GIMA
Geographical Information Management and Applications

Exploring the value of the Bregman Block Average Co-clustering algorithm for missing value imputation in geo-referenced time series

Final Report

Joris Timmermans
4140214
j.m.timmermans@students.uu.nl

Supervisor:
Raul Zurita-Milla
r.zurita-milla@utwente.nl

Responsible professor:
Menno-Jan Kraak
m.j.kraak@utwente.nl

Universiteit Utrecht
TU Delft
UNIVERSITY OF TWENTE.
WAGENINGEN UR

Abstract

Introduction

Missing values frequently introduce loss of information in spatial analysis. A common approach to manage missing values is to impute missing values. This is often done by using spatial interpolation models, and more recently machine learning methods. The Bregman Block Average Co-clustering algorithm with I-Divergence (BBAC-I) has recently been applied to explore spatial patterns. Among other things, the original authors of this algorithm used it for missing value imputation. This thesis explored the value of the BBAC-I algorithm in missing value imputation of Geo-referenced time series.

Methods

This model comparison study compared the imputation value of a selection of machine learning and spatial interpolation models to the BBAC-I models on four data sets with distinctly different spatial characteristics. Three objectives were set to explore the BBAC-I algorithm in this context: (1) Compare the prediction accuracy, (2) compare the computational run time, (3) analyze the spatial properties of the prediction residuals.

Results and Conclusion

BBAC-I produced less accurate results than the selection of Machine learning models, but produced more accurate than spatial interpolation methods. The BBAC-I run time was faster than any other model, especially for larger data sets. However, it did consistently produce positively spatially correlated residuals. The value of BBAC-I for missing value imputation lies in a limited selection of data sets that are very large, and for which limiting computational requirements is more important than accuracy. Future research should continue to address the value of recently developed non spatial models in the spatial domain.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and problem introduction

Missing values frequently introduce loss of information in spatial analysis (Baker, White, & Mengersen, 2014; Hibbert et al., 2009; Zhang et al., 2017; Kornelsen & Coulibaly, 2014). A common approach to manage missing values is to remove incomplete entries. However, this could lead to the loss of useful information (Donders, van der Heijden, Stijnen, & Moons, 2006; Haining, 2003, p. 154). A more desired approach is to impute missing values with a well-grounded prediction. Successful implementation of data imputation prevents loss of information. Spatial interpolation is often used to impute spatial data (Lam, 1983; Smith, Goodchild, & Longley, 2018; Haining, 2003, pp.154-160), but does not always perform well on spatio-temporal data (Li & Revesz, 2004; Zhang et al., 2017). Li and Revesz (2004) state that there are surprisingly few publications that discuss interpolating spatio-temporal data. The models that specifically target spatio-temporal data are often computationally complex (e.g., Zhang et al., 2017) and do not scale effectively on large data sets (e.g., Teegavarapu, 2014; Feng, Nowak, O'Neill, & Welsh, 2014). Outside of geographic information science (GIS), methods such as Singular Value Decomposition (SVD) (Paterek, 2007; Brand, 2002; Donders et al., 2006), Non Negative Matrix Factorization (NNMF) (Luo, Zhou, Xia, & Zhu, 2014; Zhu, 2016) and more computationally complex machine learning models (Donders et al., 2006; Honaker & King, 2010; Witten et al., 1999, p. 58) are used to impute temporal data.

Recent work from Wu, Zurita-Milla, and Kraak (2016), Wu (2016), Wu, Zurita-Milla, and Kraak (2015) indicates that the Bregman Block average co-clustering algorithm with I-Divergence (BBAC-I) is a fast, scalable algorithm that identifies spatio-temporal co-clusters. The authors that originally presented the BBAC-I algorithm explored the use of six different co-clustering schemes, and highlighted its use in non-spatial data imputation. They found that scheme two and five provide comparable accuracy to SVD and NNMF, but offer significant benefits in training time and computational efforts (George, 2005; Banerjee, Dhillon, Ghosh, Merugu, & Modha, 2007, pp. 1959-1960).

The BBAC-I algorithm identifies both spatial and temporal patterns simultaneously, whereas regular clustering algorithms identify either spatial or temporal patterns. Respectively referred to as co-clustering and one-way clustering. Unlike one-way clustering, co-clustering groups similar values along both axes of a matrix (Wu, 2016, pp. 4-6). Thus, in a geo-referenced time serie (GTS) both timestamps and locations are treated equally, resulting in homogeneous spatio-temporal co-clusters. GTSs are structured spatio-temporal data sets that contain sequential values for fixed locations at a fixed temporal interval (Guo, Chen, MacEachren, & Liao, 2006). For example, daily updated weather stations, weekly collected remote sensing imagery, and yearly collected census data per zip code. A schematic example is displayed in figure 1.1.

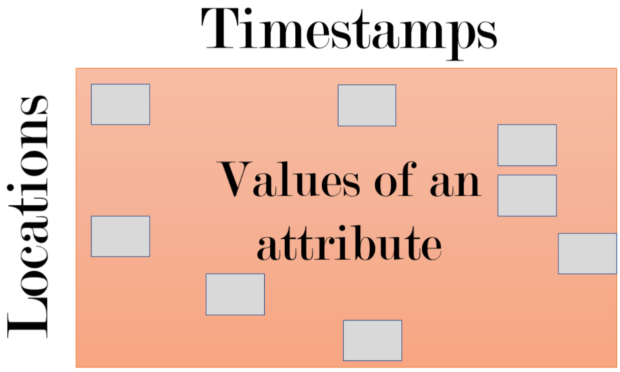


Figure 1.1: A schematic representation of a geo-referenced time serie, grey squares indicate missing values. Adapted from Wu (2016, p. 4)

1.2 Research Questions and Objectives

The BBAC-I algorithm has proven to be a able to recognize patterns in phenological and meteorological GTSs. The question remains if it could function as an imputation method on GTSs. The purpose of this thesis is to perform a first exploration of the value of the BBAC-I in missing data imputation in GTSs, and empirically compare the results with both spatio-temporal interpolation and machine learning models. The main research question of this thesis is:

> *To what extent can the Bregman Block average co-clustering algorithm schemes two and five with I-Divergence be used to impute missing data in geo-referenced time series?*

An answer to the research question will be formulated by analyzing several GTSs, in which data is removed, but the actual values remain know. After applying missing value imputation, the difference between predicted and actual values can be analyzed for each method. This is standard procedure in the machine learning domain to test the applicability of an algorithm (Hastie, Tibshirani, & Friedman, 2009, p. 24). The main research question is supported by three sub-questions based on their respective objectives:

- How does the prediction accuracy of the BBAC-I algorithm compare to the state of the art spatial interpolation and machine learning models?
- To what extent does the computational run-time from BBAC-I differ from the selected spatial interpolation and machine learning models?
- To what extent are the residuals spatially dependent, and to what extent does this differ from spatial and non spatial models?

The objective of the first question is to explore the accuracy of the BBAC-I algorithm in missing data imputation by analyzing the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE), and comparing the results to other models. The MAE and RMSE are frequently used metrics to measure model accuracy by analyzing the difference between predicted and actual values (Hastie et al., 2009, p. 24). These metrics are discussed more thoroughly in the methodology chapter.

The objective of the second question is to investigate the computational effectiveness of the BBAC-I algorithm compared to other models. The BBAC-I algorithm is thought to be scalable and fast relative to other imputation models (Banerjee et al., 2007; George, 2005). However, this has not been empirically proven with regard to spatial missing value imputation. The computational complexity of each method is measured by comparing the run time of various inputs with different sizes.

The objective of the third question is to explore the usability of BBAC-I for missing data value imputation for data sets with a high level of spatial auto-correlation. A high spatial autocorrelation between residuals indicates that the algorithm is not suitable to impute data sets with strong spatial patterns. These results are compared to both spatial and non-spatial imputation models and are used explore the value of BBAC-I as an spatio-temporal imputation method.

1.3 Relevance

Since Banerjee et al. (2007) published their work on the BBAC-I algorithm, it has been the subject of many recommender system publications (e.g., George, 2005; Deodhar & Ghosh, 2007; Kwon & Cho, 2010a), but its missing value imputation potential remains unexplored in other domains, including GIS. Despite the emphasis from Banerjee et al. (2007, pp. 1959-1960) that missing value prediction is an important task in many real-world domains, and that the models in their work should be tried and tested in other real-world domains.

The same applies to other statistical models: many geo-information scholars state that comparing statistical models from both the geographical and machine learning perspectives could lead to a better understanding of the advantages and disadvantages of machine learning applications in GIS (Yuan, 2017; Lloyd & Atkinson, 2010; Kanevski, Pozdnoukhov, & Timonin, 2009; Tsou, 2015). Predominantly interesting is the ability of those applications to identify spatial patterns. Most statistical models assume that measurements are independent (Moran, 2016). This is in contrast with datasets that have positive spatial autocorrelation, in which measurements that are geographically closer are more alike. Clearly those measurements are not independent (Stojanova, Ceci, Appice, Malerba, & Džeroski, 2013; Wilhelmsson, 2002; Dubin, 1998) Inappropriate usage of data with a positive spatial autocorrelation can obscure valuable information. Missing value imputation relies on the same principles as most statistical models (Donders et al., 2006; Honaker & King, 2010; Little, Rubin, & B., 2002, pp. 3-6), therefore it is of value to analyze the spatial properties of both BBAC-I and other imputation models.

1.4 Research limitations

The main purpose of this work is to explore the value of the BBAC-I algorithm in missing value imputation of GTSs. The comparison with other models is to validate the usability of the BBAC-I algorithm. Nowadays there are a large amount of spatial interpolation models and machine learning models. Therefore a well grounded selection will be made that includes relevant models. The selection is based on the theoretical chapter and discussed in chapter 3.

Furthermore, this work only explores the value of these models and models in GTSs. The application of the BBAC-I algorithm on unstructured spatial temporal datasets has not yet been explored. Before its application in missing value imputation on unstructured data can be explored, the values as a co-clustering technique on this type of data has to be analysed as previously done on structured data by Wu (2016). Nonetheless it is a scientifically interesting topic, but it is out of the scope of this research due to time and resource limitations.

Chapter 2

Literature review

2.1 Missing values

The main issue of missing values is that most statistical procedures do not work on incomplete data sets. Handling missing values began to gain attention in 1987 when Little et al. (2002) published the first version of their book *Statistical Analysis with missing values* coupled with an increasing computational power this surged interest in missing values (Graham, 2009). At this time the systematic development of the first approaches to impute missing values started. For example, structural equation modelling software (Allison, 1987; Muthén, Kaplan, Hollis, Planning, & Angeles, 1987) and the first steps in multiple imputation (Tanner & Wong, 1987). A lot of characteristics of modern-day imputation approaches are based on these publications (Graham, 2009). One of those characteristics is the missing values pattern present in a data set, it is important because it affects the bias in your results, and should be accounted for when choosing an imputation approach.

2.1.1 Missing value patterns

Three missing value patterns can be distinguished: Missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) (Donders et al., 2006; Little et al., 2002, pp. 11-19). The first type of missing values, MCAR, occurs when missing values are not associated to any other variable. The probability of y missing is not dependent on either x or y. The missingness has nothing to do the the studied subject. Examples are: surveys lost in the mail, or a damaged blood sample in a medical lab. This is not mathematically random, but the missingness is not related to the data (Bland, 2000, pp. 306-307). This means that all complete samples in a data set remain completely random (Little et al., 2002, pp. 11-19).

The second type of missing values, MAR, has a confusing name. The data is not missing at random, but missing conditionally. The missing values are not related to unobserved values, but are related to the observed data (Graham, 2009). The probability of data y missing is dependent on x. For example, a child does not show up for his doctors appointment, there is no data about the current sickness of the child. However, data from previous appointments indicates that the child is in bad health, and a doctor might conclude that the child is sick, therefore did not show up (Bland, 2000, pp. 306-307). A more geographical example comprises the measurements of wind speeds. During a storm some weather stations along the coastline do not report any data. These values are missing, but the meteorological institute knows that at winds speeds higher than 150 kilometres per hour the weather stations shut down. These speeds are not uncommon along the coastline during a storm, thus, the institute fills in the values to be higher than 150 kilometres per hour. The The MAR pattern is not verifiable without the actual missing values but it can be recognized by someone with expert domain knowledge.

The third type of missing values, MNAR, occurs when the missing values are associated with unobserved variables. For example, people with a low income feel uncomfortable sharing this information in a survey, resulting in missing values for many respondents that fall into lower income categories. A geographical example comprises the measurement a neighborhood survey, people living in social housing estates feel less comfortable sharing this information, resulting in missing values for many neighbourhoods that contain a lot of social housing estates. This creates a geographical pattern that the researcher is not aware off. The MNAR missing value pattern is not verifiable without knowing the actual values. MNAR does affect statistical procedures and is known as the non-ignorable pattern. Since the data is affected by values the researcher has not measured, any imputation method will result in skewed predictions. Therefore, this is the worst case scenario (Bland, 2000, pp. 306-307).

In the first two patterns it is safe to remove entries with missing values and impute them accordingly. The statistical power is reduced, but as as long as the total amount of original entries is still large, the estimates remain close to full population values. However, the third pattern can cause bias in a statistical model. If a population survey has a lot of missing values in the lower income group, researchers could conclude that the population is richer, which is not a correct representation of the population. Therefore, imputation should be handled with care (Donders et al., 2006; Muthén et al., 1987; Little et al., 2002, pp. 11-19) Furthermore, it is hard to define the missing value pattern without obtaining information about the missing values. There are some tests to check for MCAR or MAR, but those are not widely used and their accuracy is disputed (Graham, 2009). To verify if a missing value pattern is MAR or MNAR is not possible because it requires the actual values to be known (Graham, 2009; Muthén et al., 1987). Although these mechanism are presented as mutually exclusive patterns, they should not be seen as such. In reality the assumptions of these patterns are most often untenable, and researchers should focus on the extent of which a pattern is present, and to what extent this influences their imputation approach (Graham, 2009, p. 567).

2.1.2 Classic missing value approaches

Assumptions and characteristics of classic non spatial imputation approaches are widely used in state art machine learning and spatial imputation models Graham (2009) states that three algorithms had a significant influence to development of state of the art approaches: the Expectation-maximization (EM) algorithm, multiple imputation (MI) and full information maximum likehood (FIML) methods. Both EM and MI iterate their process on small batches until the variance in their results is below a threshold based on their probability distribution (Tanner & Wong, 1987). FIML is a maximum likehood approach that processes the complete data set at once and predicts missing values on their maximum likehood. Many machine learning algorithms incorporate one or more of these methods to create more accurate predictions than traditional missing value imputation approaches (Collins, Schafer, & Kam, 2001). These traditional statistical imputation approaches perform quite well on all kinds of data (Graham, 2009), however, state of the art machine learning models perform significantly better on missing value imputation and outperform imputation on MNAR data by a large margin (Jerez et al., 2010; Garciarena & Santana, 2017).

2.2 Machine learning

Machine learning is the study of algorithms that computers use to progressively improve their performance on a specific task without being explicitly programmed to complete this task (Murphy, 2012a, p.1). The term machine learning was first used in 1959 by Arthur Samuel (Koza, Bennett, Andre, & Keane, 1996), since then a lot of algorithms have been developed. In recent years the scientific and commercial interest in machine learning has increased enormously (Jordan & Mitchell, 2015). Nowadays many day-to-day task and scientific methods encompass machine learning (Hastie et al., 2009, p. 9). For example:

- Image analysis: face detection on a mobile phone, and automatic building recognition from satelite imagery.
- Text analysis: filtering spam emails and customer support chat-bots.
- Data mining, finding disease patterns in medical data.
- Predicting future events such as weather patterns.

Typically machine learning is divided in two types: supervised and unsupervised models (Hastie et al., 2009, pp. 9, 485). Supervised learning approaches aim to predict the missing value y based on variable(s) x or $y = f(x)$. The supervised learning process works in two consecutive phases: training and prediction. In the training phase an algorithm processes a set of measurements for which both y and x are known and by minimizing the error margin the model adapts. In the prediction phase the trained model processes a set of measurements for which only x is known and predicts the corresponding y value. Supervised learning is further split into regression and classification. Regression approaches predict a continuous variable (e.g., length in meters), classification approaches predict a category (e.g., spam email or not spam email) (Murphy, 2012a, pp. 1-5).

Unsupervised learning approaches only process a set of input measurements x and no corresponding output values. The purpose of unsupervised approaches is to model the underlying patterns in the input measurements. Unsupervised learning is further split into clustering and association. Clustering groups the data in an n amount of groups that have similar values (e.g., people with similar online shopping behaviour), association discovers links between large portions of the input data (e.g., people of type x tend to buy product y) (Hastie et al., 2009, p.p. 9, 485).

