gamma 0.05	2.78	4.24	3.51
Nmodels SVM cost 10 gamma 0.1	3.03	3.99	3.51
Nmodels SVM cost 10 gamma 0.01	3.18	3.92	3.55
Quantile QRF ntree 500	3.59	3.56	3.57
Iterated SVM cost 5 gamma 0.1	2.91	4.23	3.57
Quantile QRF ntree 1000	3.59	3.55	3.57
Quantile QRF ntree 1500	3.59	3.56	3.57
Iterated SVM cost 1 gamma 0.1	2.97	4.21	3.59
Iterated RF ntree 1000	3.76	3.43	3.6
Iterated RF ntree 1500	3.77	3.44	3.6
Iterated RF ntree 500	3.77	3.45	3.61
Nmodels SVM cost 5 gamma 0.01	3.24	4	3.62
Nmodels RT se 0	3.38	3.94	3.66
Iterated SVM cost 5 gamma 0.05	3.1	4.24	3.67
Iterated SVM cost 1 gamma 0.05	3.13	4.22	3.68
Nmodels SVM cost 50 gamma 0.1	3.17	4.2	3.68
Iterated SVM cost 10 gamma 0.05	3.07	4.3	3.69
Iterated SVM cost 100 gamma 0.01	3.32	4.06	3.69
Nmodels SVM cost 50 gamma 0.05	3.24	4.14	3.69
Quantile SVM cost 100 gamma 0.1	2.92	4.5	3.71
Iterated SVM cost 10 gamma 0.1	2.98	4.46	3.72
Nmodels SVM cost 100 gamma 0.001	3.43	4.08	3.76
Iterated SVM cost 50 gamma 0.01	3.37	4.16	3.77
Nmodels SVM cost 1 gamma 0.01	3.33	4.21	3.77
Nmodels SVM cost 50 gamma 0.001	3.43	4.11	3.77
Iterated SVM cost 5 gamma 0.01	3.24	4.3	3.77
Iterated SVM cost 10 gamma 0.01	3.31	4.27	3.79
Nmodels SVM cost 100 gamma 0.05	3.39	4.24	3.81
Iterated SVM cost 1 gamma 0.01	3.24	4.42	3.83

Nmodels SVM cost 10 gamma 0.001	3.46	4.2	3.83
Nmodels SVM cost 5 gamma 0.001	3.5	4.23	3.86
Nmodels RT se 0.5	3.7	4.09	3.9
Iterated SVM cost 100 gamma 0.001	3.5	4.31	3.9
Nmodels SVM cost 100 gamma 0.1	3.33	4.49	3.91
Iterated SVM cost 50 gamma 0.001	3.49	4.37	3.93
Iterated SVM cost 50 gamma 0.05	3.13	4.78	3.95
Nmodels SVM cost 1 gamma 0.001	3.67	4.32	4
Iterated SVM cost 1 gamma 0.001	3.44	4.63	4.03
Nmodels RT se 1	3.95	4.16	4.06
Iterated SVM cost 10 gamma 0.001	3.53	4.59	4.06
Iterated SVM cost 5 gamma 0.001	3.5	4.7	4.1
Iterated SVM cost 50 gamma 0.1	3.57	4.69	4.13
RW	2.94	5.33	4.13
Nmodels RT se 1.5	4.12	4.18	4.15
Iterated SVM cost 100 gamma 0.05	3.46	5.01	4.24
Iterated SVM cost 100 gamma 0.1	4	4.85	4.42
Iterated RT se 1.5	5.73	5.34	5.54
Iterated RT se 1	5.73	5.53	5.63
Iterated RT se 0	5.8	5.61	5.71
Iterated RT se 0.5	5.8	5.62	5.71

Table A.1: All setups, k = 12 and MAQ

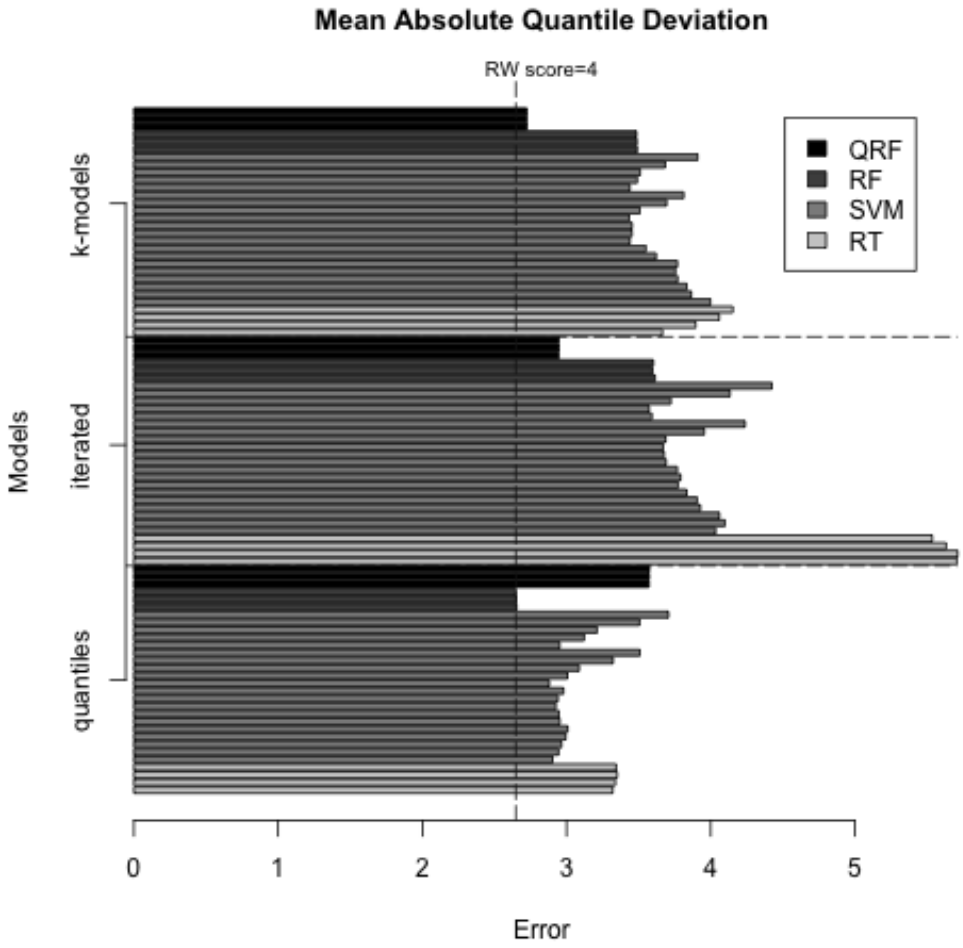


Figure A.1: Water Consumption, k = 12 and MAQ

A.1.2 TQE - Total Quantile Error

Figure A.2 shows the results using the TQE error measure for all different setups.

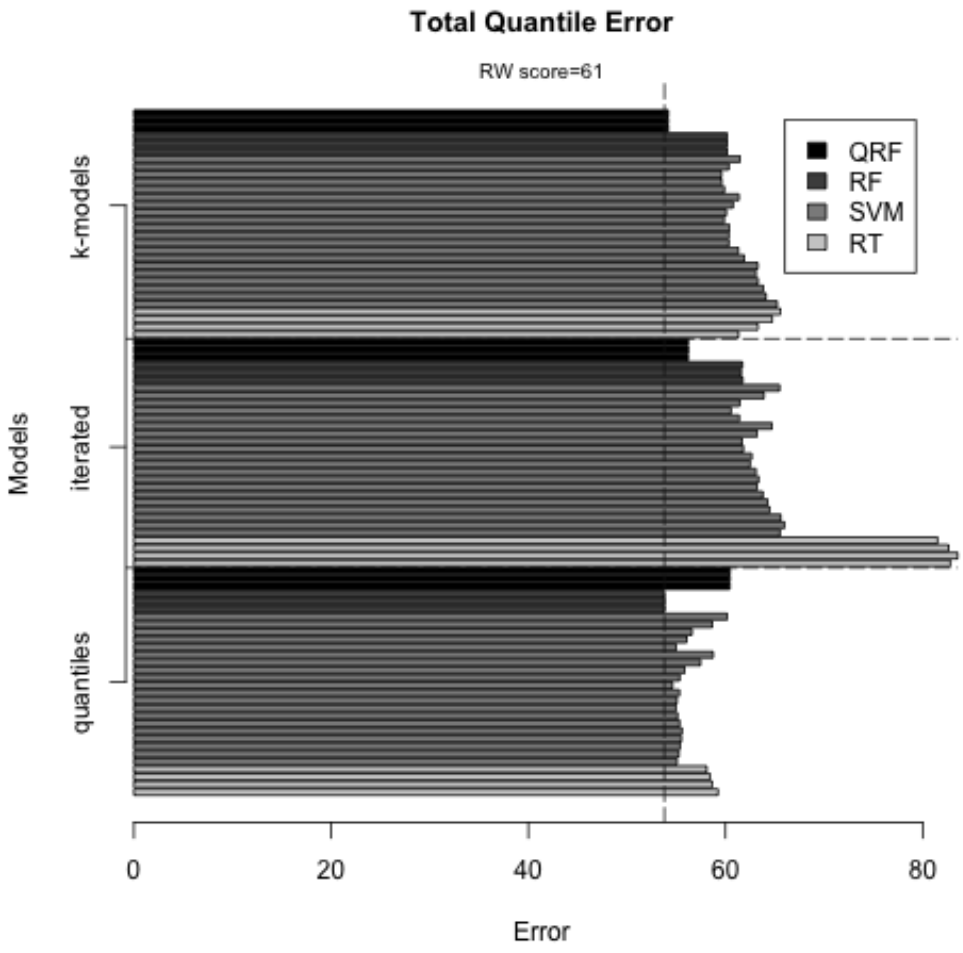


Figure A.2: Water Consumption, k = 12 and TQE

Table A.2 shows all setups order by the TQE error.

