Essays on Business Analytics

by

Mostafa Rezaei

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Operations and Information Systems

Faculty of Business

University of Alberta

Abstract

This dissertation consists of three separate essays on business analytics. Abstract of the three essays are as follows:

Essay 1: Peer-production projects are increasingly attracting the attention of Human-Computer Interactions (HCI) scholars. Such complex socio-technical systems could be viewed as Complex Adaptive Systems (CAS). The complexity of such projects presents a challenge to researchers trying to understand the dynamics of co-production processes. Visualization makes relevant processes visible that would otherwise be difficult to interpret, and thus our objective is to develop an information visualization tool that would surface important pattern in peer-production. In this paper we introduce an interactive visualization simulation—WikiAttractors—that is inspired by techniques from the area of CAS to visualize the process by which a knowledge-based product evolves over time. Using Wikipedia as an example, we trace the evolution of articles from their inception until they are fully developed (i.e., Featured Articles). We show how WikiAttractors is able to identify both local (vandalism, negotiation) and global patterns (rate of convergence) of co-production.

Essay 2: Although data mining problems require a flat mining table as input, in many real-world applications analysts are interested in finding patterns in a relational database. To this end, new methods and software have been recently developed that automatically add attributes (or features) to a target table of a relational database which summarize information from all other tables. When attributes are automatically constructed by these methods, selecting the important attributes is particularly difficult, because a large number of the at-

tributes are highly correlated. In this setting, attribute selection techniques such as the Least Absolute Shrinkage and Selection Operator (lasso), elastic net, and other machine learning methods tend to under-perform. In this paper, we introduce a novel attribute selection procedure, where after an initial screening step, we cluster the attributes into different groups and apply sparse modelling techniques (e.g. Group lasso) to select both the true attributes groups and the true attributes. The procedure is particularly suited to high dimensional data sets where the attributes are highly correlated. We test our procedure on several simulated data sets and a real-world data set from a marketing database. The results show that our proposed procedure obtains a higher predictive performance while selecting a much smaller set of attributes when compared to other state-of-the-art methods.

Essay 3: Forecasting Emergency Medical Services (EMS) call volumes is critical for resource allocation and planning. The value of call volume forecasts increases with forecast accuracy and with spatial resolution; however as the spatial resolution is increased, sample sizes for each spatial unit decrease, and hence accuracy decreases. Thus, there is a trade-off between forecast accuracy and spatial resolution. We study this trade-off in this paper, using 5 years of data from 3 cities in Alberta. We compare various exponential smoothing methods to capture weekly seasonality, differences, and correlations across neighbourhoods. Our findings suggest that including a seasonal component in the forecasting method improves the accuracy while having a trend component, transforming the call volumes, or accounting for autocorrelations among errors have little pay off. Furthermore, a top-down approach, where forecasts are made on a lower resolution than the resolution of interest and then divided, performs as well as a bottom-up approach, where forecasts are made at a higher resolution and then aggregated.

Preface

Chapter 2 was co-authored by Dr. Ofer Arazy and Dr. Michele Samorani, and was submitted to the Association for Computing Machinery (ACM) Conference on Human Factors in Computing Systems (CHI). The visualization tool provided the starting point for quantification of the *exploration* variable, as part of the analysis in a paper titled "The Evolutionary Trajectories of Peer-Produced Artifacts", co-authored by Dr. Ofer Arazy, Dr. Aron Lindberg and Dr. Michele Samorani, and currently under review for publication.

Chapter 3 was co-authored by Dr. Michele Samorani and Dr. Ivor Cribben and has been published as:

Rezaei, M., Cribben, I., & Samorani, M. (2018). A clustering-based feature selection method for automatically generated relational attributes. Annals of Operations Research, `https://doi.org/10.1007/s10479-018-2830-2`.

Chapter 4 is part of a working paper co-authored with Dr. Armann Ingolfsson. Ethics application and data request were submitted and approved by the Research Ethics Board of the University of Alberta.

Acknowledgements

My sincere gratitude to my supervisors, Dr. Armann Ingolfsson and Dr. Michele Samorani, for their motivation, patience, and continuous support of my PhD study and research. I would like to thank Dr. Ivor Cribben, the other member of the supervisory committee. I would also like to thank other co-authors that I had the honour to work with and learn from: Dr. Ofer Arazy, Dr. Ivor Cribben, and Dr. Aron Lindberg. Many thanks to Dr. David Deephouse, Business PhD Program Chair, Dr. Karim Jamal, Department of Accounting, Operations and Information Systems Chair, Debbie Giesbrecht, and Helen Wu, PhD Program Office staff, and Karmeni Govender and Beth Boison, Department of Accounting, Operations and Information Systems staff, for their support. Last but not the least, I would like to thank my wife and my parents for their love and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this dissertation we present three essays related to business analytics. The use of analytics in business can be traced back to the 1960s when practitioners started using computers and data to build decision support systems (DSS) for better production planning, investment portfolio management, and transportation routing (Davenport and Harris, 2017). The amount of data and computation power that firms possess today is, however, incomparable to what was available at that time and we now see many companies that use analytics strategically to differentiating themselves from their competitors. Firms like Amazon, Walmart, and Netflix extensively use quantitative methods to optimize supply chains, improve inventory management, and achieve higher customer satisfaction. In addition, we have vendors like IBM providing analytical services to a wide range of sectors such as healthcare and finance.