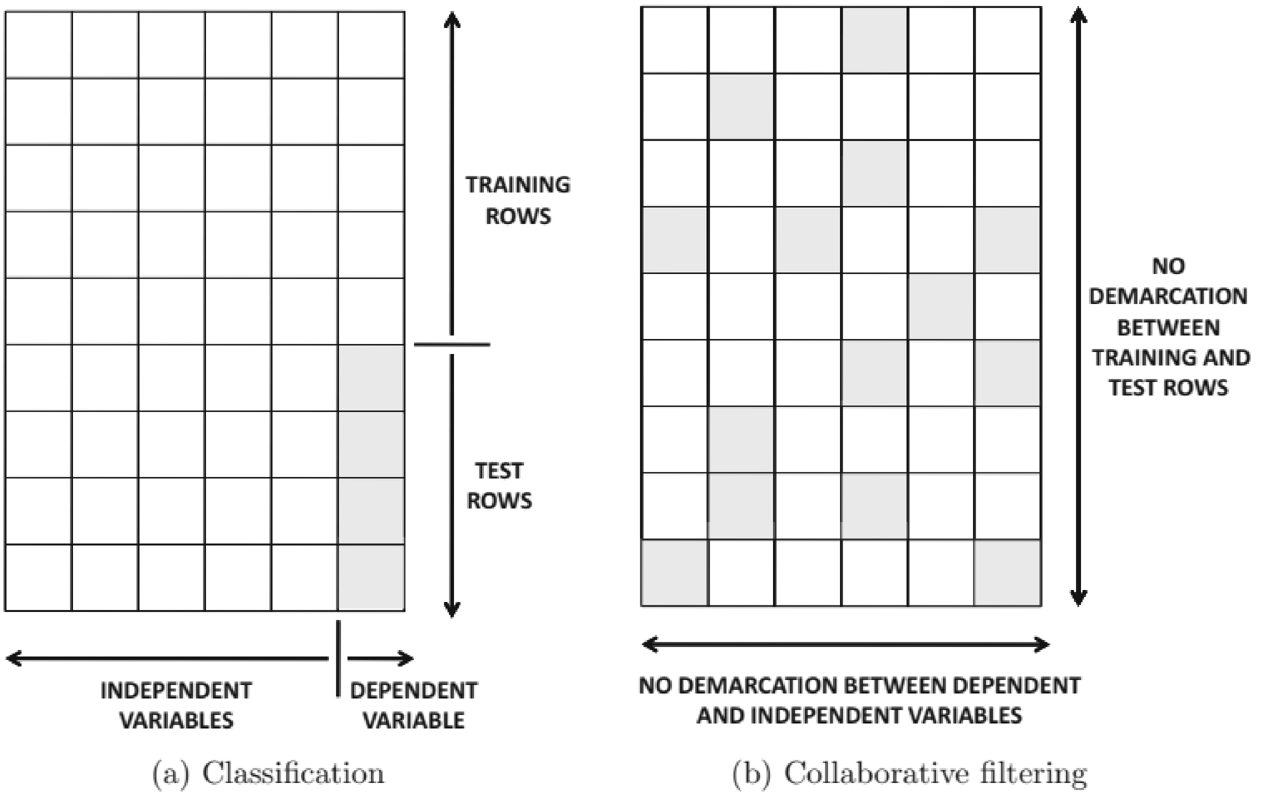
2.2.1 Missing value prediction using machine learning

The machine learning domain comprises an increasing amount of algorithms. A major part of these algorithms has never been used for missing value imputation (Luengo, García, & Herrera, 2012). This introduces a problem because machine learning models assume that there is an $m \times n$ matrix in which all but the last columns are independent variables. The last column is the dependent variable with an n amount of missing values, whereas missing values often occur in multiple columns. This means that there is no demarcation between variables as illustrated in figure 2.1.

Collaborative filtering comprises the imputation of missing user information in recommender systems using unsupervised association (e.g., promoting films to people with similar movies preferences). Within the domain of collaborative filtering different machine learning models are rewritten to predict users association The collaborative filtering scientific domain has a lot of similarities with missing value imputation and extensive research has been done to implement machine learning models in which there is no clear demarcation between dependent and independent variables (Marlin, 2004; Aggarwal, 2016, pp. 71-74). Aggarwal (2016, p. 72) summarizes the crucial difference between regular machine learning and missing value imputation:

- There is no clear separation between dependent and independent variables.
- Subsequently, there is no clear separation between test and training data.
- Traditionally columns represent features and rows data instances. However, there is not a single column to be predicted, thus this can be transposed.

To address these differences machine learning models for missing value imputation are approached in two groups: Neighborhood-based algorithms and learning based algorithms. Aggarwal (2016, pp. 29-45) states that


TRAINING ROWS
TEST ROWS
INDEPENDENT VARIABLES
DEPENDENT VARIABLE
NO DEMARCATION BETWEEN TRAINING AND TEST ROWS
NO DEMARCATION BETWEEN DEPENDENT AND INDEPENDENT VARIABLES

(a) Classification

(b) Collaborative filtering

Figure 2.1: (a) Traditional machine learning, (b) missing value imputation and collaborative filtering. Grey squares indicate missing values. From Aggarwal (2016, p. 72)

neighborhood-based algorithms are based on the assumption that rows displays similar patterns, thus missing values are predicted by using values from similar rows. The earliest neighborhood based algorithms predicted values using measures such as cosine similarity and Pearson correlation. Neighborhood-based algorithms are easy to understand, and relatively easy to debug. The prediction process is interpretable, so individual predictions are understandable. Besides, adapting these algorithms to handle missing values with no demarcation between dependent and independent variables is only a small preprocessing step (Luengo et al., 2012). However, neighborhood-based algorithms are impractical in large-scale settings, because assuming you have a $m \times n$ matrix, the corresponding complexity is $O(m^2 \cdot n')$, adding a single row or column increases complexity quadratic. To make this process adapt to large-scale settings the data can be clustered beforehand, subsequently missing values are predicted based on the center of the cluster with the highest similarity (Aggarwal, 2016, pp. 45-47).

Learning based algorithms create a summarized model of the data up front. Next, missing value predictions are predicted based on this model in the next phase. Model-based algorithms have a number of advantages over neighborhood-based algorithms. The complexity for prediction is smaller, because the learned model does not have to revisit all the present data once the first phase is complete. The initial building time of most model-based algorithms are not quadratic to their input, although the building time does vary between models (Aggarwal, 2016, p.73). However, most model-based algorithms operate as a black box (Murphy, 2012a), intermediate steps in the process are hard or impossible to interpret, predictions of missing values are harder to explain than in neighborhood-based algorithms. Processing missing values with no demarcation between dependent and independent variables requires either adjustment to the algorithm, or an time intensive iterative process in which the algorithm has to re-run for every column that contains one or more missing values (Luengo et al., 2012; Aggarwal, 2016, pp. 73-75). In practice this means that applying machine learning models to impute missing values is done by creating a prediction function for each column, which uses all the other columns as input.

The majority of machine learning models are not available in out-of-the-box software packages. The next section describes a selection of both neighborhood-based and model-based algorithms that are either available in out-of-the-box software packages or are relatively easy to rebuild. Most important, the selection only comprises those algorithms that have been scientifically studied with regard to missing value imputation.

2.2.2 Neighborhood-based algorithms

The k-Nearest Neighbour algorithm is one of the earliest machine learning algorithms, it has the ability to predict both discrete and continuous values. The k-Nearest Neighbour algorithm calculates the distance between each entry in a matrix and stores these in a new matrix. Each missing value is predicted by an average based on the k-number of nearest neighbours in the distance matrix (Hastie et al., 2009, pp. 463- 474). An advantage of the k-Nearest neighbour is the easy integration of new columns, and adjustment of the distance metric. It requires only minor adjustment to process data with no demarcation between dependent and independent variables. One of the main objectives of this algorithm is analyzing large databases. However, for each iteration the algorithm searches through the complete data set, which makes it less suitable for large dimensional data sets (Batista & Monard, 2002).

2.2.3 Model-based algorithms

Support Vector Machine (SVM) is a popular machine learning algorithm for classification and regression. SVM is a linear classifier that transforms its inputs to a multidimensional feature space, thus allowing it to make non-linear decision in a two-dimensional feature space (Chang, Lin, & Tieleman, 2008) The main advantages of SVM are its applicability in high dimension data sets, even if the number of columns (features) is larger the the number of rows. The algorithm is memory effective because it uses a limited amount of samples to train the model, and just like the k-Nearest Neighbour algorithm adding new columns and re-calculating the distance metric is relatively simple. SVM often has high accuracy in missing values imputation, but is susceptible to overfitting (Zhang & Iyengar, 2002). An overfitted model corresponds to closely to its training data, which limits the models capabilities to make predictions for new data (Leinweber, 2007).

Naive Bayes is a group of algorithms that all assume that a single feature is independent from any other feature. For example, the features size and color are not assumed to be associated, although this will often be correct, it will be false in many instances. Although this assumption is rarely true in real life, the oversimplified design and assumptions of Bayes classifiers have been theoretically grounded (Kuncheva, 2004). The algorithm produces highly accurate missing value imputations, even on very large databases (Su, Khoshgoftaar, Zhu, & Greiner, 2008) Most naive Bayes algorithms use a maximum likehood approach to make predictions and are able to support those productions with a probability.

More recently developed algorithms such as random forests decision trees often outperform Bayes classifiers but require a larger set of training samples to make accurate predictions (Su et al., 2008; Caruana & Niculescu-Mizil, 2006). Decision and regression trees are algorithms that split the independent variables in a hierarchical matter (Aggarwal, 2016, pp. 74-77). A simplified version of a decision tree to predict the dependent variable gender based on independent variables weight and height is visualized in figure 2.2. Assuming there is a person of which the gender is missing, but their height and weight are respectively 175 centimeters and 85 kilograms, according to the model in figure 2.2 the missing value would be predicted as *Male*. Scaling affects decision trees by a relatively small margin whereas they are able to exclude irrelevant data during the training phase. However, decision trees accuracy is low, and the models are prone to overfitting (Hastie et al., 2009, pp. 305-317). To combat these problems the random forest decision tree algorithm has the ability to increase accuracy and prevent overfitting by introducing a multitude of decision trees during the training phase. To predicts a class or continuous variable the algorithm takes the prediction from the modal decision tree created in the training phase. The accuracy, training time and applicability in multiple domains is relatively high for an easy-to-set-up algorithm in both traditional machine learning and missing value imputation (Aggarwal, 2016; Hastie et al., 2009, pp. 305-317)

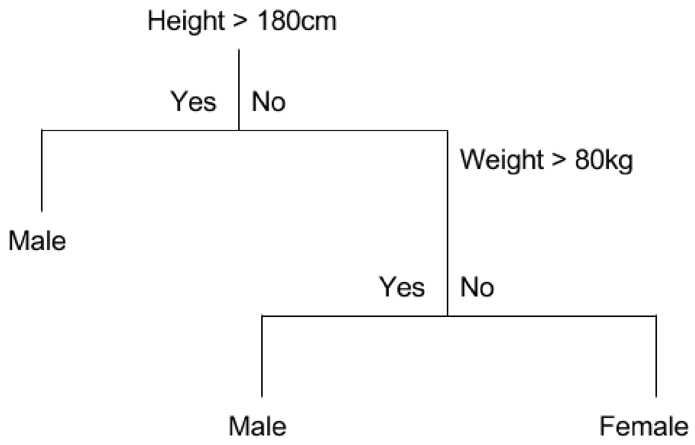


Figure 2.2: A decision tree to determine gender based on height and weight (Machine Learning Mastery, 2017)

Gradient boosting algorithms predict a class or continuous variable by combining multiple algorithms. Combining multiple algorithms is a technique many machine learning models use. Unlike many models that combine algorithms, gradient boosting does not take the average, median, or modal prediction, instead it uses consecutive models to correct the error of the previous model (Aggarwal, 2016, pp. 74-77). For example, after applying the initial model on a data set with missing values and inspecting the residuals, the residuals of the first model are repeatedly 5 higher than the actual values, the second model improves by adding 5 to each predicted missing value. The residuals of the second model are repeatedly 2 higher for all males in the data, but not for females. The third model improves by adding 2 to each predicted missing value that is a male. The result is a robust model that has high accuracy and is not prone to overfitting for both traditional machine learning domains and missing value imputation (Su et al., 2008; Hastie et al., 2009, pp. 359 - 361).

2.3 Spatio-temporal interpolation

Spatial interpolation predicts values neighbouring in space from a limited numbers of sample data points, polygons or raster cells. Whereas traditional imputation methods and machine learning algorithms predict based on statistical similarity of features, spatial interpolation predicts missing values based on their geographical closeness (De Smith, Goodchild, & Longley, 2018, p. 387, 400). Spatial interpolation in its simplest form predicts a value by copying the nearest value in space. Spatio-temporal interpolation predicts values on both geographical distance and temporal distance. A large number of spatial interpolation algorithms achieve highly accurate prediction in both space and time, by treating temporal distance in the same manner as geographical distance (Gerber, De Jong, Schaepman, Schaepman-Strub, & Furrer, 2018; Li & Revesz, 2004). The following sections will describe both spatial and spatio-temporal algorithms that are available in out-of-the-box packages or are easy to rebuild and are scientifically grounded in spatio-temporal missing value interpolation.

2.3.1 Inverse distance weighted interpolation

Inverse distance weighted (IDW) is one of the simplest and most often applied spatial interpolation methods (De Smith et al., 2018, pp. 388-391). IDW interpolation assumes that geographically close measurements are more alike than those further apart. IDW takes each measurement into account to predict missing values, but gives a higher weight to measurements located closeler in space (De Smith et al., 2018, pp. 388-391). The general prediction formula for IDW is as follows:

$$\widehat{Z}(s_0) = \sum_{i=1}^{N} \lambda_i Z(s_i)$$

Where s_0 is the prediction location, N is the number of measured (non-missing) values, λ_i is the weight for the measured value ith location decreasing by distance, and $Z(s_i)$ is the actual value at the ith location (ESRI, 2009, p. 114). Weights are determined by a function of distance:

$$\lambda_i = d_{i0}^{-p} / \sum_{i=1}^{N} d_{i0}^{-p}$$

Where d_{i0} is the distance between prediction location (s_o) and the measured locations (s_i). The parameter p influences the weights of the measured locations, if $p = 0$, there is no decrease in weight over distance. As p increases, the weights decrease faster per distance unit. In many modelling scenarios p is set to $p = 1$ or $p = 2$ (ESRI, 2016b). The total sum of weights is scaled to equal 1:

$$\sum_{i=1}^{N} \lambda = 1$$

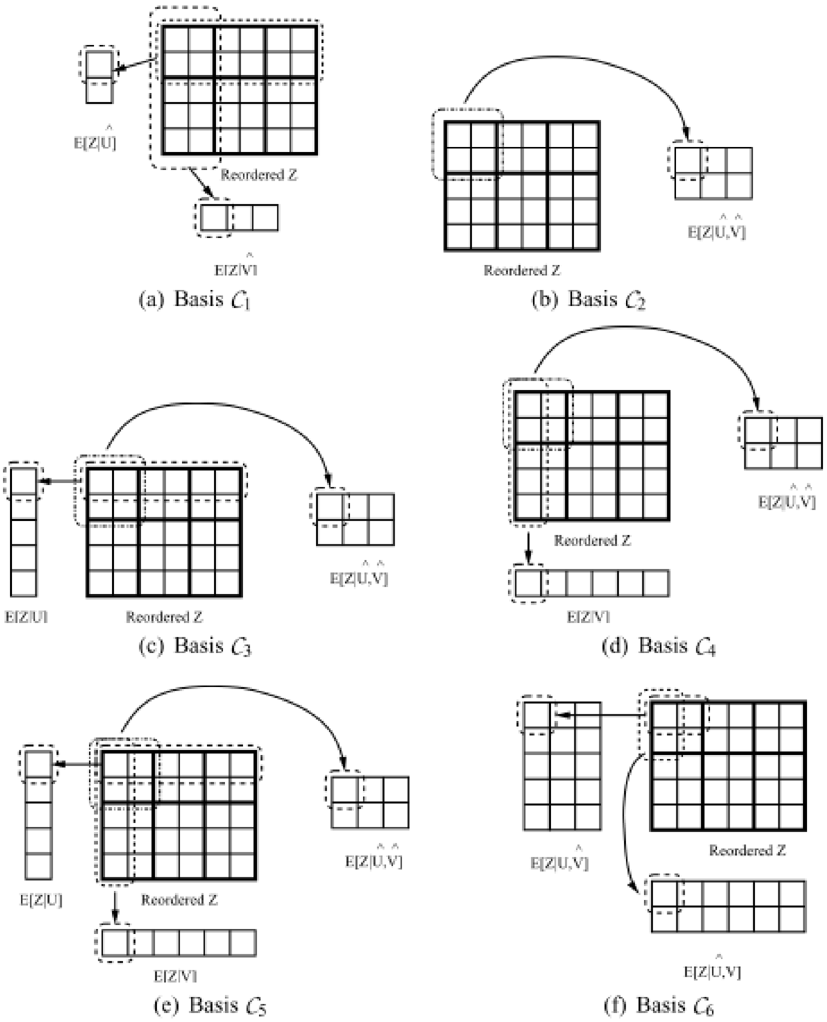
IDW achieves higher accuracy than non-spatial imputation methods on geospatial data Mueller et al., 2004; Li and Revesz, 2004. Spatio-temporal IDW increases computational complexity linearly per timestamp compared to non temporal spatial data(Li & Revesz, 2004; Gerber et al., 2018). IDW is available and comprehensive documented in most GIS software suites and programming packages such as ArcGIS (ESRI, 2016a), QGis(QGis, 2017), rGdal (R Documentation, 2015), and Pysal (PySaL, 2018).