	QTE.Q1	QTE.Q3	QTE
Quantile RF ntree 1500	24.93	28.88	53.81
Quantile RF ntree 1000	24.93	28.88	53.81
Quantile RF ntree 500	24.93	28.89	53.82
Nmodels QRF ntree 1000	25.4	28.79	54.19

Nmodels QRF ntree 1500	25.4	28.79	54.19
Nmodels QRF ntree 500	25.4	28.79	54.19
Quantile SVM cost 1 gamma 0.05	25.41	29.22	54.63
Quantile SVM cost 10 gamma 0.01	25.46	29.51	54.97
Quantile SVM cost 1 gamma 0.1	25.48	29.51	54.99
Quantile SVM cost 50 gamma 0.01	25.58	29.49	55.08
Quantile SVM cost 1 gamma 0.001	25.52	29.57	55.09
Quantile SVM cost 5 gamma 0.01	25.45	29.68	55.14
Quantile SVM cost 5 gamma 0.001	25.55	29.73	55.27
Quantile SVM cost 100 gamma 0.01	25.67	29.66	55.33
Quantile SVM cost 1 gamma 0.01	25.4	29.96	55.35
Quantile SVM cost 5 gamma 0.05	25.71	29.65	55.36
Quantile SVM cost 10 gamma 0.001	25.55	29.85	55.4
Quantile SVM cost 50 gamma 0.001	25.55	29.98	55.53
Quantile SVM cost 100 gamma 0.001	25.57	30.02	55.59
Quantile SVM cost 10 gamma 0.05	25.94	29.92	55.86
Quantile SVM cost 5 gamma 0.1	25.82	30.25	56.07
Iterated QRF ntree 500	26.82	29.39	56.21
Iterated QRF ntree 1500	26.81	29.41	56.22
Iterated QRF ntree 1000	26.81	29.41	56.23
Quantile SVM cost 10 gamma 0.1	26.05	30.56	56.61
Quantile SVM cost 50 gamma 0.05	26.37	31.1	57.47
Quantile RT se 1.5	26.5	31.6	58.1
Quantile RT se 1	26.76	31.67	58.43
Quantile SVM cost 50 gamma 0.1	26.93	31.74	58.67
Quantile RT se 0.5	27.07	31.61	58.68
Quantile SVM cost 100 gamma 0.05	26.78	31.98	58.75
Quantile RT se 0	27.58	31.66	59.24
Nmodels SVM cost 10 gamma 0.1	28.09	31.51	59.6
Nmodels SVM cost 5 gamma 0.1	28.07	31.55	59.62
Nmodels SVM cost 5 gamma 0.05	27.98	31.89	59.86

Nmodels SVM cost 1 gamma 0.1	28.05	31.84	59.89
Nmodels SVM cost 10 gamma 0.05	28.1	31.99	60.08
Nmodels RF ntree 1500	30.03	30.13	60.16
Nmodels RF ntree 1000	30.04	30.12	60.16
Nmodels RF ntree 500	30.03	30.14	60.16
Quantile SVM cost 100 gamma 0.1	27.31	32.86	60.17
Nmodels SVM cost 50 gamma 0.1	28.44	31.89	60.34
Nmodels SVM cost 50 gamma 0.01	28.33	32.01	60.34
Nmodels SVM cost 100 gamma 0.01	28.35	31.99	60.34
Nmodels SVM cost 1 gamma 0.05	28.17	32.23	60.4
Quantile QRF ntree 500	29.91	30.53	60.44
Quantile QRF ntree 1000	29.93	30.52	60.45
Quantile QRF ntree 1500	29.93	30.53	60.45
Iterated SVM cost 5 gamma 0.1	27.5	33.06	60.56
Nmodels SVM cost 50 gamma 0.05	28.78	31.99	60.77
RW	27.6	33.48	61.08
Nmodels SVM cost 10 gamma 0.01	28.82	32.48	61.3
Nmodels RT se 0	29.52	31.81	61.33
Nmodels SVM cost 100 gamma 0.05	29.2	32.17	61.37
Iterated SVM cost 1 gamma 0.1	27.87	33.53	61.4
Iterated SVM cost 10 gamma 0.1	27.73	33.71	61.44
Nmodels SVM cost 100 gamma 0.1	28.79	32.67	61.46
Iterated RF ntree 1000	31.13	30.47	61.6
Iterated SVM cost 10 gamma 0.05	28.34	33.33	61.67
Iterated RF ntree 1500	31.19	30.52	61.71
Iterated RF ntree 500	31.19	30.55	61.75
Iterated SVM cost 5 gamma 0.05	28.57	33.25	61.82
Nmodels SVM cost 5 gamma 0.01	29.07	32.84	61.92
Iterated SVM cost 100 gamma 0.01	29.44	33.05	62.49
Iterated SVM cost 1 gamma 0.05	28.76	33.91	62.67
Nmodels SVM cost 100 gamma 0.001	29.84	33.25	63.09

Iterated SVM cost 50 gamma 0.01	29.66	33.43	63.09
Iterated SVM cost 5 gamma 0.01	28.89	34.29	63.19
Iterated SVM cost 50 gamma 0.05	28.54	34.66	63.21
Nmodels SVM cost 1 gamma 0.01	29.59	33.69	63.28
Nmodels RT se 0.5	30.87	32.42	63.29
Nmodels SVM cost 50 gamma 0.001	29.92	33.38	63.3
Iterated SVM cost 10 gamma 0.01	29.25	34.1	63.36
Iterated SVM cost 1 gamma 0.01	28.88	34.91	63.79
Nmodels SVM cost 10 gamma 0.001	30.18	33.65	63.83
Iterated SVM cost 50 gamma 0.1	29.58	34.32	63.9
Nmodels SVM cost 5 gamma 0.001	30.36	33.75	64.11
Iterated SVM cost 100 gamma 0.001	29.85	34.41	64.26
Iterated SVM cost 50 gamma 0.001	29.86	34.6	64.47
Iterated SVM cost 100 gamma 0.05	29.27	35.44	64.71
Nmodels RT se 1	32.03	32.7	64.73
Nmodels SVM cost 1 gamma 0.001	31.07	34.18	65.24
Iterated SVM cost 100 gamma 0.1	31.1	34.39	65.49
Iterated SVM cost 1 gamma 0.001	29.67	35.86	65.52
Nmodels RT se 1.5	32.83	32.75	65.58
Iterated SVM cost 10 gamma 0.001	30.05	35.55	65.6
Iterated SVM cost 5 gamma 0.001	29.92	36.05	65.97
Iterated RT se 1.5	43.71	37.8	81.51
Iterated RT se 1	43.66	38.96	82.63
Iterated RT se 0	43.64	39.16	82.81
Iterated RT se 0.5	44.01	39.54	83.54

Table A.2: All setups, k = 12 and TQE

A.1.3 Utility

Figure A.3 shows the results using the Utility error measure (larger is better) for all different setups.