2.3.2 Kriging and co-kriging

Kriging interpolation is based on univariate and multivariate linear regression models. Similar to IDW, Kriging applies the same weighted distance function to predict missing values. However, the weights depend on spatial arrangement of the missing values to the present values. Determining the weights is done by fitting the measured values to a model (e,g., linear, gaussian or spherical models) (Lam, 1983; Oliver & Webster, 1990; De Smith et al., 2018, pp. 414-424). The method predicts robust high accuracy results that outperform IDW (Mueller et al., 2004; Li & Revesz, 2004) and has lower levels of spatial correlation in residuals than IDW (QGis, 2017; De Smith et al., 2018, pp. 414-424). Co-kriging is a form of kriging that analyzes auto-correlation in all columns of a data set allowing the analysis of multiple timestamps per location. Co-kriging often results in more accurate results than regular kriging in spatio-temporal missing value imputation, but is computationally more complex (Aalto, Pirinen, Heikkinen, & Venäläinen, 2013). Kriging and co-kriging are available in most GIS software suites, although the number of available models to create weights differ. ArcGIS currently has the highest quantity of models and parameter adjustment available (ESRI, 2016b).

2.4 Bregman Block Average Clustering with I-divergence

Co-clustering as a dimensionality reduction method has been explored before the BBAC-I algorithm was introduced (Oliveira & Madeira, 2004; Hartigan, 1972). Banerjee et al. (2007) published the first version of the BBAC-I algorithm in 2004 (Banerjee, Dhillon, Ghosh, Merugu, & Modha, 2004), and published a more comprehensive article in 2007 (Banerjee et al., 2007). Apart from the BBAC algorithm the 2007 article introduces six schemes to compute co-clusters from a matrix. The article explores the use of these schemes in several domains, such as dimensionality reduction, text clustering, natural language processing and missing value imputation. Figure 2.4 displays the schemes, each scheme uses a different combination of summary statistics to reconstruct the matrix. For example, scheme two that was previously used for missing value imputation uses the average of the co-cluster for reconstruction, while scheme five uses the co-cluster average, row and column averages, and row and column cluster averages. Besides the schemes, the authors present two different measures to calculate statistical distance in their algorithm: euclidean distance, and Bregman I-divergence. Banerjee et al. (2007) explored the missing value potential of their algorithm using the scheme two and five with I-Divergence. Ten years later the BBAC-I algorithm with I-divergence was first explored for concurrently analyzing timestamps and locations in GTSs (Wu et al., 2015; Wu, 2016; Wu et al., 2016).



(a) Basis C_1 (b) Basis C_2

(c) Basis C_3 (d) Basis C_4

(e) Basis C_5 (f) Basis C_6

Figure 2.3: *"Schematic diagram of the six co-clustering bases. In each case, the summary statistics used for reconstruction (e.g.,* $E[Z|\hat{H}]$ *and* $E[Z|\hat{V}]$*) are expectations taken over the corresponding dotted regions (e.g., over all the columns and all the rows in the row cluster determined by* $\hat{U}$ *in case of* $E[Z|\hat{U}]$*)." Banerjee, Dhillon, Ghosh, Merugu, and Modha (2007, p. 1938)*

2.4.1 Advantages of co-clustering over one-way clustering

Clustering is a data mining technique used widely in many domains, including spatio-temporal research. Regular or One-way clustering search for rows or columns that contain similar values and subsequently groups similar rows or columns. This process reduces the amount of data that has to be processed to obtain information, this reduction is called dimensionality reduction. Clustering facilitates the exploration of large data sets, especially when the information about a data set is limited (De Smith et al., 2018, pp. 183-154). One-way clustering is limited by its ability to analyze either rows or columns. Whereas co-clustering seeks blocks of rows and columns that are inter-related, and thus results in information about both rows and columns (Banerjee et al., 2007). Banerjee et al. (2007) state three reason why co-clustering is desirably over one-way clustering. Firstly, row column blocks give more easy and digestible information, while preserving more information about the originial data than one-way clustering (Banerjee et al., 2007, p. 1920).

Secondly, Co-Clustering can be seen as a more efficient dimensionality reduction than one-way clustering. Instead of analyzing all values in a matrix, only the clusters centroid, mean, or other statistical measure is analyzed. This leads to a compact representation of the original data. Co-clustering leads to simultaneous clustering along rows and columns, reducing dimensionality in both axes, whereas one-way clustering only reduces the dimensionality on one axis (Banerjee et al., 2007, pp. 1920-1921).

Thirdly, the processing time of Co-clustering algorithms are less than one-way clustering algorithms. The computational time of one-way clustering algorithms is $O(mnk)$ per iteration where m is the number of rows, n is the number of columns, and k is the number of row clusters. Co-clustering complexity can be approximated by $O(mkl + nkl)$, where all the parameters are the same as for one-way clustering and l is the number of column clusters. In most applications the number of row and column clusters is much smaller than the number of rows and columns, which leads to a reduction in computational time for co-clustering (Banerjee et al., 2007, p. 1921).

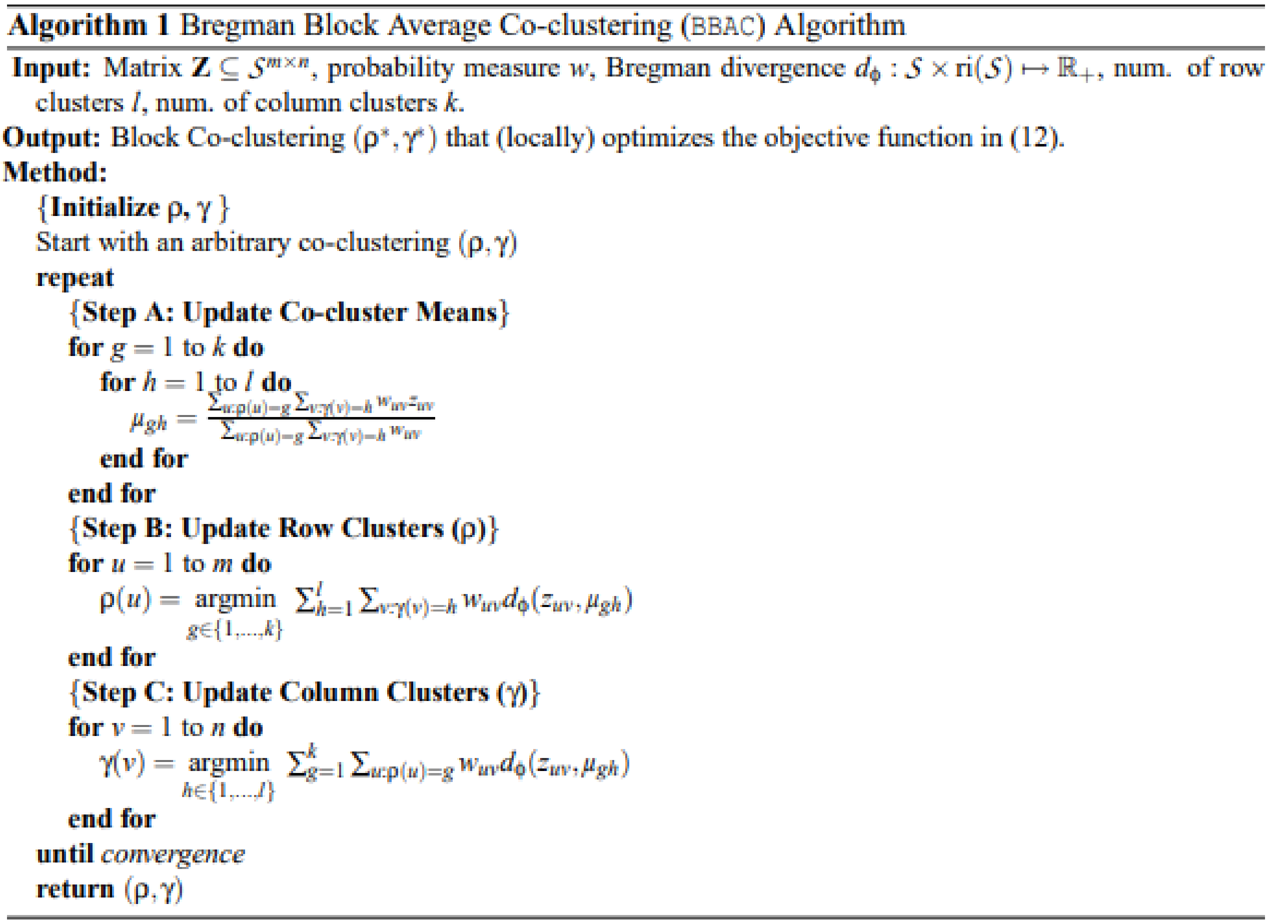
2.4.2 BBAC-I Algorithm

The BBAC algorithm starts with an random choice of co-clustering row clustering (p) and column clustering (y). At every iteration of the algorithm either p or y is updated to decrease the loss function. These iterations are repeated till convergence. The pseudo code presented by (Banerjee et al., 2007, p. 1934) is displayed in figure 2.4 and lists the inputs, outputs, and steps of the algorithm.

The required inputs are a matrix Z that contains data in the form of real numbers, the measure matrix w, a Bregman Divergence (e.g., Squared euclidean distance, I-Divergence), the number of row clusters l, and the number of column clusters k. The algorithm starts with random ordering of co-clusters p, y. Then the row and column clusters are updated by calculating their averages, calculating the distances between the updated clusters using the selected Bregman Divergence respectively. Then the current error is calculated. If the current error is larger than the convergence, a new iteration commences. If the current error is equal or smaller than the convergence the algorithm returns the co-clustering p, y.

Banerjee et al. (2007) state that Euclidean squared distance and I-divergence are the most suitable distance measures for their algorithm, with a slight preference for I-divergence. Subsequent papers reviewing or applying the BBAC algorithm confirm this and recommend to use I-Divergence, as it provides a stable and more accurate similarity measure than euclidean distance (Kwon & Cho, 2010b; Wu, Jin, & Hoi, 2009). The formula to calculate the I-divergence distance between matrix elements z_1 and z_2 in the matrix Z is:

$$z_1 \log(z_1/z_2) - (z_1 - z_2)$$



Algorithm 1 Bregman Block Average Co-clustering (BBAC) Algorithm

```
Input: Matrix $\mathbf{Z} \subseteq \mathcal{S}^{m \times n}$, probability measure $w$, Bregman divergence $d_\phi : \mathcal{S} \times \mathrm{ri}(\mathcal{S}) \mapsto \mathbb{R}_+$, num. of row
  clusters $l$, num. of column clusters $k$.
Output: Block Co-clustering $(\rho^*, \gamma^*)$ that (locally) optimizes the objective function in (12).
Method:
  {Initialize $\rho$, $\gamma$ }
  Start with an arbitrary co-clustering $(\rho, \gamma)$
  repeat
    {Step A: Update Co-cluster Means}
    for $g = 1$ to $k$ do
      for $h = 1$ to $l$ do
        $\mu_{gh} = \frac{\sum_{u:\rho(u)=g} \sum_{v:\gamma(v)=h} w_{uv} z_{uv}}{\sum_{u:\rho(u)=g} \sum_{v:\gamma(v)=h} w_{uv}}$
      end for
    end for
    {Step B: Update Row Clusters ($\rho$)}
    for $u = 1$ to $m$ do
      $\rho(u) = \underset{g \in \{1,\ldots,k\}}{\mathrm{argmin}} \sum_{h=1}^{l} \sum_{v:\gamma(v)=h} w_{uv} d_\phi(z_{uv}, \mu_{gh})$
    end for
    {Step C: Update Column Clusters ($\gamma$)}
    for $v = 1$ to $n$ do
      $\gamma(v) = \underset{h \in \{1,\ldots,l\}}{\mathrm{argmin}} \sum_{g=1}^{k} \sum_{u:\rho(u)=g} w_{uv} d_\phi(z_{uv}, \mu_{gh})$
    end for
  until convergence
  return $(\rho, \gamma)$
```

Figure 2.4: Bregman Block Average Co-clustering pseudo code Banerjee, Dhillon, Ghosh, Merugu, and Modha (2007, p. 1934)

This formula is applied to calculate the distance between all elements (z_{xyz}) in matrix Z which results in a distance, or co-occurrence matrix that is subsequently used as input to re-order the co-clusters in the BBAC algorithm.

2.4.3 Imputing missing values with BBAC-I

Banerjee et al. (2007, pp. 1959-1960) briefly explained how the BBAC algorithm was implemented as a missing value imputation algorithm. George (2005) reviewed the missing value imputation technique more thoroughly and wrote a more comprehensive documentation. George (2005) implemented an early version of the BBAC algorithm with a euclidean distance measure, instead of one of the Bregman divergences. Although this does influence the creation of clusters, it does not affect the missing value imputation method itself. The BBAC missing value method consist of three steps. The first steps requires co-clustering the data, with the input data matrix and a measure matrix of ones and zeros for present and missing values respectively. This ensures that only the known values contribute to the loss function in the co-clustering algorithm.

The second step calculates the averages of the previously calculated co-clusters. This step results in three output matrices: row cluster averages, column cluster averages and co-cluster averages.

The third step is predicting the missing values by reconstructing the matrix based on cluster statistics. This step differs between schemes. Scheme two is not incorporated in the method from George (2005), and requires the co-clustering average. The corresponding prediction formula to reconstruct the matrix is:

$$\hat{z}_{ij} = Z_{ij}^{COC}$$

Where $\hat{z}_{ij}$ is the missing value, and Z_{ij}^{COC} is the average of the corresponding co-cluster. The formula presented by George (2005) uses scheme five to reconstruct the matrix and predict a single missing value is:

$$\hat{z}_{ij} = Z_{ij}^{COC} + (Z_i^R - Z_i^{RC}) + (Z_j^C - Z_j^{CC})$$

Where Z_i^R and Z_j^C are the average row and column mean, and Z_{ij}^{COC}, Z_i^{RC}, and Z_j^{CC} are the average values of the corresponding co-cluster, row-cluster and column-cluster respectively (George, 2005, figure 3.1). This formula is the best additive approximation as a co-clustering problem given the available summary statistics (George, 2005, p. 3). A more practical advantage concerns the implementation of the prediction. The prediction for both schemes can be calculated from inputs and outputs from the BBAC-I algorithm without re-building the whole algorithm.

Chapter 3

Materials and Methods

3.1 Study design and procedure

This chapter contains a model comparison study to explore the missing value imputation of geographical data using the BBAC-I algorithm. The study compares several synthethic case studies: data is removed from complete data sets with varying levels of temporal and spatial autocorrelation and those removed values are predicted. The residuals of the BBAC predictions will be compared to the residuals from machine learning and spatial interpolation models.

The study procedure consists of four steps: data pre-processing, missing value imputation per model, residual calculation, and analysis. The procedure is displayed in figure 3.1. All the steps will be executed in Python and both code and data will be stored in a public GitHub repository to make this research reproducible.

The purpose of the pre-processing step is to create four data sets that are ready to use as input for the selected models. Values will be removed MCAR from the data sets to produce missing values. In total 12 data sets will be created with 10, 20, and 30 percent of missing values respectively per data set. Using varying amounts of missing values is standard procedure in collaborative filtering and missing value imputation exploration (Donders et al., 2006; Marlin, 2004; Graham, 2009), introducing more than 30 percent missing values creates a lot of random noise in the results and is discouraged (Graham, 2009).

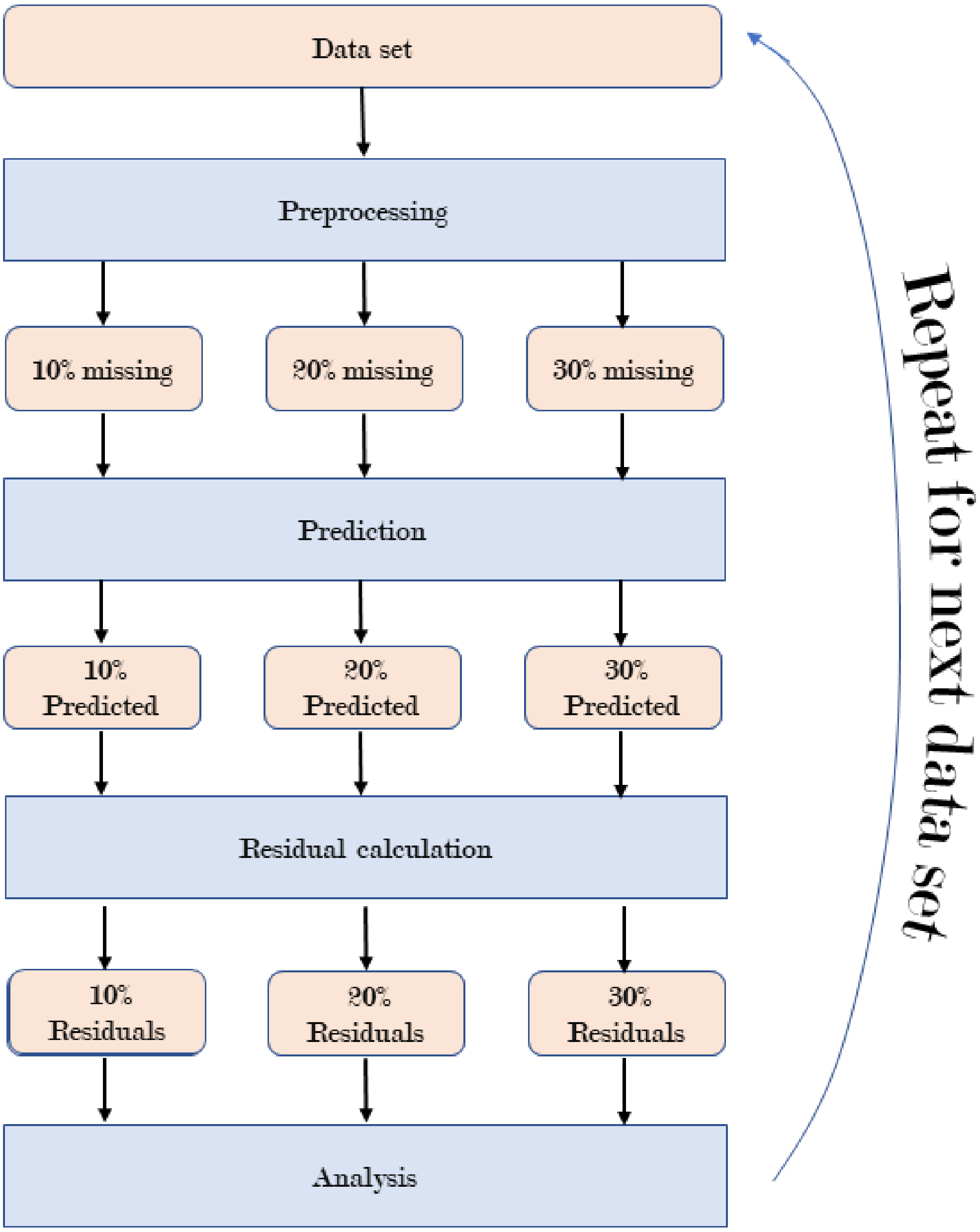


Figure 3.1: Flowchart of the study procedure

The purpose of the missing value imputation is to create an output matrix containing predicted values per model in a new matrix. Each model receives the six data sets as input and after imputation the missing value each output will be processed to a $m \times n$ matrix with locations as rows, and timestamps as columns. Each output will have the exact same order of rows and columns. The analysis process of each model is described in subsection 3.4. The prediction process for each data set and model is timed using the Time package in Python (Python Documentation, 2018).

The purpose of the residual calculation is to calculate the residual for each row-column combination. The output is a $m \times n$ matrix containing the difference between the actual values and the predicted values in the same structured matrix as the imputation output.

The purpose of the analysis step is to calculate the MAE and RMSE and compare the difference per model and missing value size. The accuracy measurements are described in section 3.5, for each residual output matrix in the same structured matrix as the imputation output. In total each model will have twelve accuracy matrices, three per datasets with 10, 20, and 30 percent of missing values respectively. Subsequently, the global Moran's-I is calculated to analyze the spatial properties of each algorithm. The analysis will be completed by comparing the accuracy measures and spatial auto correlation for each models. Lastly, the run times are compared between models and varying sizes.

3.2 Data

Exploring spatio-temporal missing value imputation requires data sets with a varying levels of spatial auto-correlation and temporal correlation. Spatio-temporal missing value studies often use a multitude of data sets covering different topics (Garciarena & Santana, 2017; Kornelsen & Coulibaly, 2014; Graham, 2009). Therefore, this study will explore the imputation value of BBAC-I on four data sets that cover a different expertise domain with distinct characteristics.

The Snow Data Assimilation System (SNODAS) contains the snow pack properties from the United States National Weather Service's National Operational Hydrologic Remote Sensing Center (NOHRSC). As part of SNODAS, NOHRSC publishes daily snow depth data for the full geographical coverage of the Continental United States. The data set has a resolution of 1 × 1 kilometers and contains the snow depth in millimeters per grid cell (National Weather Service's National Operational Hydrologic Remote Sensing Center, 2019). Processing the whole coverage of the United States of America (USA) for all timestamps, models, and missing value sizes would require a lot of computational time. Therefore the state of Washington has been selected as geographical extent for this study, which surface size is roughly one fiftieth of the USA's surface size. The temporal extent covers the whole year of 2018, there will be 365 timestamps. Washington has has both mountain ranges and a coastline, resulting in different precipitation and temperatures across the state plane (USA Today, 2018).

The second data set contains daily maximum temperatures collected at 23 weather stations in the Netherlands from 1 January 1992 to 31 December 2016. This is available for 23 weather stations for the full time coverage. It is available for free from the Royal Netherlands Meteorological Institute (KNMI), and the coordinates of each station are retrieved in a separate file from the same website. (KNMI, 2019).

The third data sets contains the first leaf dates over the state Ohio, on a 4000 meter resolution. The data set contains all the years from 1980 till 2016, indicating 37 timesetps. For each cell the first leaf date is given as an integer, an integer of 100 indicates the 100th day as the first leaf day. To process the full area of the US for all models and percentages requires a lot of processing time. Therefore a subset of the state of Ohio was made. This state is characterized by high auto-correlation for both spatial and temporal patterns, making it a suitable subset to compare the spatial and temporal properties of each model.

The fourth data set contains the yearly deaths from cardiovascular diseases per municipality for the Netherlands. Since 1996 the data has been collected by Statistics Netherlands (CBS) , 2017 and 2018 have yet to be published, the study will use all the years from 1996 till 2016 (Statistics Netherlands, 2018).

3.3 Exploratory Data Analysis

Exploratory data analysis (EDA) is a method that focuses on distinguishing characteristics of a data set. Spatial exploratory data analysis (ESDA) is a subset of this method that describes spatial distributions (Anseling, 1995). Normally, EDA is used to test the assumptions for errors in statistical interference, and in missing value imputation it is used to select the proper model and corresponding parameters. Hastie et al. (2009) state that for a model comparison study the EDA has another purpose: to determine what underlying structures in the data will effect the imputation process. All four data sets will be subject to ESDA in this study and the results are discussed in the first section of the results chapter. The objective of this ESDA is to find the patterns that might influence imputation and give a general overview of the data. The most important structure examined, is the spatial distribution of measurements, it is theorized that BBAC-I might be able to capture patterns in data with a strong positive spatial auto-correlation in its input measurements. While it is more certain to say that a spatial model will be able to capture such as pattern as opposed to a machine learning model. Each data set will have distinct results from the ESDA, these results function as a baseline for the discussion in the final chapter and can be used to find causes of surprising results. In the ESDA the following topics are discussed:

- General overview of the data set;
- Moran's I for spatial patterns;
- Temporal correlation for temporal patterns;
- Spatial configuration and Neighbourhood matrix justification.

3.4 Model selection and implementation

3.4.1 Machine Learning

To explore the value of machine learning algorithms is spatio-temporal missing value imputation a selection of models has been made consisting of varying algorithm types such as Neighborhood, linear, bayes, decision trees, and gradient boosting. The complete list of models is:

- k-nearest Neighbour
- Support Vector Machine
- Naive Bayes
- Random Forest
- XGBoost

The models description have been discussed in section 2.2. Machine learning learning implementation requires attention to two processes: Parameter optimization and reducing overfitting. Hastie et al. (2009, p. 141) advice to reduce overfitting by introducing cross-validation. K-Fold Cross-Validation is easy to integrate into most machine learning models, and is available in all major machine learning packages. K-Fold Cross-Validation uses random samples of the available data to fit the model, and different random samples to test it. This reduces the chance of overfitting the model to a specific data set (Hastie et al., 2009). However, applying K-Fold cross validation on temporal data risks ignoring temporal correlation, therefore it is recommended to use day forward-chaining (Varma & Simon, 2006; Tashman, 2000; Bergmeir & Benítez, 2012). After training an initial amount of successive time-steps (e.g., the first three time-steps), day forward-chaining considers each time-step (day) as a test set, and all previous time-steps are seen as training data. The method is displayed in figure 3.2. Day forward-chaining produces many different training and test data sets, in the end average MAE or RMSE off all folds is used as model performance. If the amount of time-steps is high, (Tashman, 2000) recommend combining years consecutive years to limit the amount of training and test data sets.

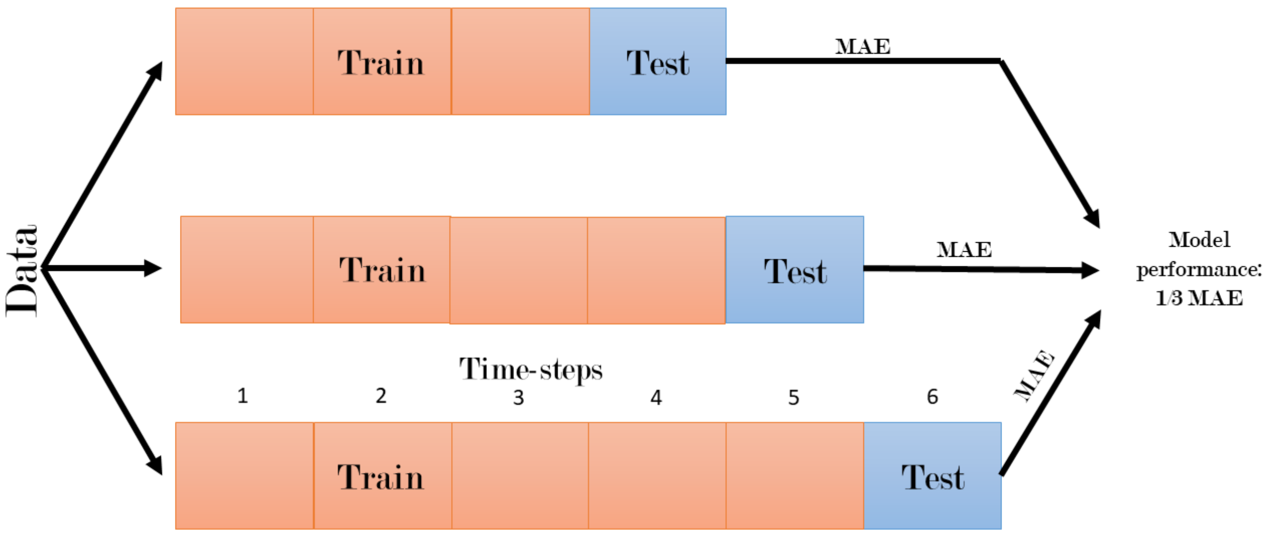


Figure 3.2: Schematic diagram of day forward-chaining

Unlike Cross-Validation, parameter optimization is model specific. For example, the k-Nearest Neighbour algorithm requires a number of k, obviously the number of neighbours that the algorithm uses to predict influences its predictions. Random Forest Trees does not require to set a value for k, but it does requires other parameters such as values for leaf depth and maximum split. Most modern machine learning packages offer tools to determine the optimal parameters based on a small random subset from the whole data set.

One of the major machine learning packages, Scikit-Learn offers all those tools, and is available open source for Python. Furthermore, it has an extensive documentation and is optimized using bindings from the programming language C, which makes it computationally more effective than Python code (Pedregosa, Weiss, & Brucher, 2011). However, it does not offer extensive imputation support. Impyter is a open source addition to Scikit-learn that integrates all its machine learning models as imputation algorithm (Rubin-Schwarz, 2017). Impyter does not rewrite the algorithms, but uses the algorithm to iterate over all the columns that contain missing values as target variables to deal with demarcation between dependent and independent variables. The selection of models will be used by means of Impyter while utilizing the benefits from the Scikit-Learn library. Each training fold consists of 60% training data and 40%, which is a commonly used number in both collaborative filtering and machine learning (Hastie et al., 2009; Ricci, Rokach, Shapira, & Kantor, 2009). If the results are inconsistent, it is advised to try both a smaller and larger training set (Hastie et al., 2009).

3.4.2 Spatial interpolation

In spatio-temporal missing value imputation several packages offer ready-to-use kriging solutions. Arcgis Pro ordinary co-krigging will be used in this study to explore and compare its missing value potential. This suite offers the krigging in digestible manner for both GUI and programming users. Furthermore it allows several input types (e.g., raster, polygon and points).

3.4.3 BBAC-I

For earlier research purposes the R code for the BBAC algorithm including squared euclidean distance and I-Divergence has been published at GitHub (Filipe, 2016). The other models are run, timed, and analyzed in Python. Therefore, the script will be run using the r2py package, that allows users to execute R code in a Python environment (Gautier, n.d.). The clusters are converted from R dataframes to Python arrays and subsequently used to predict missing values in Python based on the matrix reconstruction in section 2.4.3.

3.4.4 Parameter optimization

The cross validation module does offer additional tools for parameter optimization as well. The parameter optimization is based on grid search. Grid search creates a dictionary of parameters and their possible values, subsequently it tests all the parameter combinations and returns the set of parameter values that produces the highest accuracy (Hastie et al., 2009; Machine Learning Mastery, 2017). This process is repeated for each model and data set. The possible values are determined by the researcher, for example the number of k-neighbors are given by the researcher as 5-25 with steps of 5. The results is five different parameter values for k-neighbors: 5, 10, 15, 20, and 25. If another parameter in the model has three different values this would result in 15 (3 times 5) possible combinations in the grid search. This process becomes more computationally intensive if tested on a large data set with a lot of possible parameter values, that is why often a random subset is used. In this study the first two steps in day-forward chaining cross validation are used as subset as recommended by (Bergmeir & Benítez, 2012). The parameters that are implemented in the grid search Cross validation are shown in table 3.1. The choice of parameters incorporated in the grid search is defined by three sources: (1) Scikit-learn documentation, and (2) Hastie et al. (2009) for the machine learning parameters. These parameters are expected to influence the accuracy of a model. The third set of parameters is specific for the BBAC-I models, and are discussed thoroughly in the original BBAC-I work (Banerjee et al., 2007). Moreover, some parameters increase processing time substantially, if the accuracy shows minor changes coupled with substantially processing times the slightly less accurate options is chosen. Parameters that are subject to this choice, are discussed in the results from the grid search. Some parameters are discussed more thoroughly. The gamma parameter of SVM determines how far a value ranges in the training phase, low values meaning a far reach, and high values a low reach. This means that a low value incorporates more points to create the prediction function, creating a more generalized model that requires more processing time. On the other hand a high value uses a more precise function requiring less processing time to fit the model, but is prone to overfitting. The range of the SVM gamma parameter increases one decimal position over four steps, as is commonly advised by (Murphy, 2012b). Furthermore, the Random Forest parameter minimum split samples determines how many data points are required to split a into a new leaf. In most models this only marginally improves accuracy. However, after optimizing it for the accuracy on the CVD data set it produced a higher accuracy. For the other models optimizing the minimum split sample did not lead to improvement in accuracy.

Table 3.1: Parameter optimization possibilities for the grid search procedure

Model	**Parameter 1**	**Parameter 2**	**Parameter 3**
k-neareast Neighbour	Number of neighbors, integer values from 1 - 20 with steps of one, twenty possible values	Weight function in the distribution, all values are equal, or values are weighted by their distance in the matrix. Uniform or distance, two possible values.	
Support vector machine	Kernel method to classify in higher dimnensional spaces, possible values are rbf, linear, and poly.	C range of integers from 1 -10 with steps of 1.	Gamma/bandwidth possible values: 0.0001, 0.001, 0.01, 0.1.
Naive Bayes	Maximum number of iterations, 100-500 steps of 100. A total of five possible values.	Fit_intercept, true or false. Adds an non probalistic intercept to center the data.	
Random Forest	Number of leafs (n_estimators), possible values are: 10, 25, 50, 100, and 200.	Minimum split samples required to split a leaf, and possible values 1,2 and 3.	
XGBoost	Loss function: is, lad, huber, or quantlile	Number of estimators: 1,5 ,10, 25, 50, 100, 200, and 300.	
BBAC scheme 2	Number of row clusters 1 -50 steps of 1.	Number of row clusters 1 - 50 steps of 1.	Distance metric: Only I-divergence is used in this study, since there is no theoretical improvement to using eucledean distance.
BBAC scheme 5	Number of row clusters 1 -50 steps of 1.	Number of row clusters 1 - 50 steps of 1.	Distance metric: Only I-divergence is used in this study, since there is no theoretical improvement to using eucledean distance.

3.5 Accuracy measures

Missing value imputation research often measures it output using either Mean Absolute Error (MAE) or the Root Mean Squared Error (RMSE) (Hastie et al., 2009, p. 24). The MAE measures the average magnitude of residuals in which all individual residuals have equal weight. The RMSE is a quadratic scoring rule that measures the average magnitude of residuals (Chai & Draxler, 2014). Both measures are negatively oriented, lower scores indicate better results. The RMSE squares the error, therefore larger residuals have higher weights. Although many studies have used the RMSE for model comparison, (Willmott & Matsuura, 2005) state that this is often not the most suitable accuracy measure. Models with high inter variation between residuals may be identified as less suitable because the RMSE is used. If the residuals are more homogeneous the RMSE is lower. More homogeneous residuals do not reflect a models prediction capabilities. Moreover, if the sample size is larger the RMSE tend to skew upwards, making models on large sample sizes seem less suitable Willmott, Matsuura, and Robeson (2009). However, the RMSE penalizes large errors, which appends information to model performance. The model comparison consists of several distinct models, therefore this study compares models using both the MAE and RMSE.

Chapter 4

Results and Discussion

4.1 Exploratory Data Analysis

This section presents the exploratory data analysis (EDA) of each data set used in the model comparison study. The goal of the EDA is not to extract full information of the features, but to explore the properties of the features that could affect the imputation as discussed in the section 3.3. For example spatial and temporal auto-correlation, administrative boundaries for vector data, and resolution for raster data.