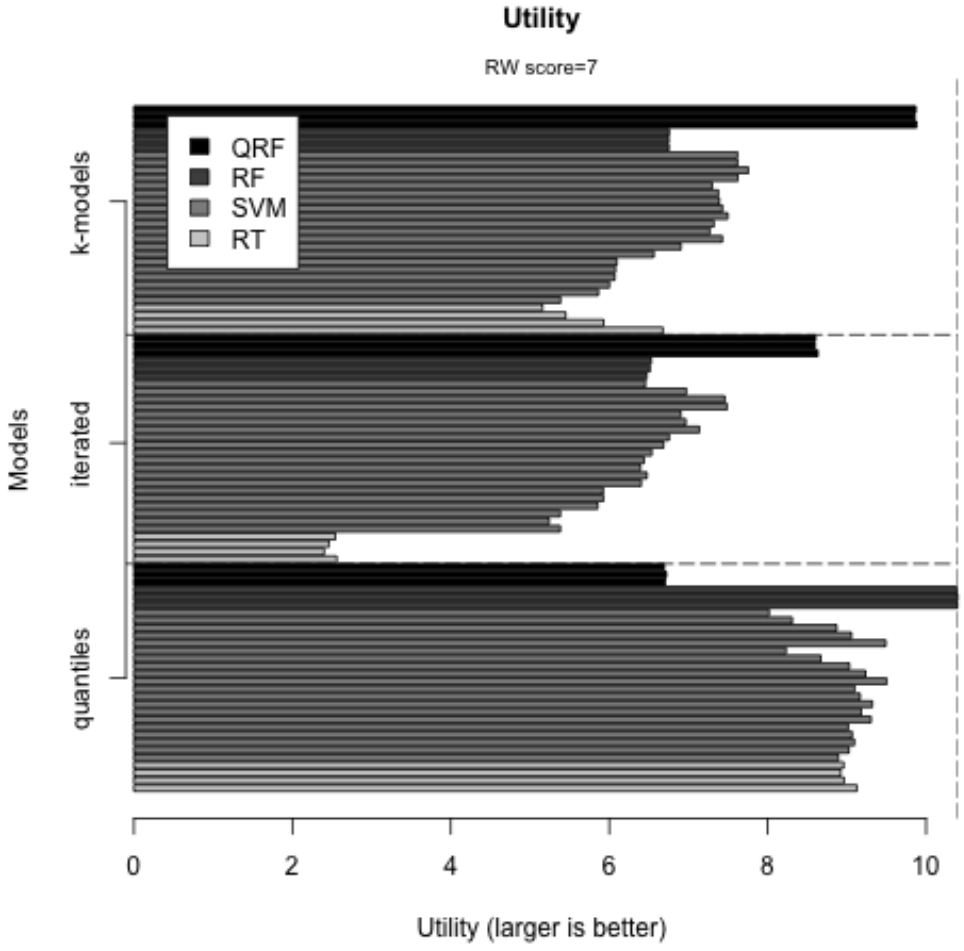


Figure A.3: Water Consumption, k = 12 and Utility

Table A.3 shows all setups order by the Utility.

	utility
Quantile RF ntree 1000	10.4
Quantile RF ntree 500	10.39
Quantile RF ntree 1500	10.39

Nmodels QRF ntree 500	9.88
Nmodels QRF ntree 1500	9.87
Nmodels QRF ntree 1000	9.86
Quantile SVM cost 1 gamma 0.05	9.5
Quantile SVM cost 1 gamma 0.1	9.49
Quantile SVM cost 10 gamma 0.01	9.32
Quantile SVM cost 1 gamma 0.01	9.3
Quantile SVM cost 5 gamma 0.05	9.23
Quantile SVM cost 5 gamma 0.01	9.18
Quantile SVM cost 50 gamma 0.01	9.17
Quantile RT se 0	9.12
Quantile SVM cost 100 gamma 0.01	9.1
Quantile SVM cost 10 gamma 0.001	9.09
Quantile SVM cost 50 gamma 0.001	9.07
Quantile SVM cost 5 gamma 0.1	9.06
Quantile SVM cost 10 gamma 0.05	9.03
Quantile SVM cost 5 gamma 0.001	9.02
Quantile SVM cost 100 gamma 0.001	9.02
Quantile RT se 0.5	8.96
Quantile RT se 1.5	8.96
Quantile RT se 1	8.92
Quantile SVM cost 1 gamma 0.001	8.89
Quantile SVM cost 10 gamma 0.1	8.87
Quantile SVM cost 50 gamma 0.05	8.67
Iterated QRF ntree 500	8.63
Iterated QRF ntree 1500	8.6
Iterated QRF ntree 1000	8.6
Quantile SVM cost 50 gamma 0.1	8.31
Quantile SVM cost 100 gamma 0.05	8.23
Quantile SVM cost 100 gamma 0.1	8.02
Nmodels SVM cost 10 gamma 0.1	7.75

Nmodels SVM cost 5 gamma 0.1	7.62
Nmodels SVM cost 50 gamma 0.1	7.62
Nmodels SVM cost 100 gamma 0.1	7.62
Nmodels SVM cost 5 gamma 0.05	7.5
Iterated SVM cost 5 gamma 0.1	7.48
Iterated SVM cost 10 gamma 0.1	7.46
Nmodels SVM cost 10 gamma 0.05	7.43
Nmodels SVM cost 50 gamma 0.01	7.43
Nmodels SVM cost 50 gamma 0.05	7.39
Nmodels SVM cost 100 gamma 0.05	7.38
Nmodels SVM cost 1 gamma 0.05	7.32
Nmodels SVM cost 1 gamma 0.1	7.3
Nmodels SVM cost 100 gamma 0.01	7.27
Iterated SVM cost 50 gamma 0.05	7.14
Iterated SVM cost 50 gamma 0.1	6.97
Iterated SVM cost 100 gamma 0.05	6.97
Nmodels SVM cost 10 gamma 0.01	6.91
Iterated SVM cost 1 gamma 0.1	6.9
Nmodels RF ntree 1500	6.76
Iterated SVM cost 10 gamma 0.05	6.76
Nmodels RF ntree 500	6.75
Nmodels RF ntree 1000	6.75
Quantile QRF ntree 1000	6.72
Quantile QRF ntree 500	6.71
Quantile QRF ntree 1500	6.69
RW	6.69
Iterated SVM cost 5 gamma 0.05	6.68
Nmodels RT se 0	6.68
Nmodels SVM cost 5 gamma 0.01	6.56
Iterated SVM cost 1 gamma 0.05	6.54
Iterated RF ntree 1500	6.53

Iterated RF ntree 1000	6.51
Iterated SVM cost 10 gamma 0.01	6.48
Iterated RF ntree 500	6.47
Iterated SVM cost 100 gamma 0.1	6.46
Iterated SVM cost 100 gamma 0.01	6.44
Iterated SVM cost 5 gamma 0.01	6.4
Iterated SVM cost 50 gamma 0.01	6.39
Nmodels SVM cost 1 gamma 0.01	6.1
Nmodels SVM cost 100 gamma 0.001	6.08
Nmodels SVM cost 50 gamma 0.001	6.07
Nmodels SVM cost 10 gamma 0.001	6
Nmodels RT se 0.5	5.93
Iterated SVM cost 100 gamma 0.001	5.93
Iterated SVM cost 1 gamma 0.01	5.92
Nmodels SVM cost 5 gamma 0.001	5.86
Iterated SVM cost 50 gamma 0.001	5.85
Nmodels RT se 1	5.45
Nmodels SVM cost 1 gamma 0.001	5.38
Iterated SVM cost 10 gamma 0.001	5.38
Iterated SVM cost 1 gamma 0.001	5.38
Iterated SVM cost 5 gamma 0.001	5.24
Nmodels RT se 1.5	5.16
Iterated RT se 0	2.56
Iterated RT se 1.5	2.54
Iterated RT se 1	2.46
Iterated RT se 0.5	2.4

Table A.3: All setups, k = 12 and Utility

A.2 Window Size 24

This section presents the results for the water consumption problem for the window size of 24 values, $k = 24$.

A.2.1 MAQ - Mean Absolute Quantile Deviation

Figure A.4 shows the results using the MAQ error measure for all different setups.