4.1.1 Cardiovascular deaths in the Netherlands

The 2016 CVD data set contains 390 polygons, each polygon is adjacent to at least one other polygon. The administrative boundaries of many municipalities changed through the years, therefore the Statistics Netherlands can not provide accurate CVD statistics for all years. Since 1996 a lot of municipalities have merged, after a merge of municipalities the data collection continued for the new municipality. This problem could be circumvented by adding the total from both merged municipalities to their new municipality, this has been done for 50 municipalities. Besides merging, 85 municipalities were split into multiple municipalities, which does not allow the simple addition of CVDs. Therefore this study has excluded the split municipalities. In total 305 of 390 municipalities have correct data, those are used for the analysis. The included and excluded municipalities are shown in figure 4.1. The exclusion of municipalities leads to islands in the data, that is, not all polygons have an adjacent neighbor. Spatio-temporal statistics are based on the idea that values are more dependent on measurements closer in space, therefore these statistics require a set of neighbors for each value. All polygons that do not have an adjacent neighbor were assigned the four closest neighbors by euclidean distance.

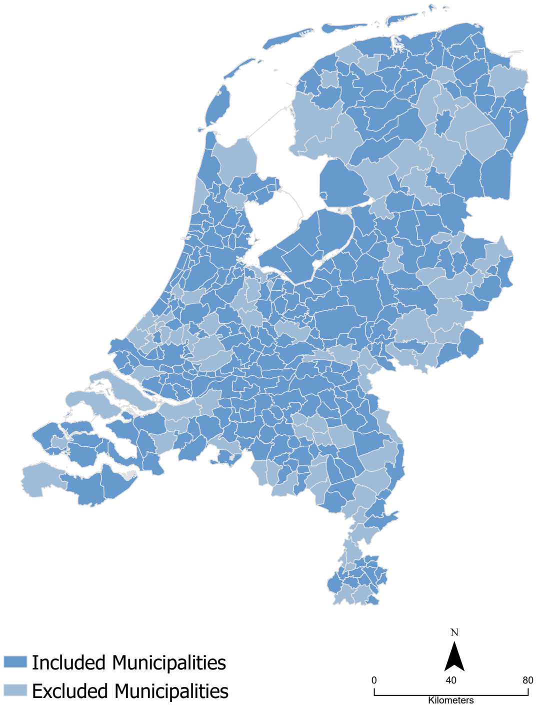


Figure 4.1: Included (n=305) and excluded(n=85) municipalites for the CVD dataset

Figure 4.2 displays the Morans'I for all the years in the data set. None of the years are spatially correlated, and there are no patterns of increasing or decreasing spatial dependence over time. The average Morans' I is 0.038 with a very small standard deviation (0.006), the corresponding p-values indicate that the observed pattern of spatial values could be random.

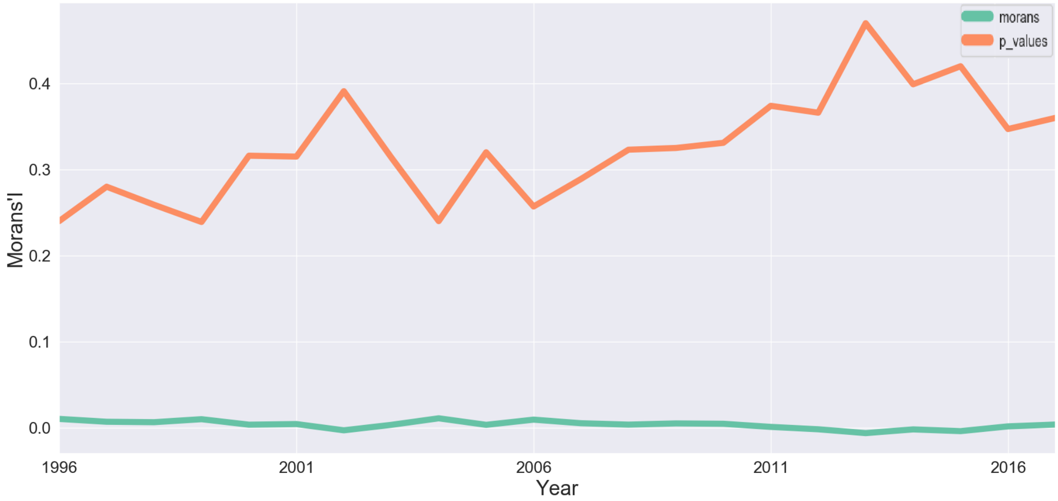


Figure 4.2: Morans' I and corresponding p-values of absolute municipal CVDs between 1996 and 2017 in the Netherlands.

Figure 4.3 displays the temporal auto-correlation for the CVD data set. The temporal auto-correlation is higher with a lower lag, thus years closer have more common properties. However, this effect diminishes after after 3 years of lag and is not significant (temporal correlation = 0.045, p=0.23). To conclude, the data set does not have distinct temporal properties. From the exploratory data analysis it can be concluded that the CVD data set is not characterized by spatial or temporal auto correlation. Therefore, it is hypothesized that non spatial machine learning methods will have better or comparable results to spatio-temporal methods.

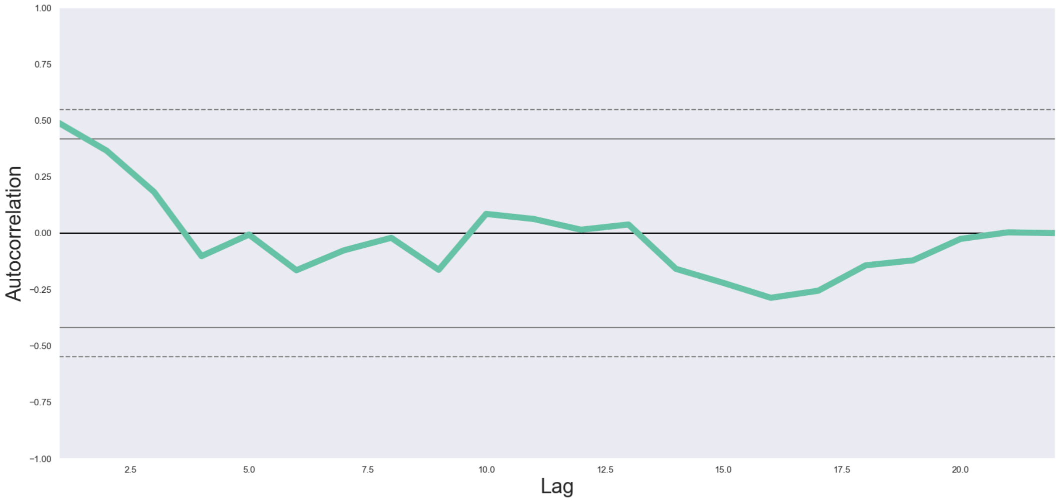


Figure 4.3: Temporal auto-correlation of absolute municpal CVDs between 1996 and 2017 in the Netherlands.

4.1.2 Dutch temperature

The Dutch temperature data set contains 23 weather stations that measured average daily temperature. Weather stations are represented in vector data using points. Each weather station has 10228 timestamps representing all the days in the years 1991-2018. No further pre-processing was necessary, to make the data set suitable for analysis. Each weather station was assigned its three closest neighbors to calculate spatio-temporal statistics. Figure 4.4 displays the locations of the weather station in the Dutch temperature data set. For this data set the figures presented for the other data sets were not included, due to the large amount of time steps these became visually cluttered, the corresponding statistics are as following: The average Moran's I, standard deviation and p-value are 0.4, 0.2 and 0.032 respectively. The positive spatial auto-correlation indicates a spatial pattern in the data set. The temporal auto-correlation is higher with a lower lag, and diminishes, furthermore, every 12 months the lag becomes significant again, indicating seasonal similarity. This effect is significant (temporal correlation = 0.3, p = 0.031). From the above figures it can be concluded that the Dutch temperature data set is characterized by spatial and temporal correlation.

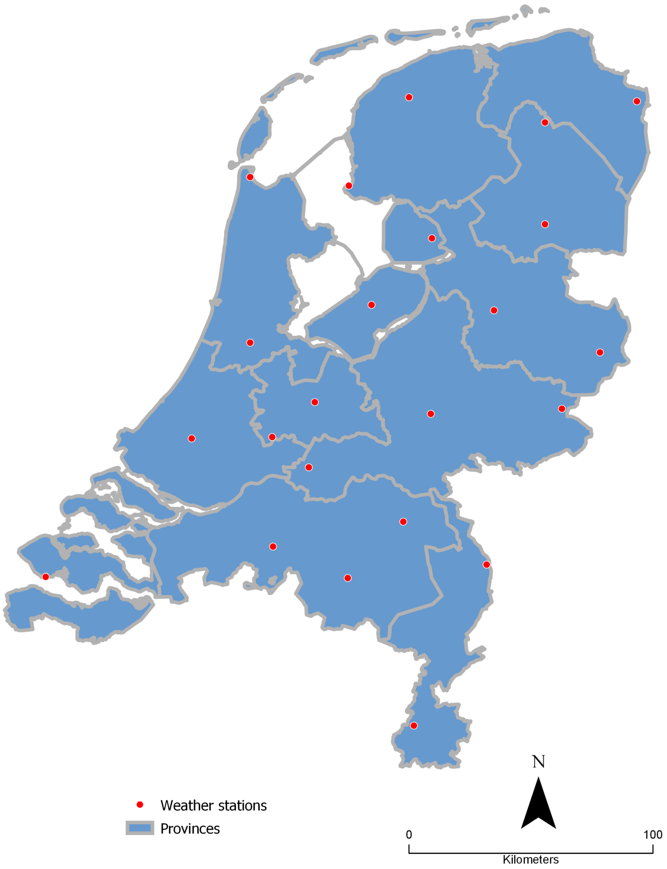


Figure 4.4: Locations of weather station present in the Dutch temperature data set.

4.1.3 Snow depth in Washington State

The snow depth in Ohio state data set contains 365 days of the year 2016. The data set is modelled from a set of weather stations. Both temporal and spatial structures are created using an Inverse-distance weighing based model (National Weather Service's National Operational Hydrologic Remote Sensing Center, 2019). This creates a strong spatial pattern in the data set as more thoroughly described in the next paragraph, such an pattern should be easily adapted by a kriging based model as well. The data is represented in a raster format with a spatial resolution of 1000 metres. For analysis 365 single .tif images, each representing one day, were converted to a multiband raster data set. No further pre-processing had to be performed to make the data set suitable for analysis. Spatial weights were set to queen contiguity. Figure 4.5 displays the 100th timestep and the spatial extent of the data set.

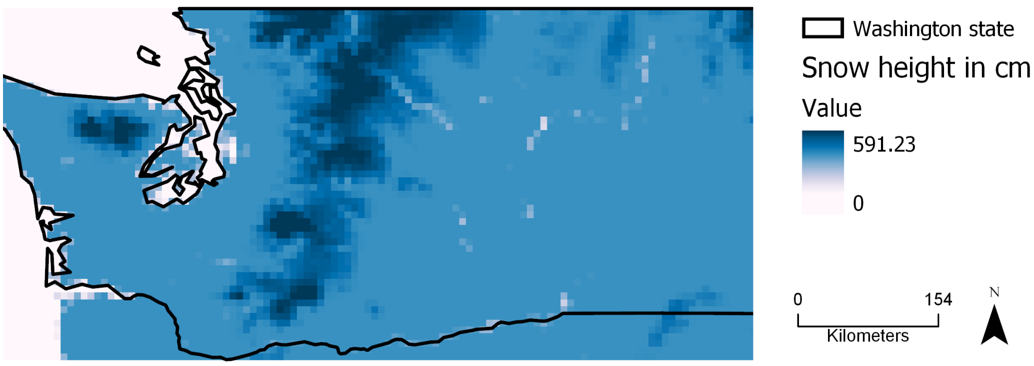


Figure 4.5: The snowheight at the 100th day of the 2016.

Figure 4.6 summarizes the Morans'I for all the timesteps in the data set. The average Moran's I and standard deviation, and p value are, 0.998, 0.001, and 0.001 respectively. Which indicates there is a very strong spatial pattern. The spatial auto-correlation is positive, therefore, values close in space are more alike. Figure 4.7 summarizes the temporal auto-correlation. The figure displays a perfect linear downward correlation between lag and auto-correlation (t=0.99, p=0.001). To conclude, these strong spatial and temporal patterns are ideal to test the capabilities of the BBAC algorithm. Ideally, the algorithm should be able to detect these patterns, and predict comparable results to spatial-temporal methods. Moreover, due to this structure it is expected that co-kriging produces the most accurate results.

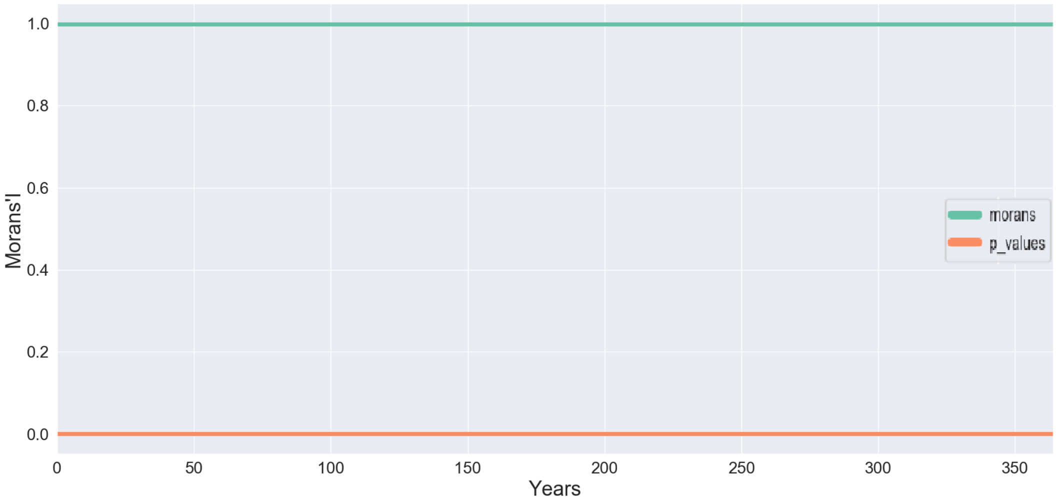


Figure 4.6: Morans' I and corresponding p-values of daily snow depth in Washington State in 2016.

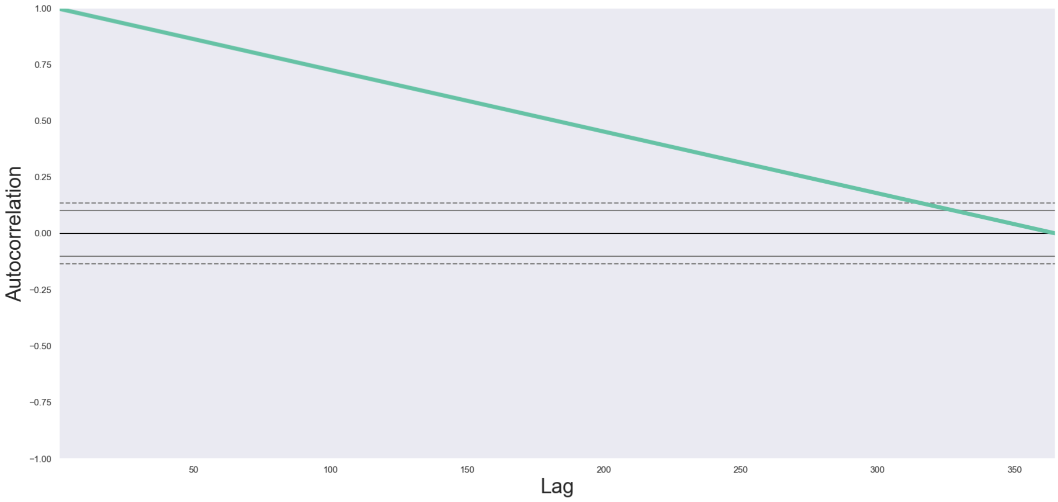


Figure 4.7: Temporal autocorrelation of daily snow depth in Washington State in 201 .

4.1.4 First Leaf Dates in Ohio State

The first leaf dates in Ohio data set contains the first leaf date of every year between 1980 and 2016. The data is represented in raster format with a spatial resolution of 4000 metres. For analysis 37 single .tif images, each representing one year, were converted to a multiband raster. No further pre-processing had to performed to make the data set suitable for analysis. Spatial weights were set to queens contiguity. Figure 4.8 displays the first leaf dates for 1980, the first times tep of the data set.

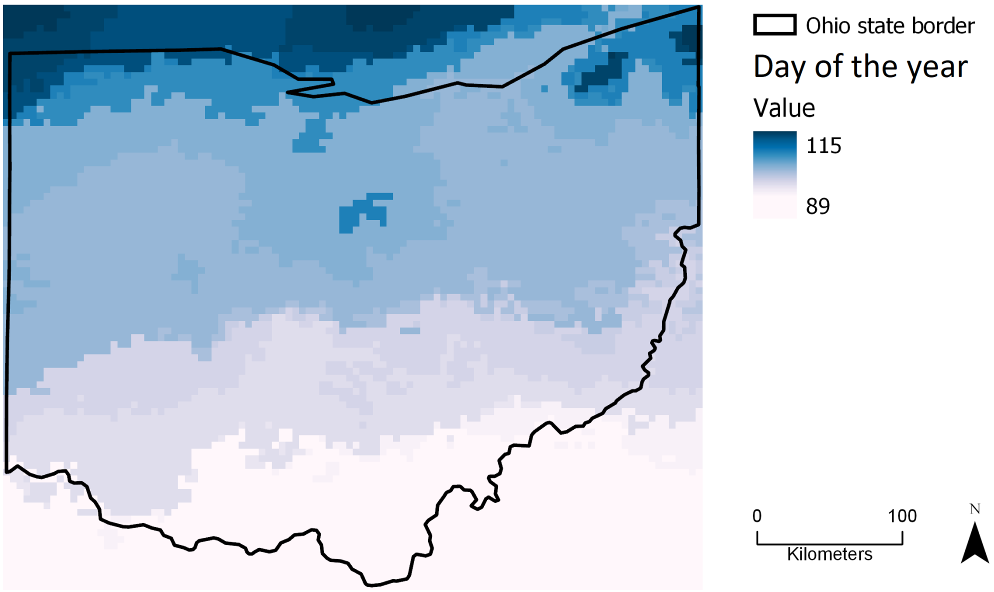


Figure 4.8: The first leaf dates in Ohio state in 1980.

Figure 4.9 summarizes the Morans'I for all the time steps in the data set. The average Moran's I and standard deviation, and p value are 0.984, 0.003, and 0.001 respectively. The spatial auto-correlation is positive, therefore, values close in space are more alike. Figure 4.10 summarizes the temporal auto-correlation. The temporal correlation was positive but not significant for the whole data set (t=0.11, p=0.078). The two closest years were most similar, after two years the lag is more randomly distributed. To conclude, the first leaf date data set has a strong spatial pattern, and a limited temporal correlation.

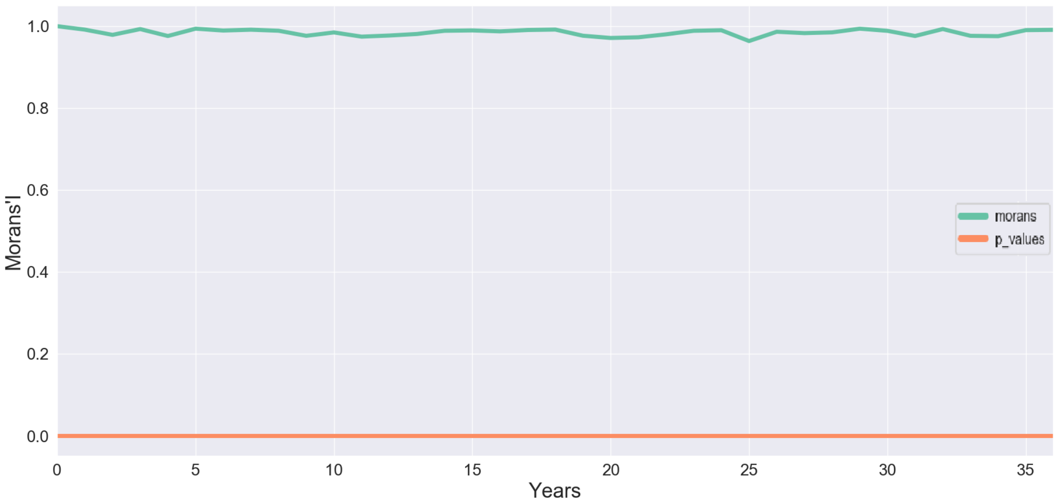


Figure 4.9: Morans' I and corresponding p-values of first leaf dates in Ohio State between 1980 and 2016.

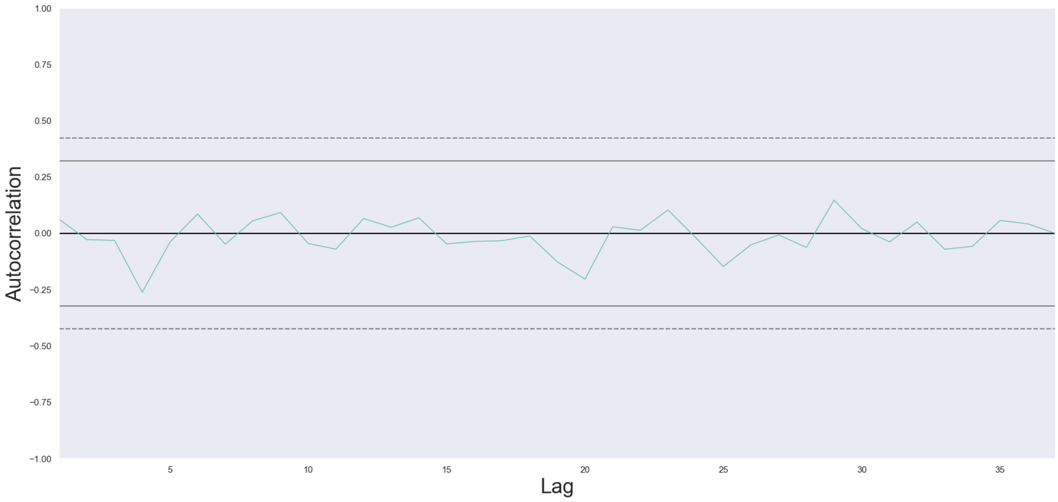


Figure 4.10: Temporal autocorrelation of first leaf dates in Ohio State between 1980 and 2016.

4.2 Cross validation and hyper parameters

The four tables (4.1, 4.2 , 4.3, and 4.4) show the grid search results for each separate data set. Hastie et al. (2009) stated that machine learning models operate as a black box without examining the parameters. This is valid for both reproducing the research and understanding the results from imputation. Three main conclusion can be drawn from the tables. Firstly, BBAC scheme 2 and 5 have similar row and column parameters to produce the most accurate results, which indicates that their various summary statistics are most accurate at the same number of co-clusters. Secondly, Support vector machine consistently produces the most accurate result using the rbf kernel, which is supported by the advice from the Scikit-learn documentation about their default settings for SVM (Pedregosa et al., 2011). Thirdly, Random forest does not lead to a substantially higher accuracy when applying a higher number of leafs. The differences in accuracy were less than 2 percent. Therefore, a lower number of leafs (10) was selected, because a lower number of leafs decreases the computational requirements of the model substantially.

Table 4.1: Grid search parameter optimization results for the CVD data set.

Model	**Parameter 1**	**Parameter 2**	**Parameter 3**
k-neareast Neighbour	Number of neighbors: 9	Distance metric: Uniform.	
Support vector machine	Kernel method: rbf	C value: 1.	0.0001
Naive Bayes	Maximum number of iterations: 300.	Fit_intercept: False	
Random Forest	Number of leafs (n_estimators): 10	Minimum split samples required to split a leaf: 2	
XGBoost	Loss function: is (Least squares regression)	Number of estimators: 100	
BBAC scheme 2	Number of row clusters: 10.	Number of column clusters: 5.	Distance metric: I-Divergence
BBAC scheme 5	Number of row cluster: 10	Number of column clusters: 5.	Distance metric: I-divergence

Table 4.2: Grid search parameter optimization results for the Dutch Temperature data set.

Model	**Parameter 1**	**Parameter 2**	**Parameter 3**
k-neareast Neighbour	Number of neighbors: 10	Distance metric: Uniform.	
Support vector machine	Kernel method: rbf	C value: 1.	0.01
Naive Bayes	Maximum number of iterations: 100.	Fit_intercept: False	
Random Forest	Number of leafs (n_estimators): 10	Minimum split samples required to split a leaf: 2	
XGBoost	Loss function: lad (least absolute deviaiton)	Number of estimators: 100	
BBAC scheme 2	Number of row clusters: 4.	Number of column clusters: 14.	Distance metric: I-Divergence
BBAC scheme 5	Number of row cluster: 6	Number of column clusters: 14.	Distance metric: I-divergence

Table 4.3: Grid search parameter optimization results for the Snow Depth.

Model	Parameter 1	Parameter 2	Parameter 3
k-neareast Neighbour	Number of neighbors: 6	Distance metric: Distance.	
Support vector machine	Kernel method: rbf	C value: 1.	0.0001
Naive Bayes	Maximum number of iterations: 300.	Fit_intercept: True	
Random Forest	Number of leafs (n_estimators): 10	Minimum split samples required to split a leaf: 1	
XGBoost	Loss function: Is (least squares regression)	Number of estimators: 50	
BBAC scheme 2	Number of row clusters: 30.	Number of column clusters: 30.	Distance metric: I-Divergence
BBAC scheme 5	Number of row cluster: 27	Number of column clusters: 31.	Distance metric: I-divergence

Table 4.4: Grid search parameter optimization results for the First leaf Dates.

Model	Parameter 1	Parameter 2	Parameter 3
k-neareast Neighbour	Number of neighbors: 15	Distance metric: Uniform.	
Support vector machine	Kernel method: rbf	C value: 1.	0.001
Naive Bayes	Maximum number of iterations: 100.	Fit_intercept: False	
Random Forest	Number of leafs (n_estimators): 10	Minimum split samples required to split a leaf: 2	
XGBoost	Loss function: Is (least squares regression)	Number of estimators: 10	
BBAC scheme 2	Number of row clusters: 4.	Number of column clusters: 11.	Distance metric: I-Divergence
BBAC scheme 5	Number of row cluster: 4	Number of column clusters: 11.	Distance metric: I-divergence

4.3 Model results

4.3.1 Cardiovascular deaths in the Netherlands

Tables 4.5, 4.6, and 4.7 summarize the prediction results for 10, 20 and 30 percent missing values for the cardiovascular death data set. BBAC_2 produced the most accurate predictions for both MAE and RMSE, however unexpectedly BBAC_5 produced less accurate results. BBAC_5 uses more summary statistics to reconstruct a matrix, therefore it is expected that this method yields more accurate results than scheme_2, which is not true for any of the data sets. This contradictory results is discussed in section 4.4.2. The Bayes model produced the most accurate predictions according to both the MAE and RMSE. The SVM and BBAC_5 predictors produced the least accurate predictions. Although KNN does have a low MAE score, its RMSE score is relatively high, which indicates that the prediction created outliers. These findings indicate that some of the one-way clusters created overgeneralize the data. Both BBAC models and the SVM imputation required 3.65 seconds to process, the least off all models. Co-kriging required the most processing time, yet produced the least accurate results for all percentages. Bayes and RF took only seven seconds, twice the time of the former. KNN and GB both required more processing time, 14.6 and 36.5 seconds respectively. The tables show the change in MAE over increasing percentages of missing values. For 20 percent missing values the MAE and RMSE of all models increased, the accuracy between the models did not change. In contrast with all machine learning models, the processing time for both BBAC models did not increase. For 30 percent the MAE and RMSE of all models increased, the accuracy between models did not change. The processing time of BBAC_2 remained 3.65 seconds, while the BBAC_5 increased to 7.29 seconds. However, the prediction time of SVM and GB was reduced, although this reduction was coupled with an increase in both MAE and RMSE. None of the models output were spatially correlated, which is in line with the findings from the exploratory data analysis that the model did not have a significant spatial pattern.

Table 4.5: CVD death imputation results 10 percent missing.

Model	**MAE**	**RMSE**	**Runtime (seconds)**	**Moran's I**	**Moran's I p-value**
Bayes	0.90	4.18	7.29	0.066	0.157
SVM	5.46	53.17	7.29	0.022	0.201
KNN	2.01	31.61	14.60	0.087	0.125
GB	1.14	7.82	36.47	0.035	0.166
RF	1.23	10.22	7.29	0.048	0.126
BBAC_2	1.47	9.33	3.65	0.032	0.229
BBAC_5	7.66	57.1	3.65	0.003	0.200
Kriging	88.03	195.42	13	0.024	0.201

Table 4.6: CVD death imputation results 20 percent missing.

Model	**MAE**	**RMSE**	**Runtime (seconds)**	**Moran's I**	**Moran's I p-value**
Bayes	1.74	5.89	7.29	0.066	0.157
SVM	12.32	69.45	28.30	0.093	0.197
KNN	3.8	39.45	14.60	0.131	0.151
GB	2.36	12.78	36.47	0.106	0.188
RF	2.43	14.38	7.29	0.128	0.154
BBAC_2	2.93	13.94	3.65	0.073	0.172
BBAC_5	15.53	79.75	3.65	0.066	0.194
Kriging	85.65	188.45	18	0.003	0.200

Table 4.7: CVD death imputation results 30 percent missing.

Model	**MAE**	**RMSE**	**Runtime (seconds)**	**Moran's I**	**Moran's I p-value**
Bayes	2.81	8.40	21.90	0.105	0.158
SVM	17.36	91.56	25.87	0.071	0.171
KNN	5.88	60.42	14.60	0.136	0.136
GB	3.48	19.3	10.10	0.109	0.178
RF	3.56	21.99	10.10	0.134	0.099
BBAC_2	4.04	16.95	3.65	0.108	0.166
BBAC_5	22.09	91.35	7.29	0.095	0.192
Kriging	83.21	186.29	17.5	0.066	0.157

4.3.2 Dutch temperature

Tables 4.8, 4.9, and 4.10 summarize the prediction results for 10, 20, and 30 percent missing values for the Dutch temperature data set. All the results are in tenths of degrees Celsius, a residual of 4, indicates 0.4 degrees Celsius prediction error. Notably, kriging did not produce the best results, even though the EDA concluded that the data set is characterized by positive spatial auto-correlation. Again, BBAC_2 produced the most accurate predictions for both MAE and RMSE. All of the other models produced higher, but similar results. There were no major differences between MAE and RMSE. Bayes and GB's runtimes were nearly 100 hours, therefore those models were excluded from the 20 and 30 percent missing run. The MAE a percentages of BBAC slightly increased compared to missing value percentage missing values, the same pattern was seen 20 and 30 percent the MAE, RMSE and processing times of both BBAC models slightly increased. The machine learning models' MAE decreased slightly when increasing the percentage of missing values. Although the original data set was characterized by spatial auto-correlation, all of the models did not produce statistically significant spatially auto correlated residuals, which indicates that the models captured the spatial structure of the data set.

Table 4.8: Dutch temperature imputation results 10 percent missing.

Model	**MAE**	**RMSE**	**Runtime (seconds)**	**Moran's I**	**Moran's I p-value**
Bayes	4.05	12.53	358225	-0.391	0.143
SVM	7.46	11.06	9085	-0.355	0.126
KNN	5.87	13.73	868	-0.385	0.131
GB	4.7	12.99	32579	-0.386	0.144
RF	5.07	13.22	46617	-0.235	0.190
BBAC_2	1.12	5.63	870	-0.328	0.120
BBAC_5	5.33	20.44	705	-0.327	0.116
Kriging	10.59	25.44	412523	-0.384	0.137

Table 4.9: Dutch temperature imputation results 20 percent missing.

Model	**MAE**	**RMSE**	**Runtime (seconds)**	**Moran's I**	**Moran's I p-value**
Bayes	-	-	-	-	-
SVM	7.06	10.99	9115	-0.178	0.163
KNN	5.64	11.31	950	-0.181	0.172
GB	4.22	10.19	399049	-0.249	0.196
RF	4.67	10.54	41216	-0.235	0.190
BBAC_2	1.98	7.05	850	-0.136	0.158
BBAC_5	9.84	27.55	709	-0.136	0.156
Kriging	19.54	35.44	394523	-0.249	0.196

Table 4.10: Dutch temperature imputation results 30 percent missing.

Model	MAE	RMSE	Runtime (seconds)	Moran's I	Moran's I p-value
Bayes	-	-	-	-	-
SVM	6.88	10.95	8837	-0.931	0.145
KNN	5.29	10.13	1427	-0.065	0.190
GB	-	-	-	-	-
RF	4.62	9.53	41790	-0.12	0.212
BBAC_2	2.75	7.05	873	0.032	0.157
BBAC_5	13.96	32.8	810	0.032	0.159
Kriging	18.64	31.44	355684	-0.146	0.159

4.3.3 Snow depth in Washington State

Table 4.11, 4.12, and 4.13 summarize the prediction results for 10, 20, and 30 percent missing values for the Washington State snow depth data set. All the results are in thousands of a meter snow depth, a residual of 4000, indicates a snow depth of 4 metres prediction error. The main finding is that co-kriging outperforms all other models on its accuracy and does not result in spatially correlated residuals. Another findings is that SVM was not able to converge after 100 hours of run time, even though the computer running the models should have had plenty of computational power (32gb ram DDR4, 4gb dedicated graphics card). For all percentages the machine learning models outperform the BBAC algorithms by a large margin. Although the non spatial models are more accurate, they produced large differences between MAE and RMSE, which indicates that outliers are present in the imputation results. All the models' MAE increased when increasing the percentage of missing values. Furthermore, all of the non-spatial models produced a statically significant positive spatial auto-correlation, yet co-kriging did. which is in line with the theory and exploratory analysis. The snow depth data set was derived from a inverse distance weighing approach, which should be easy to capture for a kriging method.

Table 4.11: Snow depth imputation results 10 percent missing.

Model	MAE	RMSE	Runtime (seconds)	Moran's I	Moran's I p-value
Bayes	115.12	841	2443	0.14	0.001
SVM	-	-	-	-	-
KNN	753	5658	13776	0.14	0.001
GB	263	2694	40562	0.14	0.001
RF	239	3120	15013	0.14	0.001
BBAC_2	2147	7623	653	0.05	0.001
BBAC_5	37150	42197	5278	0.065	0.001
BBAC_5	201	1381	41652	0.065	0.001
Kriging	106	241	15884	0.077	0.137

Table 4.12: Snow depth imputation results 20 percent missing.