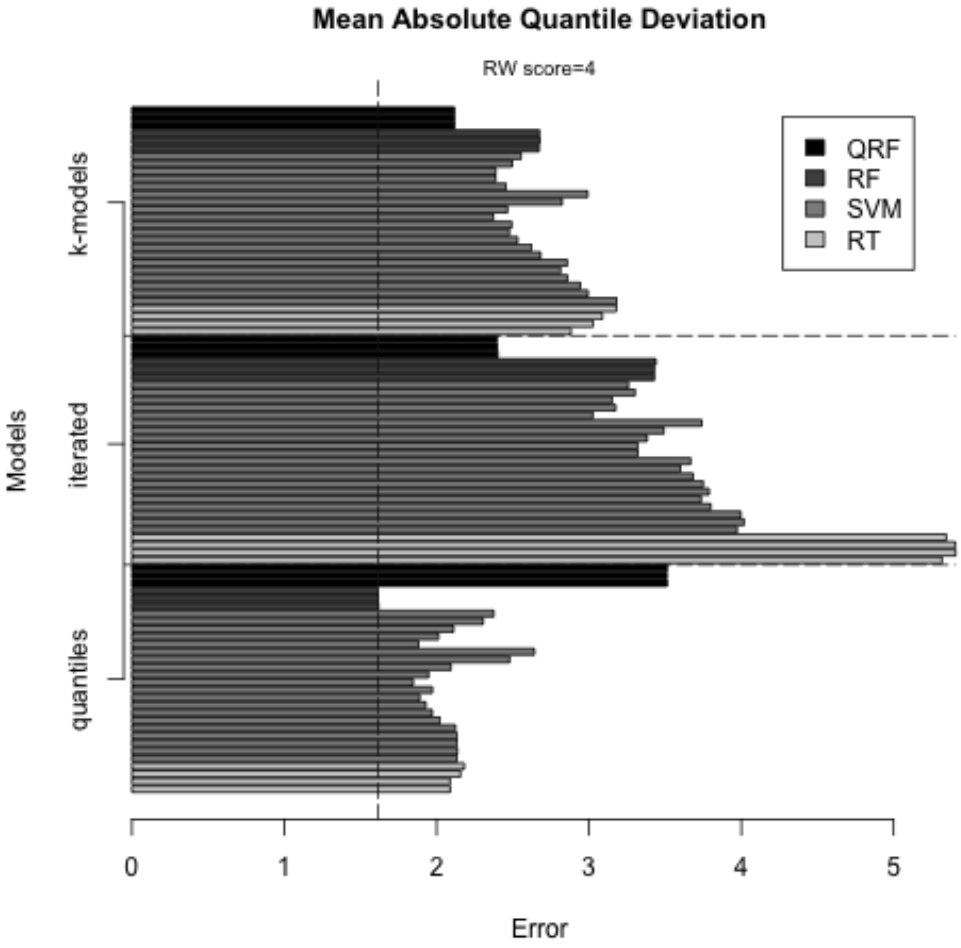


Figure A.4: Water Consumption, k = 24 and MAQ

Table A.4 shows all setups order by the MAQ error.

	MAQ.Q1	MAQ.Q3	MAQ
Quantile RF ntree 1000	1.25	1.99	1.62
Quantile RF ntree 1500	1.25	1.99	1.62
Quantile RF ntree 500	1.25	1.99	1.62
Quantile SVM cost 1 gamma 0.05	1.48	2.21	1.85
Quantile SVM cost 1 gamma 0.1	1.51	2.25	1.88
Quantile SVM cost 50 gamma 0.01	1.51	2.27	1.89
Quantile SVM cost 10 gamma 0.01	1.5	2.35	1.93
Quantile SVM cost 5 gamma 0.05	1.56	2.33	1.95
Quantile SVM cost 5 gamma 0.01	1.51	2.43	1.97
Quantile SVM cost 100 gamma 0.01	1.55	2.4	1.97
Quantile SVM cost 5 gamma 0.1	1.6	2.43	2.01
Quantile SVM cost 1 gamma 0.01	1.56	2.48	2.02
Quantile RT se 0.5	1.48	2.7	2.09
Quantile RT se 0	1.47	2.71	2.09
Quantile SVM cost 10 gamma 0.05	1.64	2.55	2.09
Quantile SVM cost 10 gamma 0.1	1.69	2.54	2.11
Nmodels QRF ntree 1500	1.88	2.36	2.12
Nmodels QRF ntree 1000	1.88	2.36	2.12
Nmodels QRF ntree 500	1.88	2.36	2.12
Quantile SVM cost 100 gamma 0.001	1.63	2.62	2.12
Quantile SVM cost 1 gamma 0.001	1.76	2.51	2.13
Quantile SVM cost 50 gamma 0.001	1.64	2.62	2.13
Quantile SVM cost 10 gamma 0.001	1.67	2.59	2.13
Quantile SVM cost 5 gamma 0.001	1.7	2.58	2.14
Quantile RT se 1	1.5	2.82	2.16
Quantile RT se 1.5	1.52	2.84	2.18
Quantile SVM cost 50 gamma 0.1	1.81	2.8	2.31
Nmodels SVM cost 5 gamma 0.05	2.08	2.67	2.37
Quantile SVM cost 100 gamma 0.1	1.82	2.93	2.38
Nmodels SVM cost 5 gamma 0.1	2.2	2.57	2.39

Nmodels SVM cost 10 gamma 0.1	2.2	2.57	2.39
Iterated QRF ntree 1500	2.38	2.41	2.4
Iterated QRF ntree 1000	2.38	2.41	2.4
Iterated QRF ntree 500	2.38	2.42	2.4
Nmodels SVM cost 1 gamma 0.1	2.28	2.63	2.46
Nmodels SVM cost 10 gamma 0.05	2.19	2.74	2.47
Nmodels SVM cost 100 gamma 0.01	2.15	2.81	2.48
Quantile SVM cost 50 gamma 0.05	1.97	2.99	2.48
Nmodels SVM cost 1 gamma 0.05	2.18	2.8	2.49
Nmodels SVM cost 50 gamma 0.1	2.23	2.76	2.5
Nmodels SVM cost 50 gamma 0.01	2.22	2.84	2.53
Nmodels SVM cost 100 gamma 0.1	2.3	2.8	2.55
Nmodels SVM cost 10 gamma 0.01	2.33	2.92	2.62
Quantile SVM cost 100 gamma 0.05	2.13	3.16	2.64
Nmodels RF ntree 500	2.95	2.4	2.67
Nmodels RF ntree 1500	2.95	2.4	2.67
Nmodels RF ntree 1000	2.96	2.4	2.68
Nmodels SVM cost 5 gamma 0.01	2.38	2.98	2.68
Nmodels SVM cost 100 gamma 0.001	2.49	3.14	2.81
Nmodels SVM cost 50 gamma 0.05	2.63	3.02	2.82
Nmodels SVM cost 1 gamma 0.01	2.46	3.25	2.86
Nmodels SVM cost 50 gamma 0.001	2.48	3.23	2.86
Nmodels RT se 0	3.05	2.71	2.88
Nmodels SVM cost 10 gamma 0.001	2.54	3.35	2.94
Nmodels SVM cost 100 gamma 0.05	2.77	3.21	2.99
Nmodels SVM cost 5 gamma 0.001	2.61	3.38	2.99
Iterated SVM cost 1 gamma 0.1	2.95	3.1	3.03
Nmodels RT se 0.5	3.4	2.66	3.03
Nmodels RT se 1	3.57	2.61	3.09
Iterated SVM cost 10 gamma 0.1	2.94	3.36	3.15
Iterated SVM cost 5 gamma 0.1	3.02	3.33	3.17

Nmodels RT se 1.5	3.74	2.62	3.18
Nmodels SVM cost 1 gamma 0.001	2.93	3.43	3.18
Iterated SVM cost 100 gamma 0.1	2.88	3.63	3.26
Iterated SVM cost 50 gamma 0.1	2.92	3.69	3.3
Iterated SVM cost 5 gamma 0.05	3.38	3.26	3.32
Iterated SVM cost 1 gamma 0.05	3.25	3.39	3.32
Iterated SVM cost 10 gamma 0.05	3.37	3.4	3.38
Iterated RF ntree 1000	4.04	2.82	3.43
Iterated RF ntree 500	4.03	2.82	3.43
Iterated RF ntree 1500	4.05	2.83	3.44
Iterated SVM cost 50 gamma 0.05	3.31	3.67	3.49
Quantile QRF ntree 1000	3.78	3.25	3.51
Quantile QRF ntree 1500	3.78	3.25	3.51
Quantile QRF ntree 500	3.78	3.25	3.51
Iterated SVM cost 50 gamma 0.01	3.63	3.57	3.6
Iterated SVM cost 100 gamma 0.01	3.66	3.67	3.67
Iterated SVM cost 10 gamma 0.01	3.66	3.71	3.68
Iterated SVM cost 100 gamma 0.001	3.37	4.11	3.74
Iterated SVM cost 100 gamma 0.05	3.62	3.86	3.74
Iterated SVM cost 5 gamma 0.01	3.63	3.87	3.75
Iterated SVM cost 1 gamma 0.01	3.14	4.44	3.79
RW	2.7	4.89	3.79
Iterated SVM cost 50 gamma 0.001	3.31	4.28	3.8
Iterated SVM cost 1 gamma 0.001	3.31	4.63	3.97
Iterated SVM cost 10 gamma 0.001	3.22	4.78	4
Iterated SVM cost 5 gamma 0.001	3.2	4.84	4.02
Iterated RT se 0	5.35	5.28	5.32
Iterated RT se 1.5	5.67	5.02	5.35
Iterated RT se 1	5.61	5.2	5.4
Iterated RT se 0.5	5.52	5.29	5.41

Table A.4: All setups, k = 24 and MAQ

A.2.2 TQE - Total Quantile Error

Figure A.5 shows the results using the TQE error measure for all different setups.

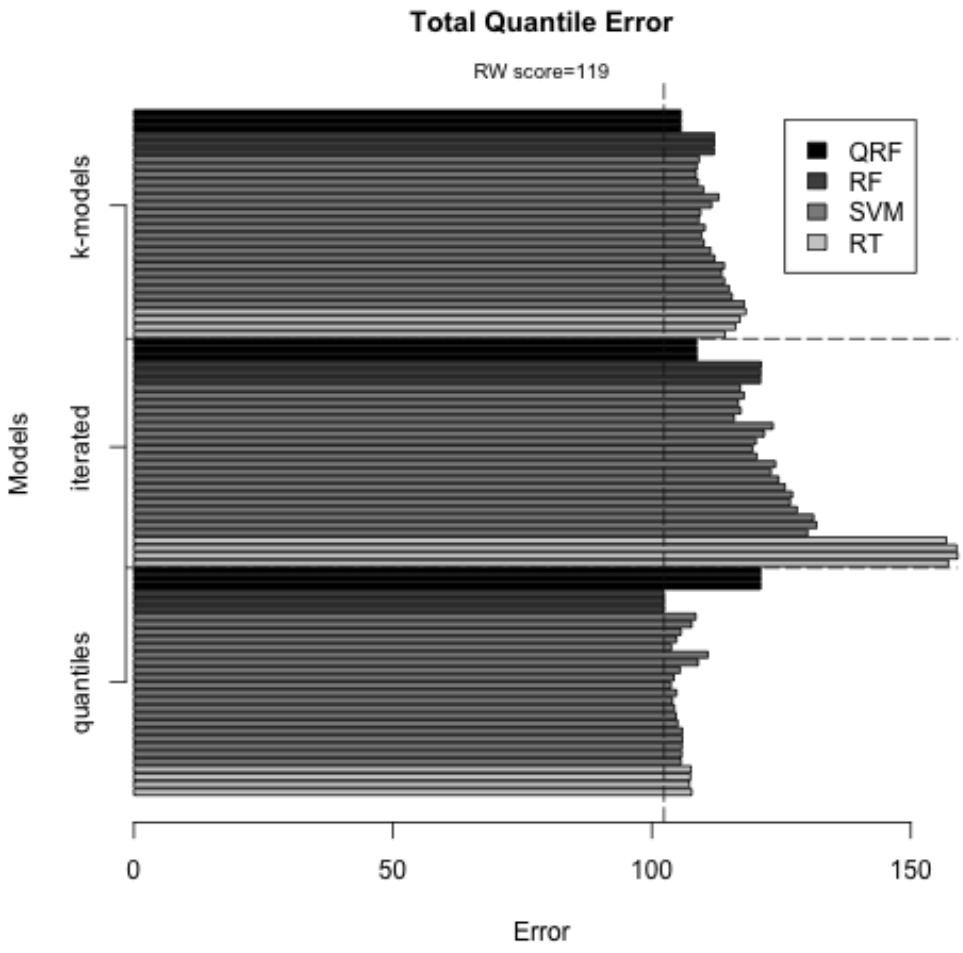


Figure A.5: Water Consumption, k = 24 and TQE

Table A.5 shows all setups order by the TQE error.

	QTE.Q1	QTE.Q3	QTE
Quantile RF ntree 1000	47.09	55.25	102.34
Quantile RF ntree 1500	47.09	55.25	102.34
Quantile RF ntree 500	47.09	55.26	102.35
Quantile SVM cost 1 gamma 0.05	48.06	55.62	103.68

Quantile SVM cost 1 gamma 0.1	48.15	55.6	103.75
Quantile SVM cost 50 gamma 0.01	48.19	55.78	103.97
Quantile SVM cost 5 gamma 0.05	48.32	55.94	104.26
Quantile SVM cost 10 gamma 0.01	48.06	56.31	104.37
Quantile SVM cost 5 gamma 0.01	48.06	56.6	104.66
Quantile SVM cost 5 gamma 0.1	48.53	56.23	104.76
Quantile SVM cost 100 gamma 0.01	48.35	56.42	104.76
Quantile SVM cost 1 gamma 0.01	48.23	56.77	105
Quantile SVM cost 10 gamma 0.05	48.61	56.83	105.44
Nmodels QRF ntree 1500	49.57	56.05	105.62
Nmodels QRF ntree 1000	49.57	56.06	105.62
Nmodels QRF ntree 500	49.57	56.06	105.62
Quantile SVM cost 10 gamma 0.1	48.95	56.68	105.63
Quantile SVM cost 1 gamma 0.001	49.04	56.6	105.64
Quantile SVM cost 5 gamma 0.001	48.86	56.91	105.77
Quantile SVM cost 10 gamma 0.001	48.75	57.05	105.81
Quantile SVM cost 100 gamma 0.001	48.55	57.32	105.87
Quantile SVM cost 50 gamma 0.001	48.61	57.32	105.93
Quantile RT se 0.5	48.5	58.73	107.23
Quantile RT se 1	48.52	58.95	107.47
Quantile RT se 1.5	48.59	59	107.59
Quantile SVM cost 50 gamma 0.1	49.56	58.14	107.7
Quantile RT se 0	48.51	59.19	107.7
Quantile SVM cost 100 gamma 0.1	49.65	58.83	108.48
Nmodels SVM cost 10 gamma 0.1	51.38	57.18	108.56
Iterated QRF ntree 1500	52.04	56.61	108.65
Iterated QRF ntree 1000	52.05	56.6	108.65
Nmodels SVM cost 50 gamma 0.1	51.1	57.59	108.69
Iterated QRF ntree 500	52.06	56.63	108.69
Nmodels SVM cost 5 gamma 0.1	51.63	57.3	108.93
Quantile SVM cost 50 gamma 0.05	50.09	58.86	108.95

Nmodels SVM cost 5 gamma 0.05	51.29	57.77	109.06
Nmodels SVM cost 100 gamma 0.1	51.37	57.75	109.12
Nmodels SVM cost 10 gamma 0.05	51.67	57.73	109.4
Nmodels SVM cost 100 gamma 0.01	51.41	58.14	109.55
Nmodels SVM cost 1 gamma 0.1	52.08	57.88	109.96
Nmodels SVM cost 50 gamma 0.01	51.65	58.42	110.06
Nmodels SVM cost 1 gamma 0.05	51.61	58.65	110.25
Quantile SVM cost 100 gamma 0.05	50.95	59.87	110.81
Nmodels SVM cost 10 gamma 0.01	52.23	59.11	111.34
Nmodels SVM cost 50 gamma 0.05	53.21	58.3	111.52
Nmodels RF ntree 500	55.46	56.56	112.02
Nmodels RF ntree 1500	55.49	56.55	112.04
Nmodels RF ntree 1000	55.51	56.55	112.06
Nmodels SVM cost 5 gamma 0.01	52.56	59.56	112.13
Nmodels SVM cost 100 gamma 0.05	53.78	59.14	112.92
Nmodels SVM cost 100 gamma 0.001	53.2	60.28	113.49
Nmodels SVM cost 1 gamma 0.01	52.92	61.06	113.98
Nmodels SVM cost 50 gamma 0.001	53.21	60.8	114.02
Nmodels RT se 0	56.26	57.89	114.15
Nmodels SVM cost 10 gamma 0.001	53.42	61.48	114.91
Nmodels SVM cost 5 gamma 0.001	53.8	61.69	115.48
Iterated SVM cost 1 gamma 0.1	55.89	60.02	115.91
Nmodels RT se 0.5	58.49	57.66	116.15
Iterated SVM cost 10 gamma 0.1	56.28	60.41	116.69
Nmodels RT se 1	59.55	57.46	117.01
Iterated SVM cost 100 gamma 0.1	55.62	61.46	117.08
Iterated SVM cost 5 gamma 0.1	56.64	60.55	117.19
Iterated SVM cost 50 gamma 0.1	55.97	61.83	117.81
Nmodels SVM cost 1 gamma 0.001	55.62	62.22	117.84
Nmodels RT se 1.5	60.61	57.54	118.14
RW	53.79	65.37	119.15

Iterated SVM cost 5 gamma 0.05	58.78	60.64	119.43
Iterated SVM cost 10 gamma 0.05	58.71	61.41	120.11
Iterated SVM cost 1 gamma 0.05	58.32	61.93	120.25
Iterated RF ntree 500	62.12	58.86	120.97
Quantile QRF ntree 500	60.27	60.72	120.99
Quantile QRF ntree 1500	60.27	60.73	121
Quantile QRF ntree 1000	60.29	60.71	121
Iterated RF ntree 1000	62.17	58.85	121.02
Iterated RF ntree 1500	62.23	58.89	121.11
Iterated SVM cost 50 gamma 0.05	59.36	62.36	121.72
Iterated SVM cost 50 gamma 0.01	60.41	62.59	123
Iterated SVM cost 100 gamma 0.05	60.14	63.25	123.39
Iterated SVM cost 100 gamma 0.01	60.51	63.35	123.86
Iterated SVM cost 10 gamma 0.01	60.83	63.64	124.47
Iterated SVM cost 5 gamma 0.01	60.87	64.85	125.71
Iterated SVM cost 100 gamma 0.001	59.25	67.47	126.72
Iterated SVM cost 1 gamma 0.01	57.91	69.21	127.12
Iterated SVM cost 50 gamma 0.001	59.38	68.66	128.04
Iterated SVM cost 1 gamma 0.001	59.37	70.69	130.05
Iterated SVM cost 10 gamma 0.001	59.37	71.86	131.23
Iterated SVM cost 5 gamma 0.001	59.22	72.6	131.82
Iterated RT se 1.5	84.76	72.1	156.86
Iterated RT se 0	81.65	75.58	157.23
Iterated RT se 1	84.53	74.31	158.84
Iterated RT se 0.5	83.64	75.4	159.04

Table A.5: All setups, k = 24 and TQE

A.2.3 Utility

Figure A.6 shows the results using the Utility error measure (larger is better) for all different setups.