Model	MAE	RMSE	Runtime (seconds)	Moran's I	Moran's I p-value
Bayes	157	1180	3201	0.14	0.001
SVM	-	-	-	-	-
KNN	782	5715	13413	0.14	0.001
GB	280	2743	42732	0.14	0.001
RF	266	3373	12704	0.14	0.001
BBAC_2	1187	7391	1076	0.104	0.012
BBAC_5	31468	38040	4660	0.1465	0.001
Kriging	154	510	21984	0.105	0.156

Table 4.13: Snow depth imputation results 30 percent missing.

Model	MAE	RMSE	Runtime (seconds)	Moran's I	Moran's I p-value
Bayes	198	1327	3611	0.14	0.001
SVM	-	-	-	-	-
KNN	814	5488	18835	0.14	0.001
GB	294	2672	3611	0.14	0.001
RF	277	3138	14841	0.14	0.001
BBAC_2	2107	7391	1633	0.14	0.012
BBAC_5	51220	54622	16276	0.148	0.001
Kriging	161	430	17273	0.08	0.194

4.3.4 First Leaf Dates in Ohio

Tables 4.14, 4.15, and 4.16 summarize the prediction results for 10, 20 and 30 percent missing values for the first leaf dates data set. Co-kriging produced a striking result: even though the original data set was characterized by positive spatial auto-correlation its accuracy was lower than all other models and the residuals were spatially correlated as well. Moreover, KNN, GB, and RF produced the most accurate predictions according to both the MAE and RMSE. SVM and both BBAC models produced the least accurate predictions. The MAE and RMSE do not indicate different results. Bayes was the fastest model. BBAC_2 was the second fastest, however it was two times slower than Bayes. GB and BBAC_5 had respectable processing times as well. SVM, KNN and RF took magnitudes longer to process, varying from just over 20 minutes to nearly three hours. The tables show the change in MAE over increasing percentages of missing values. The results for 20 percent missing values indicate similar prediction accuracy as for 10 percent missing. For all models MAE, RMSE and processing time slightly increased. The results for 30 percent missing values indicate similar results as for 10 and 20 percent. The MAE, RMSE and runtime marginally increased, while the spatial properties remained the same. All of the non-spatial models produced significantly positively spatially correlated residuals, which indicates that the models' predictions do not capture the spatial properties of the original data set.

Table 4.14: First leaf dates imputation results 10 percent missing.

Model	MAE	RMSE	Runtime (seconds)	Moran's I	Moran's I p-value
Bayes	1.36	1.99	76	0.194	0.01
SVM	3.61	5.88	14982	0.194	0.001
KNN	0.48	1.09	1082	0.194	0.001
GB	0.72	1.23	578	0.194	0.001
RF	0.34	0.90	2950	0.194	0.001
BBAC_2	2.70	3.80	184	0.074	0.001
BBAC_5	8.27	10.32	408	0.113	0.0124
Kriging	13.01	17.31	8854	0.188	0.001

Table 4.15: First leaf dates imputation results 20 percent missing.

Model	MAE	RMSE	Runtime (seconds)	Moran's I	Moran's I p-value
Bayes	1.40	2.03	120	0.194	0.001
SVM	4.22	6.80	14505	0.194	0.001
KNN	0.51	1.19	1590	0.194	0.001
GB	0.74	1.26	617	0.194	0.001
RF	0.40	1.00	3765	0.194	0.001
BBAC_2	2.65	3.59	215	0.130	0.001
BBAC_5	8.27	10.35	408	0.06	0.008
Kriging	13.62	19.71	9014	0.187	0.001

Table 4.16: First leaf dates imputation results 30 percent missing.

Model	MAE	RMSE	Runtime (seconds)	Moran's I	Moran's I p-value
Bayes	1.50	2.16	78	0.194	0.001
SVM	4.55	7.06	14363	0.194	0.001
KNN	0.56	1.32	2306	0.194	0.001
GB	0.75	1.30	679	0.194	0.001
RF	0.44	1.07	3479	0.194	0.001
BBAC_2	2.71	3.65	178	0.194	0.001
BBAC_5	8.28	10.34	321	0.168	0.002
Kriging	14.12	19.95	8940	0.191	0.001

4.4 Discussion of results

4.4.1 Summary of results

Analysis of the comparison results for non-spatial models indicated that the BBAC-I scheme 5 algorithm yielded the least accurate predictions. Kriging produced less accurate predictions on the CVD data set, but produced more accurate predictions on all other data sets. BBAC-I scheme 2 produced more accurate results and was one of the best performing models, except for the SNODAS data set, for which it produced one of the least accurate predictions.

The BBAC models did not produce any spatially auto-correlated residuals for the CVD (no spatial pattern), and Dutch temperature (positive spatial auto-correlation) data sets. However, both models produced positively spatially auto-correlated residuals for the SNODAS and first leaf date data sets. These data sets were characterized by a high positive spatial auto-correlation. These results were comparable to the selection machine learning models, which resulted in the same kind of spatial outcomes. The spatio-temporal model was less accurate but the spatial configuration of the residuals was not significant.

The results for the runtime analysis indicate that both the BBAC models require less processing time to impute data than the selection of machine learning and spatio-temporal models. On the SNODAS data set the BBAC-I scheme 5 algorithm could not converge after the maximum number of iterations, resulting in a much larger processing time. The BBAC-I scheme 2 algorithm consistently was the fastest or among the fastest models for every data set and missing value percentage. Although some machine learning models required up to 100 hours of processing time for the imputation of a single data set, most machine learning models were faster than their spatial counterpart.

4.4.2 Interpretation of results

The first objective of this thesis was to explore the accuracy of the BBAC-I algorithm in missing data imputation in spatial data sets. The MAE and RMSE results differ greatly between BBAC-I scheme 2 and 5. Scheme 2 produced more accurate imputation results for every data set. These results were unexpected, in the chapter 2 both matrix reconstruction methods were discussed and BBAC_5 produced more accurate results in the original publication from (Banerjee et al., 2007), based on this information (George, 2005) applied scheme 5 in his work. (Banerjee et al., 2007) explained that scheme 5 produced more statistics on each co-cluster and therefore reconstructs a more informative matrix. To clarify the discrepancies between this thesis and the earlier results the following steps were taken: (1) re-run the results, (2) run the implementation in this thesis on a data set used in earlier publications, (3) manually inspect every intermediate step. The first two steps did not lead to any insightful findings, the third step was more use full. Scheme 2 uses the co-cluster average, additionally scheme 5 uses the co-cluster, row, and column averages for matrix reconstruction. Manual inspection showed that the row and column average values were meaning full in the MovieLens data set, yet were not insight full for the CVD and First Leaf Date data sets. The row and column averages did not resemble the original values and skewed imputation results.

As an result, scheme 5 did not produce accurate imputation results, and was ranked between the lowest scoring models for each data set and percentage. Scheme 2 did produce more accurate results, and often ranked among the higher scoring models, but was not the highest ranking model. Furthermore there were no inconsistencies between its MAE and RMSE, which indicates that its does not create many outliers. BBAC-I scheme 2 did outperform co-kriging on most data sets, but this does not conclude anything about the spatial configuration of the residuals. These findings are in line with one of the main ideas Hastie et al. (2009) proposes in his machine learning book: model selection and parameter optimization are data set specific, thus imputation should be addressed with research purpose and data set in mind. For example by thorough exploratory data analysis and producing test results on small subsets of the data set.

The second objective of this thesis was to investigate the computational effectiveness of the BBAC-I algorithm compared to other models. Banerjee et al. (2007) proposed the BBAC-I algorithm as an computationally efficient co-clustering algorithm that would require short processing times, particularly when scaling the algorithm to large data sets. The results from the runtime analysis are in line with earlier research in the collaborative filtering domain, that BBAC imputation is not only fast, but scales well on larger data sets (George, 2005; Banerjee et al., 2007). George (2005) used a subset of the MovieLens data set (n values = 7500) which size is comparable to the Cardiovascular Death data set (n values = 6710), and the Dutch temperature data set (n values = 235244) which size is comparable to Book Crossing data set (n values = of 269392). George (2005) states that the BBAC algorithm is faster on both data sets. However, the difference increased a number of folds for the larger data set. This is in line with the findings from this study: BBAC 2 and 5 require half the processing time for the CVD, and between a fifth and a tenth of the time compared to the other models on the Dutch temperature data set. Likewise, these findings are comparable to the First Leaf Date and Snow Depth Data sets. The number of values in these data sets are not comparable to data sets used in earlier publications on the subject, but they show the same results: a larger data set, requires a lower comparative run time to both machine learning and spatial imputation models. The cause of improvement in computation time for larger data sets is described by Banerjee et al. (2007, p. 1960):

> *... the training time is linear in the number of known ratings and the missing value prediction is a constant time operation unlike in other approaches. The number of parameters in the compressed representation is also much lower in the case of co-clustering.*

Furthermore, the BBAC models were both faster than their spatial counterparts. This is in line with earlier findings from Aalto et al., 2013, that concluded that co-kriging requires a large amount of distance matrices. This is due to the structure of spatial imputation models. Spatial models create a corresponding distance matrix for each of the value in the original data set. It is no surprise that creating, extracting and processing a separate matrix for each value, and then combining them for multiple time steps in a data set requires more processing time.

The third objective of this thesis was to explore the usability of the BBAC-I algorithm for missing value imputation in data sets with high spatial auto-correlation. The BBAC-I algorithms did not produce significant Moran's I for the CVD and Dutch temperature data sets. These results were expected for the CVD data set, because the original CVD data set was not spatially correlated. Although the Dutch temperature data set was spatially correlated, both BBAC schemes did not produce spatially correlated residuals and all of the machine learning models produced the similar results. The same pattern was seen in the results from the Snow depth and First leaf dates data set: both the BBAC-I and machine learning models were not able to capture the spatial pattern of the data set, resulting in significant positively spatially auto-correlated residuals. This indicates that the BBAC algorithm performs equally inaccurate spatial imputation results as machine learning models on data with a strong spatial pattern. The spatial model did not produce the best results on any data set or percentage, except for the data set (Snow depth) that was created with an distance based model. This does indicate that spatial models are still the first choice that comes to mind for spatial data imputation accuracy, but certainly not for the fastest run time.

Chapter 5

Discussion and Conclusion

5.1 Discussion of methods

5.1.1 Strengths and weaknesses

The aim of this study was to asses the value of BBAC-I in GTS imputation by comparing it to models from the spatial and machine learning domain. The spatial configuration of the residuals were analyzed to evaluate the spatial properties of each model. An improvement to the study is introducing a spatial pattern in the missingness of the data, for example spatial MNAR. This would allow a more robust analysis that leads to results on data with a spatial pattern, but also to data with a spatial pattern in the missingness.

The original selection spatio-temporal models contained a state of the art imputation model, Gapfill-Map, for structured spatio-temporal data. The authors of this model mentioned that both raster and vector data should work as input data for their model. However, this was a theoretical notion, to actually make it work on both types of geographical data would have required a lot of work outside the scope of this thesis. Therefore the selection of spatio-temporal models was limited to co-kriging. Co-kriging, a form of kriging, is considered robust, easy to implement, and has been tried and tested for several decades, yet can not be considered state of the art. Therefore, the innovation in newer models is overlooked in this study, and the possibility remains that a more recent selection of spatio-temporal models would produce better imputation results. The same applies to the selection of machine learning models. This selection comprises the most used, studied and implemented models available. Newer, or more specified models could lead to more accurate predictions. Yet the increase in accuracy by newer models is limited, and often adjusted on a specific type of data in mind. The application of such a model to a multitude of distinctly different data sets might lead to uncertainties in accuracy (Hastie et al., 2009).

The spatio-temporal models from ESRI's ArcGIS and Scikit-learns' machine learning models are maintained by professional software development teams and a large community of open-source developers, resulting in highly optimized tools that are written in language bindings computationally more efficient than vanilla Python. The BBAC-I code is produced in R, and the imputation in Python, the combination of those languages makes the imputation process less efficient. Moreover, the BBAC-I code was not written by a professional software team or large pool of open-source developers. Its optimization is limited and processing times and accuracy can be expected to improve by a more professional approach. Nonetheless, the BBAC-I algorithm required less processing times than other models, which amplifies its potential strength in large data sets.

All models were implemented using forward chaining cross-validation to prevent overfitting and adjust for temporal patterns in the data. Besides overfitting, cross-validation also optimizes a models' parameters, thus the selected models have been used with an appropriate configuration of parameters. Yet, these cross-validation has been applied to the 10 percent missing data set only. Although the pattern of missing values is completely random, an more appropriate approach would have been to run the cross-validation on each percentage missing values. However, this would have required an even larger run time. Nonetheless, this is a major weakness in the cross-validation. Overall, the benefits of cross-validation lead to a selection of models with a generalized performance that is applicable to a broader selection of data sets. Moreover, all the models were fit to multiple spatial data sets with distinct spatial properties and imputation was tested on different percentages of missing data. The combination of forward chaining, parameter optimization and the use of distinct data sets and percentages leads to conclusion that can be interpreted more generally.

Analyzing a selection of either the spatial or non-spatial models would have given more resources to thoroughly research those models, leading to more specific results for either spatial or non-spatial models. Yet the selection of both spatial and non spatial imputation methods resulted in a broader understanding of these models in the spatial domain. Using spatial models on data sets with a spatial pattern did not consistently yield better results, these findings indicate that applying non-spatial methods to impute spatial data is an option to consider. Therefore it can be concluded that the comparison of models from both the machine learning and spatial imputation domain has lead to greater insight as previously argued in subsection 1.3.

5.1.2 Recommendations for future research

Future research should continue to address differences in spatial and non-spatial imputation methods to determine in which context non spatial models are applicable in the spatial domain. Moreover, there is a large scientific output discussing new machine learning methods that could be viable in the spatial domain. Subsequently, a study conducted about new methods adds scientific relevance, yet the extra work required to implement a set of new models for different types of geographical data should not be overlooked.

Moreover, the BBAC algorithm has not proven to be a reliable imputation method for data with high levels of positive auto-correlation. However, this has only been observed for GTSs, a structured form of spatio-temporal data. This leaves room for future research that addresses unstructured spatio-temporal data, such as GPS tracks. These forms of data require neighbour matrices for each irregular time step, resulting in a large number of distance matrices compared to structured data. Although BBAC accuracy is not unrivalled, the computation effectiveness observed for BBAC in this study could lead to lower computational requirements for unstructured spatio-temporal data, which can be a worthwhile trade-off.

The accuracy results of the BBAC model in this study are not overwhelming. This does not indicate an incapability of all non-spatial models in the spatial domain, but does emphasize that non-spatial models need to be applied with caution. Moreover, model selection for spatial data imputation should be aimed at selecting the most appropriate model for a data set. In some cases BBAC-I would be the most appropriate, in others a machine learning model or spatial interpolation model. Future research should continue to address the applicability of non-spatial models in the spatial domain.

5.2 Conclusion

This research is the first to explore the value of the Bregman Black Average co-clustering algorithm with I-divergence (BBAC-I) for missing value imputation in structured spatio-temporal data (Geo-referenced time series). The study compared the BBAC-I algorithm scheme 2 and 5 with spatial interpolation models and machine learning models to impute missing values in four GTSs with distinct spatial properties. Three objectives were set to explore the algorithm in this context: (1) compare the prediction accuracy, (2) compare the computational run time, (3) analyze the spatial properties of the prediction residuals.

The results from the first objective indicate that both BBAC-I schemes produce less accurate imputations than the selection of machine learning models, yet outperform spatial models. BBAC-I scheme 2 outperformed scheme 5, yet did not consistently produce the most accurate results compared to the selection of machine learning models.

The literature indicated that both BBAC-I models require a relatively small amount of computational requirements, this was confirmed in this study: both BBAC-I models required less processing time than all other models, especially on larger data sets. To conclude, the results from the second goal indicate that BBAC-I's suitability increases with the size of the data set.

The results from the third goal indicate that the BBAC-I models are not suitable for imputation in data sets with a high positive spatial auto-correlation. Moreover, BBAC-I performs the same as machine learning models, but is less accurate than spatial interpolation models. To conclude this thesis the main research question introduced in chapter 1 is addressed:

> *To what extent can the Bregman Block average co-clustering algorithm schemes two and five with I-Divergence be used to impute missing data in geo-referenced time series?*

In conclusion, the imputation results from both BBAC-I scheme 2 and 5 were not overwhelming, the BBAC-I scheme 2 outperformed scheme 5 on run time, accuracy, and spatial accuracy. Although BBAC-I scheme 2 imputation results were comparable to the selection of machine learning models concerning accuracy and spatial accuracy, they both produced lower spatial accuracy than their spatial counterpart. Moreover, the BBAC-I models outperformed any other spatial and non-spatial model in the run time analysis. The value of BBAC-I for missing value imputation lies in a limited selection of data sets that are very large, and for which limiting computational requirements is more important than accuracy and non-spatially correlated predictions. Future research should continue to address the value of recently developed non spatial models in the spatial domain.

Bibliography

Aalto, J., Pirinen, P., Heikkinen, J., & Venäläinen, A. (2013). Spatial interpolation of monthly climate data for Finland: Comparing the performance of kriging and generalized additive models. *Theoretical and Applied Climatology*, *112*(1-2), 99–111. doi:10.1007/s00704-012-0716-9

Aggarwal, C. (2016). *Recommender systems The Textbook*. doi:10.1145/245108.245121. arXiv: arXiv:1202.1112v1

Allison, P. D. (1987). Estimation of Linear Models with Incomplete Data. *17*(1987), 71–103.