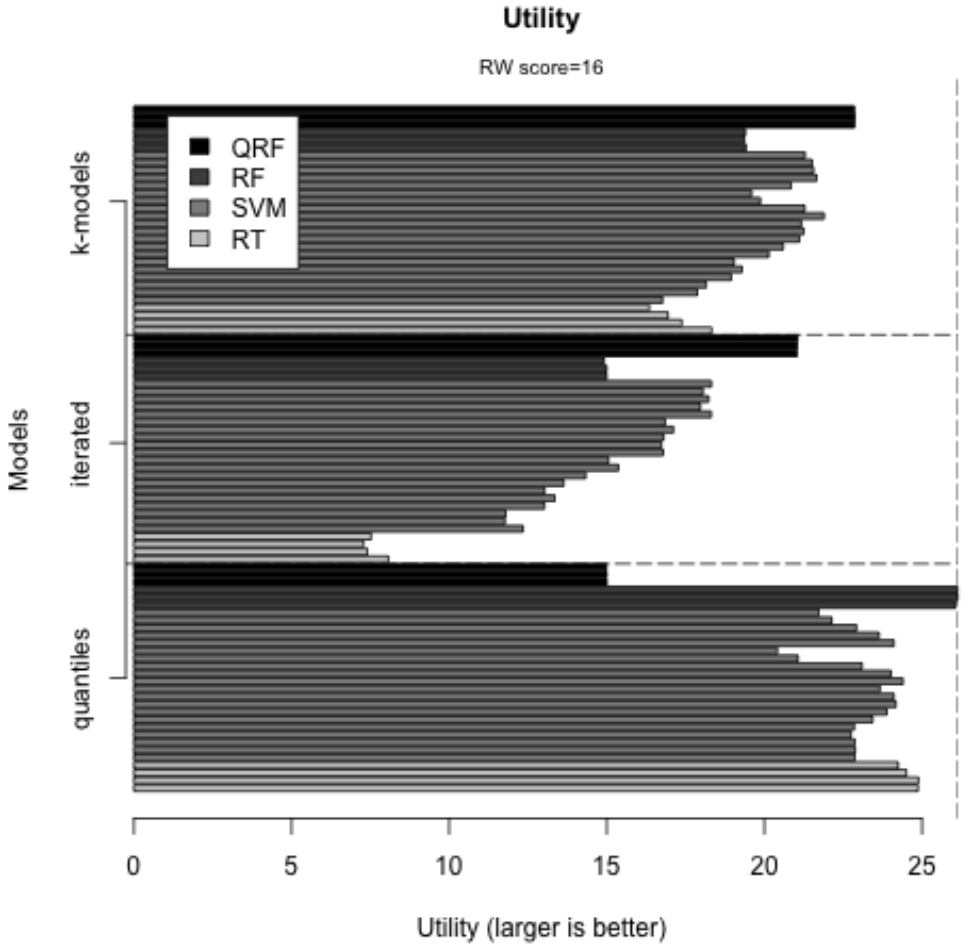


Figure A.6: Water Consumption, k = 24 and Utility

Table A.6 shows all setups order by the Utility.

	utility
Quantile RF ntree 1000	26.13
Quantile RF ntree 1500	26.12
Quantile RF ntree 500	26.06

Quantile RT se 0.5	24.89
Quantile RT se 0	24.87
Quantile RT se 1	24.5
Quantile SVM cost 1 gamma 0.05	24.4
Quantile RT se 1.5	24.25
Quantile SVM cost 10 gamma 0.01	24.17
Quantile SVM cost 1 gamma 0.1	24.11
Quantile SVM cost 50 gamma 0.01	24.11
Quantile SVM cost 5 gamma 0.05	24.01
Quantile SVM cost 5 gamma 0.01	23.88
Quantile SVM cost 100 gamma 0.01	23.66
Quantile SVM cost 5 gamma 0.1	23.64
Quantile SVM cost 1 gamma 0.01	23.43
Quantile SVM cost 10 gamma 0.05	23.1
Quantile SVM cost 10 gamma 0.1	22.92
Quantile SVM cost 5 gamma 0.001	22.88
Quantile SVM cost 10 gamma 0.001	22.87
Nmodels QRF ntree 1000	22.87
Quantile SVM cost 1 gamma 0.001	22.87
Nmodels QRF ntree 500	22.86
Nmodels QRF ntree 1500	22.85
Quantile SVM cost 100 gamma 0.001	22.85
Quantile SVM cost 50 gamma 0.001	22.75
Quantile SVM cost 50 gamma 0.1	22.13
Nmodels SVM cost 5 gamma 0.05	21.89
Quantile SVM cost 100 gamma 0.1	21.73
Nmodels SVM cost 5 gamma 0.1	21.66
Nmodels SVM cost 10 gamma 0.1	21.56
Nmodels SVM cost 50 gamma 0.1	21.51
Nmodels SVM cost 100 gamma 0.1	21.29
Nmodels SVM cost 10 gamma 0.05	21.28

Nmodels SVM cost 100 gamma 0.01	21.24
Nmodels SVM cost 1 gamma 0.05	21.17
Nmodels SVM cost 50 gamma 0.01	21.11
Quantile SVM cost 50 gamma 0.05	21.07
Iterated QRF ntree 1500	21.05
Iterated QRF ntree 1000	21.04
Iterated QRF ntree 500	21.03
Nmodels SVM cost 1 gamma 0.1	20.85
Nmodels SVM cost 10 gamma 0.01	20.6
Quantile SVM cost 100 gamma 0.05	20.41
Nmodels SVM cost 5 gamma 0.01	20.14
Nmodels SVM cost 50 gamma 0.05	19.87
Nmodels SVM cost 100 gamma 0.05	19.6
Nmodels RF ntree 500	19.41
Nmodels RF ntree 1500	19.4
Nmodels RF ntree 1000	19.37
Nmodels SVM cost 100 gamma 0.001	19.29
Nmodels SVM cost 1 gamma 0.01	19.04
Nmodels SVM cost 50 gamma 0.001	18.95
Nmodels RT se 0	18.32
Iterated SVM cost 100 gamma 0.1	18.32
Iterated SVM cost 1 gamma 0.1	18.31
Iterated SVM cost 10 gamma 0.1	18.23
Nmodels SVM cost 10 gamma 0.001	18.15
Iterated SVM cost 50 gamma 0.1	18.05
Iterated SVM cost 5 gamma 0.1	17.97
Nmodels SVM cost 5 gamma 0.001	17.88
Nmodels RT se 0.5	17.38
Iterated SVM cost 50 gamma 0.05	17.11
Nmodels RT se 1	16.94
Iterated SVM cost 100 gamma 0.05	16.85

Iterated SVM cost 10 gamma 0.05	16.81
Iterated SVM cost 1 gamma 0.05	16.79
Nmodels SVM cost 1 gamma 0.001	16.77
Iterated SVM cost 5 gamma 0.05	16.73
Nmodels RT se 1.5	16.35
RW	15.72
Iterated SVM cost 50 gamma 0.01	15.37
Iterated SVM cost 100 gamma 0.01	15.06
Quantile QRF ntree 500	15
Quantile QRF ntree 1000	15
Iterated RF ntree 500	15
Iterated RF ntree 1000	15
Quantile QRF ntree 1500	14.99
Iterated RF ntree 1500	14.91
Iterated SVM cost 10 gamma 0.01	14.33
Iterated SVM cost 5 gamma 0.01	13.63
Iterated SVM cost 100 gamma 0.001	13.34
Iterated SVM cost 1 gamma 0.01	13.03
Iterated SVM cost 50 gamma 0.001	13.02
Iterated SVM cost 1 gamma 0.001	12.34
Iterated SVM cost 10 gamma 0.001	11.8
Iterated SVM cost 5 gamma 0.001	11.77
Iterated RT se 0	8.07
Iterated RT se 1.5	7.52
Iterated RT se 0.5	7.41
Iterated RT se 1	7.28

Table A.6: All setups, k = 24 and Utility

References

Agrawal, A., Goyal, P., and Diwakar, S. (2010). Fast and enhanced algorithm for exemplar based image inpainting. In *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pages 325–330. IEEE.

Akgiray, V. (1989). Conditional heteroscedasticity in time series of stock returns: evidence and forecasts. *Journal of business*, pages 55–80.

Alexiadis, M., Dokopoulos, P., and Sahsamanoglou, H. (1999). Wind speed and power forecasting based on spatial correlation models. *Energy Conversion, IEEE Transactions on*, 14(3):836–842.

Andersen, T., Bollerslev, T., and Lange, S. (1999). Forecasting financial market volatility: Sample frequency vis-a-vis forecast horizon. *Journal of Empirical Finance*, 6(5):457–477.

Andrienko, G., Malerba, D., May, M., and Teisseire, M. (2006). Mining spatio-temporal data. *Journal of Intelligent Information Systems*, 27(3):187–190.

Assaad, M., Boné, R., and Cardot, H. (2008). A new boosting algorithm for improved time-series forecasting with recurrent neural networks. *Information Fusion*, 9(1):41 – 55. Special Issue on Applications of Ensemble Methods.

Barthelmie, R., Murray, F., and Pryor, S. (2008). The economic benefit of short-term forecasting for wind energy in the uk electricity market. *Energy Policy*, 36(5):1687–1696.

Basu, S., Mukherjee, A., and Klivansky, S. (1996). Time series models for internet traffic. In *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, volume 2, pages 611–620. IEEE.

Benevenuto, F., Magno, G., Rodrigues, T., and Almeida, V. (2010). Detecting spammers on twitter. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, volume 6.

Bermolen, P. and Rossi, D. (2009). Support vector regression for link load prediction. *Computer Networks*, 53(2):191–201.

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co.

Bilgili, M., Sahin, B., and Yasar, A. (2007). Application of artificial neural networks for the wind speed prediction of target station using reference stations data. *Renewable Energy*, 32(14):2350–2360.

Box, G., Jenkins, G., and Reinsel, G. (1976). *Time series analysis*. Holden-day San Francisco.

Boykin, P. and Roychowdhury, V. (2005). Leveraging social networks to fight spam. *Computer*, 38(4):61–68.

Breiman, L. (1984). *Classification and regression trees*. Chapman & Hall/CRC.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Bremnes, J. (2004). Probabilistic wind power forecasts using local quantile regression. *Wind Energy*, 7(1):47–54.

Bridson, M. and Haefliger, A. (1999). *Metric spaces of non-positive curvature*, volume 319. Springer Verlag.

Brown, B., Katz, R., and Murphy, A. (1984). Time series models to simulate and forecast wind speed and wind power. *Journal of Climate and Applied Meteorology*, 23:1184–1195.

Brunsdon, C., Fotheringham, S., and Charlton, M. (1998). Geographically weighted regression-modelling spatial non-stationarity. *Journal of the Royal Statistical Society. Series D*, 47:431–443.

Cao, H., Mamoulis, N., and Cheung, D. (2005). Mining frequent spatio-temporal sequential patterns. In *Fifth IEEE International Conference on Data Mining*, page 8.

Ceci, M., Appice, A., and Malerba, D. (2010). Transductive learning for spatial data classification. In *Advances in Machine Learning I*, pages 189–207. Springer.

Chatfield, C. (1993). Prediction intervals. *Journal of Business and Economic Statistics*, 11:121–135.

Chatfield, C. (2001). *Time-series forecasting.* CRC Press.

Chatfield, C. (2004). *The Analysis of Time Series, an introduction.* Chapman & Hall/CRC.

Cheng, T. and Wang, J. (2008). Integrated spatio-temporal data mining for forest fire prediction. *Transactions in GIS*, 12(5):591–611.

Ciampi, A., Appice, A., and Malerba, D. (2010). Discovering trend-based clusters in spatially distributed data streams. In *International Workshop of Mining Ubiquitous and Social Environments*, pages 107–122.

Compieta, P., Di Martino, S., Bertolotto, M., Ferrucci, F., and Kechadi, T. (2007). Exploratory spatio-temporal data mining and visualization. *Journal of Visual Languages and Computing*, 18(3):255–279.

Contreras, J., Espinola, R., Nogales, F., and Conejo, A. (2003). Arima models to predict next-day electricity prices. *Power Systems, IEEE Transactions on*, 18(3):1014–1020.

Cowles, A. (1933). Can stock market forecasters forecast? *Econometrica: Journal of the Econometric Society*, pages 309–324.

Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support Vector Machines: and other kernel-based learning methods.* Cambridge Univ Pr.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., and Weingessel, A. (2009). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien.* R package version 1.5-22.

Dokko, Y. and Edelstein, R. (1989). How well do economists forecast stock market prices? a study of the livingston surveys. *The American Economic Review*, 79(4):865–871.

Drucker, H., Wu, D., and Vapnik, V. (1999). Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*, 10(5):1048–1054.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence*, pages 973–978. Citeseer.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors (1996). *Advances in knowledge discovery and data mining.* American Association for Artificial Intelligence, Menlo Park, CA, USA.

Goovaerts, P. (2000). Geostatistical approaches for incorporating elevation into the spatial interpolation of rainfall. *Journal of hydrology*, 228(1):113–129.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182.

Hall, S. and Mitchell, J. (2007). Combining density forecasts. *International Journal of Forecasting*, 23(1):1–13.

Hays, J. and Efros, A. (2007). Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 4. ACM.

Hiemstra, P., Pebesma, E., Twenhöfel, C., and Heuvelink, G. (2008). Real-time automatic interpolation of ambient gamma dose rates from the dutch radioactivity monitoring network. *Computers & Geosciences*.

Hipel, K. and McLeod, A. (1994). *Time series modelling of water resources and environmental systems*, volume 45. Elsevier Science Ltd.

Hyndman, R. J. (2011). *forecast: Forecasting functions for time series*. R package version 3.11.

Irani, M., Anandan, P., and Hsu, S. (1995). Mosaic based representations of video sequences and their applications. In *Proceedings of the Fifth International Conference on Computer Vision*, ICCV '95, pages 605–. IEEE Computer Society.

Isaaks, E. and Srivastava, R. (1989). *Applied geostatistics*, volume 2. Oxford University Press New York.

Isengildina-Massa, O., Irwin, S., and Good, D. (2008). Quantile regression methods of estimating confidence intervals for wasde price forecasts. In *2008 Annual Meeting, July 27-29, 2008, Orlando, Florida*. American Agricultural Economics Association (New Name 2008: Agricultural and Applied Economics Association).

Jain, A. and Kumar, A. (2007). Hybrid neural network models for hydrologic time series forecasting. *Applied Soft Computing*, 7(2):585–592.

Joskow, P. and Kahn, E. (2001). A quantitative analysis of pricing behavior in california's wholesale electricity market during summer 2000. Technical report, National Bureau of Economic Research.

Kantardzic, M. (2011). *Data mining: concepts, models, methods, and algorithms*. Wiley-IEEE Press.

Kavasseri, R. and Seetharaman, K. (2009). Day-ahead wind speed forecasting using f-arima models. *Renewable Energy*, 34(5):1388–1393.

Khashei, M. and Bijari, M. (2011). A new hybrid methodology for nonlinear time series forecasting. *Modelling and Simulation in Engineering*, 2011:15.

Khashei, M., Reza Hejazi, S., and Bijari, M. (2008). A new hybrid artificial neural networks and fuzzy regression model for time series forecasting. *Fuzzy Sets and Systems*, 159(7):769–786.

Khosravi, A., Mazloumi, E., Nahavandi, S., Creighton, D., and Van Lint, J. (2011). Prediction intervals to account for uncertainties in travel time prediction. *Intelligent Transportation Systems, IEEE Transactions on*, 12(2):537–547.

Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319.

Kira, K. and Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the National Conference on Artificial Intelligence*, pages 129–129. John Wiley & Sons Ltd.

Koenker, R. (2005). *Quantile Regression*. Econometric Society Monograph Series. Cambridge University Press.

Koperski, K., Han, J., and Adhikary, J. (1998). Mining knowledge in geographical data. *Communications of the ACM*.

Krige, D. (1951). *A statistical approach to some mine valuation and allied problems on the Witwatersrand*. Univ. of the Witwatersrand.