Anseling, L. (1995). Local Indicators of Spatial Association-LISA. *Geographical analysis*, *27*(2).

Baker, J., White, N., & Mengersen, K. (2014). Missing in space: An evaluation of imputation methods for missing data in spatial analysis of risk factors for type II diabetes. *International Journal of Health Geographics*, *13*(1). doi:10.1186/1476-072X-13-47

Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., & Modha, D. S. (2004). A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *Proceedings of the 2004 acm sigkdd international conference on knowledge discovery and data mining - kdd '04*. doi:10.1145/1014052.1014111

Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., & Modha, D. S. (2007). A Generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation. *JOURNAL OF MACHINE LEARNING RESEARCH*, *8*, 1919–1986. doi:10.1023/B:JARS.0000021016.61054.3b

Batista, G. E. A. P. A. & Monard, M. C. (2002). A Study of K-Nearest Neighbour as an Imputation Method. *HIS'02: 2nd International Conference on Hybrid Intelligent Systems*, 251–260. Retrieved from http://conteudo.icmc.usp.br/pessoas/gbatista/files/his2002.pdf

Bergmeir, C. & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, *191*, 192–213. doi:10.1016/j.ins.2011.12.028

Bland, J. M. (2000). An Introduction to Medical Statistics, 3rd edition. *Oxford University Press*.

Brand, M. (2002). Incremental Singular Value Decomposition of Uncertain Data with Missing Values. In A. Heyden, G. Sparr, M. Nielsen, & P. Johansen (Eds.), *Computer vision — eccv 2002* (pp. 707–720). Berlin, Heidelberg: Springer Berlin Heidelberg.

Caruana, R. & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on machine learning - icml '06*. doi:10.1145/1143844.1143865

Chai, T. & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3), 1247–1250. doi:10.5194/gmd-7-1247-2014. arXiv: arXiv:1011.1669v3

Chang, C.-c., Lin, C.-j., & Tieleman, T. (2008). LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *307*, 1–39. doi:10.1145/1961189.1961199. arXiv: 0-387-31073-8

Collins, L. M., Schafer, J. L., & Kam, C.-M. (2001). A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychological Methods*, *6*(4), 330–351. doi:10.1037/1082-989X.6.4.330

De Smith, M., Goodchild, M., & Longley, P. (2018). *Geospatial analysis : a comprehensive guide to principles, techniques and software tools* (Sixth).

Deodhar, M. & Ghosh, J. (2007). A framework for simultaneous co-clustering and learning from complex data. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, 250. doi:10.1145/1281192.1281222

Donders, A. R. T., van der Heijden, G. J., Stijnen, T., & Moons, K. G. (2006). Review: A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, *59*(10), 1087–1091. doi:10.1016/j.jclinepi.2006.01.014

Dubin, R. A. (1998). Spatial Autocorrelation: A Primer. *Journal of Housing Economics*, *7*(4), 304–327. doi:10.1006/jhec.1998.0236

ESRI. (2009). Using ArcGIS Geostatistical Analyst.

ESRI. (2016a). IDW. Retrieved January 9, 2019, from https://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/idw.htm

ESRI. (2016b). Kriging in Geostatistical Analyst. Retrieved January 10, 2019, from https://desktop.arcgis.com/en/arcmap/latest/extensions/geostatistical-analyst/kriging-in-geostatistical-analyst.htm

Feng, L., Nowak, G., O'Neill, T. J., & Welsh, A. H. (2014). CUTOFF: A spatio-temporal imputation method. *Journal of Hydrology*, *519*(PD), 3591–3605. doi:10.1016/j.jhydrol.2014.11.012

Filipe, Y. (2016). BBAC R repository. Retrieved January 15, 2019, from https://github.com/fnyanez/bbac/blob/master/bbac.R

Garciarena, U. & Santana, R. (2017). An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. *Expert Systems with Applications*, *89*, 52–65. doi:10.1016/j.eswa.2017.07.026

Gautier, L. (n.d.). R2py. Retrieved January 15, 2018, from https://rpy2.bitbucket.io/

George, T. (2005). A Scalable Collaborative Filtering Framework based on Co-clustering. *Data Mining 5th IEEE Conf.*

Gerber, F., De Jong, R., Schaepman, M. E., Schaepman-Strub, G., & Furrer, R. (2018). Predicting Missing Values in Spatio-Temporal Remote Sensing Data. *IEEE Transactions on Geoscience and Remote Sensing*, *56*(5), 2841–2853. doi:10.1109/TGRS.2017.2785240. arXiv: 1605.01038

Graham, J. W. (2009). Missing Data Analysis: Making It Work in the Real World. *Annual Review of Psychology*, *60*(1), 549–576. doi:10.1146/annurev.psych.58.110405.085530. arXiv: 1710.05289

Guo, D., Chen, J., MacEachren, A. M., & Liao, K. (2006). A Visualization System for Space-Time and Multivariate Patterns (VIS-STAMP). *IEEE Transactions on Visualization and Computer Graphics*. doi:10.1109/TVCG.2006.84. arXiv: arXiv:1011.1669v3

Haining, R. (2003). Data quality: implications for spatial data analysis. In *Spatial data analysis: Theory and practice* (Chap. 4, pp. 116–178).

Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association*. doi:10.1080/01621459.1972.10481214

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. doi:10.1007/b94608. arXiv: arXiv:1011.1669v3

Hibbert, J. D., Liese, A. D., Lawson, A., Porter, D. E., Puett, R. C., Standiford, D., ... Dabelea, D. (2009). Evaluating geographic imputation approaches for zip code level data: An application to a study of pediatric diabetes. *International Journal of Health Geographics*, *8*(1), 54. doi:10.1186/1476-072X-8-54

Honaker, J. & King, G. (2010). What to do about missing values in time-series cross-section data. *American Journal of Political Science*, *54*(2), 561–581. doi:10.1111/j.1540-5907.2010.00447.x

Jerez, J. M., Molina, I., García-Laencina, P. J., Alba, E., Ribelles, N., Martín, M., & Franco, L. (2010). Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial Intelligence in Medicine*, *50*(2), 105–115. doi:10.1016/j.artmed.2010.05.002. arXiv: 9701101 [cs]

Jordan, M. I. & Mitchell, T. (2015). Machine learning: Trends,perspectives, and prospects. *N. Engl. J. Med. J. Med. Internet Res. PLOS ONE Clin. Pharmacol. Ther*, *360*(96), 2153–2155. doi:10.1126/science.aac4520. arXiv: arXiv:1011.1669v3

Kanevski, M., Pozdnoukhov, A., & Timonin, V. (2009). *Machine learning for spatial environmental data : theory, applications and software*. doi:10.1201/9781439808085. arXiv: arXiv:1011.1669v3

KNMI. (2019). temperatuur - geinterpoleerde dagelijkse maximum temperatuur in Nederland. Retrieved February 10, 2019, from https://data.knmi.nl/datasets/Tx1/2?q=temperatuur%7B%5C&%7Ddtstart=1990-12-31T23:00Z%7B%5C&%7Ddtend=2016-12-31T22:59Z

Kornelsen, K. & Coulibaly, P. (2014). Comparison of Interpolation, Statistical, and Data-Driven Methods for Imputation of Missing Values in a Distributed Soil Moisture Dataset. *Journal of Hydrologic Engineering*, *19*(1), 26–43. doi:10.1061/(ASCE)HE.1943-5584.0000767

Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. In *Artificial intelligence in design '96*. doi:10.1007/978-94-009-0279-4_9

Kuncheva, L. I. (2004). On the optimality of Naïve Bayes. *Florida Artificial Intelligence Research Society Conference*. doi:10.1016/j.patrec.2005.12.001. arXiv: arXiv:1011.1669v3

Kwon, B. & Cho, H. (2010a). Scalable co-clustering algorithms. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*. doi:10.1007/978-3-642-13119-6_3. arXiv: arXiv:1011.1669v3

Kwon, B. & Cho, H. (2010b). Scalable co-clustering algorithms. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*. doi:10.1007/978-3-642-13119-6_3. arXiv: arXiv:1011.1669v3

Lam, N. S.-N. (1983). Spatial Interpolation Methods: A Review. *Cartography and Geographic Information Science*. doi:10.1559/152304083783914958

Leinweber, D. J. (2007). Stupid Data Miner Tricks. *The Journal of Investing*, *16*(1), 15–22. doi:10.3905/joi.2007.681820

Li, L. & Revesz, P. (2004). Interpolation methods for spatio-temporal geographic data. *Computers, Environment and Urban Systems*. doi:10.1016/S0198-9715(03)00018-8

Little, R. J. A., Rubin, & B., D. (2002). *Statistical Analysis with Missing Data* (2nd ed.). John Wiley & Sons, Incorporated. Retrieved from https://ebookcentral.proquest.com/lib/uunl/detail.action?docID=1775204

Lloyd, C. D. & Atkinson, P. M. (2010). geoENV VII fffdfffdfffd Geostatistics for Environmental Applications. In *Quantitative geology and geostatistics* (Vol. 16). Springer Dordrecht Heidelberg London New York. doi:10.1007/978-90-481-2322-3

Luengo, J., García, S., & Herrera, F. (2012). *On the choice of the best imputation methods for missing values considering three groups of classification methods*. doi:10.1007/s10115-011-0424-2

Luo, X., Zhou, M., Xia, Y., & Zhu, Q. (2014). An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*. doi:10.1109/TII.2014.2308433

Machine Learning Mastery. (2017). Classification And Regression Trees for Machine Learning. Retrieved January 8, 2019, from https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/

Marlin, B. (2004). Collaborative filtering: A machine learning perspective. *Master Thesis*, 137. doi:10.1109/MC.2009.263. arXiv: 3

Moran, A. P. A. P. (2016). Biometrika Trust Notes on Continuous Stochastic Phenomena Published by : Oxford University Press on behalf of Biometrika Trust Stable URL : http://www.jstor.org/stable/2332142

Accessed : 13-04-2016 05 : 12 UTC Your use of the JSTOR archive indicates your. *Biometrika*, *37*(1), 17–23. Retrieved from http://www.jstor.org/stable/2332142

Mueller, T. G., Pusuluri, N. B., Mathias, K. K., Cornelius, P. L., Barnhisel, R. I., & Shearer, S. A. (2004). Map Quality for Ordinary Kriging and Inverse Distance Weighted Interpolation. *Soil Science Society of America Journal*. doi:10.2136/sssaj2004.2042

Murphy, K. P. (2012a). *Machine Learning: A Probablistic Perspective*. doi:10.1007/SpringerReference_302149. arXiv: 0-387-31073-8

Murphy, K. P. (2012b). *Machine Learning: A Probablistic Perspective*. doi:10.1007/SpringerReference_302149. arXiv: 0-387-31073-8

Muthén, B., Kaplan, D., Hollis, M., Planning, U., & Angeles, L. O. S. (1987). ON STRUCTURAL EQUATION MODELING WITH DATA THAT ARE NOT MISSING COMPLETELY AT RANDOM. *52*(3).

National Weather Service's National Operational Hydrologic Remote Sensing Center. (2019). Snow Data Assimilation System (SNODAS) Data Products at NSIDC. Retrieved January 10, 2019, from https://nsidc.org/data/g02158

Oliveira, A. L. & Madeira, S. C. (2004). Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *1*(1), 24–45.

Oliver, M. A. & Webster, R. (1990). Kriging: A method of interpolation for geographical information systems. *International Journal of Geographical Information Systems*. doi:10.1080/02693799008941549

Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. *Proceedings of KDD cup and workshop*, 2–5. doi:10.1145/1557019.1557072. arXiv: 07/0008 [978-1-59593-834-3]

Pedregosa, F., Weiss, R., & Brucher, M. (2011). Scikit-learn: Machine Learning in Python. *12*, 2825–2830. doi:10.1007/s13398-014-0173-7.2. arXiv: 1201.0490

PySaL. (2018). Spatial Weights. Retrieved from https://pysal.readthedocs.io/en/latest/users/tutorials/weights.html

Python Documentation. (2018). time fffdfffdfffd Time access and conversions. Retrieved January 18, 2019, from https://docs.python.org/3/library/time.html

QGis. (2017). Spatial Analysis (Interpolation). Retrieved January 7, 2019, from https://docs.qgis.org/testing/en/docs/gentle%7B%5C_%7Dgis%7B%5C_%7Dintroduction/spatial%7B%5C_%7Danalysis%7B%5C_%7Dinterpolation.html

R Documentation. (2015). Inverse Distance Weighting Interpolation. Retrieved January 8, 2019, from https://www.rdocumentation.org/packages/phylin/versions/1.1.1/topics/idw

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. (2009). *Recommender Systems Handbook*. London: Springer Dordrecht Heidelberg London New York. doi:10.1007/978-0-387-85820-3. arXiv: arXiv:1011.1669v3

Rubin-Schwarz, A. (2017). Impyter documentation. Retrieved January 15, 2019, from https://impyte.readthedocs.io/en/latest/

Smith, M., Goodchild, M., & Longley, P. (2018). Geostatistical Interpolation Methods. In *Geospatial analysis 6th edition, 2018* (Sixth edit, Chap. 6.7, p. 618). Winchelsea Press.

Statistics Netherlands. (2018). Doodsoorzakenstatistiek. Retrieved February 10, 2019, from https://bronnen.zorggegevens.nl/Bron?naam=Doodsoorzakenstatistiek

Stojanova, D., Ceci, M., Appice, A., Malerba, D., & Džeroski, S. (2013). Dealing with spatial autocorrelation when learning predictive clustering trees. *Ecological Informatics*, *13*, 22–39. doi:10.1016/j.ecoinf.2012.10.006

Su, X., Khoshgoftaar, T. M., Zhu, X., & Greiner, R. (2008). Imputation-boosted collaborative filtering using machine learning classifiers. *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, (2), 949. doi:10.1145/1363686.1363903

Tanner, M. A. & Wong, W. H. (1987). The Calculation of Posterior Distributions by Data Augmentation. *Journal of the American Statistical Association*, *82*(398), 528–540.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, *16*(4), 437–450. doi:10.1016/S0169-2070(00)00065-0

Teegavarapu, R. S. V. (2014). Missing precipitation data estimation using optimal proximity metric-based imputation, nearest-neighbour classification and cluster-based interpolation methods. *Hydrological Sciences Journal*, *59*(11), 2009–2026. doi:10.1080/02626667.2013.862334

Tsou, M. H. (2015). Research challenges and opportunities in mapping social media and Big Data. *Cartography and Geographic Information Science*, *42*(1), S70–S74. doi:10.1080/15230406.2015.1059251. arXiv: arXiv:1011.1669v3

USA Today. (2018). Washington Climate. Retrieved January 18, 2019, from https://traveltips.usatoday.com/climate-washington-state-62313.html

Varma, S. & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatic*, *7*(91), 1–8. doi:10.1186/1471-2105-7-91

Wilhelmsson, M. (2002). Spatial models in real estate economics. *Housing, Theory and Society*, *19*(2), 92–101. doi:10.1080/140360902760385646

Willmott, C. & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, *30*(1), 79–82. doi:10.3354/cr00799

Willmott, C. J., Matsuura, K., & Robeson, S. M. (2009). Ambiguities inherent in sums-of-squares-based error statistics. *Atmospheric Environment*, *43*(3), 749–752. doi:10.1016/j.atmosenv.2008.10.005

Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., & Cunningham, S. J. (1999). Practical machine learning tools and techniques with Java implementations. doi:10.1.1.16.949

Wu, L., Jin, R., & Hoi, S. (2009). Learning Bregman distance functions and its application for semi-supervised clustering. *Advances in neural ... 24*(10), 1–9. Retrieved from http://machinelearning.wustl.edu/mlpapers/paper%7B%5C_%7Dfiles/NIPS2009%7B%5C_%7D0334.pdf

Wu, X. (2016). *Clustering-based approaches to the exploration of geo-referenced time series*. doi:10.3990/1.9789036541619

Wu, X., Zurita-Milla, R., & Kraak, M. J. (2015). Co-clustering geo-referenced time series: exploring spatio-temporal patterns in Dutch temperature data. *International Journal of Geographical Information Science*, *29*(4), 624–642. doi:10.1080/13658816.2014.994520

Wu, X., Zurita-Milla, R., & Kraak, M. J. (2016). A novel analysis of spring phenological patterns over Europe based on co-clustering. *Journal of Geophysical Research: Biogeosciences*, *121*(6), 1434–1448. doi:10.1002/2015JG003308

Yuan, M. (2017). 30 years of IJGIS: the changing landscape of geographical information science and the road ahead. *International Journal of Geographical Information Science*, *31*(3), 425–434. doi:10.1080/13658816.2016.1236928

Zhang, T. & Iyengar, V. S. (2002). Recommender Systems Using Linear Classifiers. *Journal of Machine Learning Research*, *2*(3), 313–334. doi:10.1162/153244302760200641. arXiv: 1008.4815v1

Zhang, Z., Yang, X., Li, H., Li, W., Yan, H., & Shi, F. (2017). Application of a novel hybrid method for spatiotemporal data imputation: A case study of the Minqin County groundwater level. *Journal of Hydrology*, *553*, 384–397. doi:10.1016/j.jhydrol.2017.07.053

Zhu, G. (2016). Nonnegative Matrix Factorization (NMF) with Heteroscedastic Uncertainties and Missing data. arXiv: 1612.06037. Retrieved from http://arxiv.org/abs/1612.06037