Kusiak, A., Zheng, H., and Song, Z. (2009). Short-term prediction of wind farm power: A data mining approach. *Energy Conversion, IEEE Transactions on*, 24(1):125–136.

Lauritzen, S. (1981). Time series analysis in 1880: A discussion of contributions made by tn thiele. *International Statistical Review/Revue Internationale de Statistique*, pages 319–331.

Lee, Y. and Tong, L. (2011). Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. *Knowledge-Based Systems*, 24(1):66–72.

Lei, M., Shiyan, L., Chuanwen, J., Hongling, L., and Yan, Z. (2009). A review on the forecasting of wind speed and generated power. *Renewable and Sustainable Energy Reviews*, 13(4):915–920.

Li, G. and Shi, J. (2010). On comparing three artificial neural networks for wind speed forecasting. *Applied Energy*, 87(7):2313–2320.

Li, J. and Heap, A. (2010). A review of comparative studies of spatial interpolation methods in environmental sciences: Performance and impact factors. *Ecological Informatics*.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

Lim, C. and McAleer, M. (2002). Time series forecasts of international travel demand for australia. *Tourism Management*, 23(4):389 – 396.

Lin, G. and Chen, L. (2004). A spatial interpolation method based on radial basis function networks incorporating a semivariogram model. *Journal of Hydrology*, 288(3):288–298.

Lu, C., Lee, T., and Chiu, C. (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, 47(2):115–125.

Lu, G. and Wong, D. (2008). An adaptive inverse-distance weighting spatial interpolation technique. *Computers & Geosciences*, 34(9):1044–1055.

Malerba, D. (2008). A relational perspective on spatial data mining. *International Journal of Data Mining, Modelling and Management*, 1(1):103–118.

Malerba, D., Ceci, M., and Appice, A. (2005). Mining model trees from spatial data. In *Knowledge Discovery in Databases: PKDD 2005*, pages 169–180. Springer.

Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., and Cheung, D. (2004). Mining, indexing, and querying historical spatiotemporal data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 236–245. ACM New York, NY, USA.

Matteson, D., McLean, M., Woodard, D., and Henderson, S. (2011). Forecasting emergency medical service call arrival rates. *The Annals of Applied Statistics*, 5(2B):1379–1406.

Mazloumi, E., Rose, G., Currie, G., and Moridpour, S. (2011). Prediction intervals to account for uncertainties in neural network predictions: Methodology and application in bus travel time prediction. *Engineering Applications of Artificial Intelligence*, 24(3):534–542.

Meinshausen, N. (2006). Quantile regression forests. *The Journal of Machine Learning Research*, 7:999.

Meinshausen, N. (2007). *quantregForest: Quantile Regression Forests*. R package version 0.2-2.

Ming-yao, Q., Li-xin, M., and Jie, S. (2009). A spatio-temporal distance based two-phase heuristic algorithm for vehicle routing problem. In *Fifth International Conference on Natural Computation, ICNC'09*, pages 352–357. IEEE.

Mitas, L. and Mitasova, H. (1999). Spatial interpolation. *Geographical Information Systems: Principles, Techniques, Management and Applications, Wiley*, 481.

Mohandes, M., Halawani, T., Rehman, S., and Hussain, A. (2004). Support vector machines for wind speed prediction. *Renewable Energy*, 29(6):939–947.

Nielsen, H., Madsen, H., and Nielsen, T. (2006). Using quantile regression to extend an existing wind power forecasting system with probabilistic forecasts. *Wind Energy*, 9(1-2):95–108.

Nogales, F., Contreras, J., Conejo, A., and Espínola, R. (2002). Forecasting next-day electricity prices by time series models. *Power Systems, IEEE Transactions on*, 17(2):342–348.

O'Connor, J. and Robertson, E. F. (2012). The mactutor history of mathematics archive. `http://www-history.mcs.st-and.ac.uk/Chronology/1870_1880.html`. Accessed: 09/05/2012.

Pankratz, A. (1983). *Forecasting with univariate Box-Jenkins models*, volume 3. Wiley Online Library.

Perlich, C., Rosset, S., Lawrence, R., and Zadrozny, B. (2007). High-quantile modeling for customer wallet estimation and other applications. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, pages 977–985.

Pinson, P. and Kariniotakis, G. (2010). Conditional prediction intervals of wind power generation. *Power Systems, IEEE Transactions on*, 25(4):1845–1856.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing.

Rigol, J., Jarvis, C., and Stuart, N. (2001). Artificial neural networks as a tool for spatial interpolation. *International Journal of Geographical Information Science*, 15(4):323–343.

Robinson, T. and Metternicht, G. (2006). Testing the performance of spatial interpolation techniques for mapping soil properties. *Computers and Electronics in Agriculture*, 50(2):97 – 108.

Rokach, L. and Maimon, O. (2008). *Data mining with decision trees: theroy and applications*, volume 69. World Scientific Publishing Company Incorporated.

Sapankevych, N. and Sankar, R. (2009). Time series prediction using support vector machines: a survey. *Computational Intelligence Magazine, IEEE*, 4(2):24–38.

Serguieva, A. and Hunter, J. (2004). Fuzzy interval methods in investment risk appraisal. *Fuzzy Sets and Systems*, 142(3):443–466.

Sfetsos, A. (2000). A comparison of various forecasting techniques applied to mean hourly wind speed time series. *Renewable Energy*, 21(1):23–35.

Shi, J., Guo, J., and Zheng, S. (2012). Evaluation of hybrid forecasting approaches for wind speed and power generation time series. *Renewable and Sustainable Energy Reviews*, 16(5):3471–3480.

Shih, T. and Chang, R. (2005). Digital inpainting-survey and multilayer image inpainting algorithms. In *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on*, volume 1, pages 15–24. IEEE.

Snavely, N., Seitz, S., and Szeliski, R. (2006). Photo tourism: exploring photo collections in 3d. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 835–846. ACM.

Takens, F. (1981). Detecting strange attractors in turbulence. *Dynamical systems and turbulence Warwick 1980*, 898(1):366–381.

Tay, A. S. and Wallis, K. F. (2000). Density forecasting: a survey. *Journal of Forecasting*, 19(4):235–254.

Tay, F. and Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, 29(4):309–317.

Taylor, J. (2006). Density forecasting for the efficient balancing of the generation and consumption of electricity. *International Journal of Forecasting*, 22:707–724.

Taylor, J., McSharry, P., and Buizza, R. (2009a). Wind power density forecasting using ensemble predictions and time series models. *Energy Conversion, IEEE Transactions on*, 24(3):775 –782.

Taylor, J., McSharry, P., and Buizza, R. (2009b). Wind power density forecasting using ensemble predictions and time series models. *IEEE Transactions on Energy Conversion*, 24:775–782.

Therneau, T. M. and port by B. Ripley., B. A. R. (2009). *rpart: Recursive Partitioning*. R package version 3.1-44.

Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46.

Torgo, L. (2010). *Data Mining with R: Learning with Case Studies*. Chapman & Hall/Crc Data Mining and Knowledge Discovery Series. Chapman & Hall/CRC.

Umer, M., Kulik, L., and Tanin, E. (2010). Spatial interpolation in wireless sensor networks: localized algorithms for variogram modeling and kriging. *Geoinformatica*, 14(1):101–134.

Weron, R. and Misiorek, A. (2008). Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models. *International Journal of Forecasting*, 24(4):744 – 763. <ce:title>Energy Forecasting</ce:title>.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Wu, W., Xu, Z., and Wang, Y. (2006). Robust prediction of network traffic using quantile regression models. In *2006 IEEE International Conference on Information Reuse and Integration*, pages 220–225.

Xie, Y., Chen, T., Lei, M., Yang, J., Guo, Q., Song, B., and Zhou, X. (2011). Spatial distribution of soil heavy metal pollution estimated by different interpolation methods: Accuracy and uncertainty analysis. *Chemosphere*, 82(3):468–476.

Yao, X. (2003). Research issues in spatio-temporal data mining. In *White paper submitted to the University Consortium for Geographic Information Science (UCGIS) workshop on Geospatial Visualization and Knowledge Discovery, Lansdowne, Virginia, Nov*, pages 18–20.

Zhang, G. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175.

Zhang, L. and Luh, P. (2005). Neural network-based market clearing price prediction and confidence interval estimation with an improved extended kalman filter method. *Power Systems, IEEE Transactions on*, 20(1):59–66.

Zhao, P., Wang, J., Xia, J., Dai, Y., Sheng, Y., and Yue, J. (2011). Performance evaluation and accuracy enhancement of a day-ahead wind power forecasting system in china. *Renewable Energy*.

Zhou, B., He, D., Sun, Z., and Ng, W. (2005). Network traffic modeling and prediction with arima/garch. In *Proc. of Third International Working Conference: performance modelling and evaluation of heterogeneous networks*. Citeseer.