Analytics can be descriptive, predictive, or prescriptive (Davenport and Harris, 2017). Descriptive analytics refers to exploring the data, using visualization, data summaries, and models, for finding patterns within the data. Predictive analytics refers to building models that exploit patterns in the data in order to forecast the future. Prescriptive analytics refers to using models to optimize or improve decision making. Most often, we need to conduct all three sequentially and iteratively in order to solve a problem. Furthermore, the quantitative methods we use depend on the complexity of the problem and, the complexity

of the data. Problems can be categorized as either simple, complicated, or complex (Nason, 2017). Simple problems are those that have a well-known procedural solution. Complicated problems cannot be solved using a single procedure but can, nevertheless, be broken down to simple subproblems, which can be solved individually. Complex problems are those that, due to the many interrelated factors, cannot be broken down and solved in a reductionist manner.

In Chapter 2 we propose a new exploratory data analysis tool to visualize the process by which a knowledge-based product evolves over time. We focus on the dynamics among the editors as they make revisions on Wikipedia articles. The textual, temporal, and social aspects of the data makes the analysis a complex problem. We therefore focus mostly on descriptive analytics as a first attempt to understand the data. We show how our new proposed visualization reveals patterns of behaviour that would be hard to see otherwise: acts of vandalism, edit-wars, exploration of the knowledge space, non-linearity and complexity of an article's evolution. We believe that our proposed method will be useful in other collaborative settings, especially when complemented with other statistical methods.

In Chapter 3 we propose a feature selection procedure for high-dimensional, highly-correlated data. Feature selection is a critical step in predictive analytics since the performance of a predictive model is significantly affected by the quality of its features. In many cases, automatically generated features have some distinctive characteristics: They are high-dimensional, can be assigned a complexity score, and they are multicollinear. In addition, there are usually many features with low predictive power. Each of these three characteristics adds a layer of complexity to the problem causing off-the-shelf feature selection methods to perform poorly. Our proposed method acts as a data mining pipeline where we sequentially deal with each complication. We show how our method can be used in a real-world setting for finding the most informative variables that predict the probability of a product return.

In Chapter 4 we study the trade-off between forecast accuracy and spatiotemporal resolution for emergency medical services (EMS) call volumes. Even though it is not easy to obtain

accurate forecasts of call volumes, in our categorization of problem complexity, it is considered simple since well-known solutions exist. We compare the performance of many variants of exponential smoothing methods that differ in their spatiotemporal resolutions, their state-vector components, and in whether they are univariate or multivariate, among other factors. Since we are interested in using the forecasting model as a prescriptive analytics tool for better staffing of ambulance stations, we compare the performance of all the models at a *station catchment area* and *8-hour time interval* spatial and temporal level of resolution, respectively. We show that in many cases simple models outperform more complicated ones.

After the three substantive chapters, Chapter 5 concludes and outlines promising directions for future research that are related to Chapters 2, 3, and 4.

Chapter 2

WikiAttractors: Visualizing Wikipedia as a complex adaptive system

2.1 Introduction

Large scale collaborative efforts such as Wikipedia and open source software development projects represent a community-based model for the production of knowledge-based goods. In these peer-production projects, contributions of knowledge are made by volunteers, who self-organize to manage the production process. The sheer scale and complexity of peer-production systems present a serious barrier to both quantitative and qualitative methods for identifying relevant patterns of behavior. Information visualization techniques make relevant processes visible that would otherwise be difficult to interpret. The setting for this analysis is Wikipedia, one of the most notable examples of peer production. Past research in the areas of Human-Computer Interaction (HCI) and Computer-Supported Cooperative Work (CSCW) has explored various visualizations for capturing the dynamics of Wikipedia's co-authoring process (Viégas et al., 2004; Suh et al., 2008; Arazy et al., 2015).

The overarching objective of this project is to investigate the dynamics underlying the co-creation of knowledge-based products. A particularly relevant theoretical lens for understanding such sociotechnical systems is the theory of complexity, and specifically: Complex Adaptive Systems (CAS). CAS theory provides a multilevel scheme for linking individual actions to system-level regularities and describes how the trajectory a system takes as it transitions between various states (Miller and Page, 2009).

In this chapter,we present a visualization tool, WikiAttractors, that is inspired by the theory of complexity and aims to visualize the co-production dynamics in Wikipedia, namely the process by which a knowledge-based product (i.e., article) evolves over time. Visual methods used in CAS are particularly useful for understanding how a system transitions between stable and unstable states, often revolving around "attractors" ("center of gravity" around which the system revolves). For our purposes, we define the stable state of a Wikipedia article as a fully developed encyclopedic entry (i.e., one that has received a "Featured Article" status). Our interactive simulation employs the contents of an article at its various revisions (accessed through Wikipedia's API) to track the state of the article from its inception until the stable state of a Featured Article. By showing how an article transitions towards and away from the stable state (until converging), we are able to identify key behavioral patterns that attracted the interest of peer-production researchers. WikiAttractors makes apparent both local—vandalism and negotiations (i.e., conflicts of opinions)—and global patterns (trajectory of an article's evolution). Our discussion highlights the ways in which our tool extends prior visualizations in this area, and considers how newly-identified patterns could inform research in this area.

2.2 Prior studies visualizing Wikipedia process

The area of online collaboration has attracted the attention of HCI researchers and a relatively many number of information visualizations have been developed for the purpose of

illustrating co-creation processes. Below, we briefly review some of these visualizations, focusing on those developed for Wikipedia. Many tools have attempted to capture the overall structure of Wikipedia and its editor base (Massa et al., 2012; Otjacques et al., 2009). Other tools were developed to visualize the co-production process (Suh et al. (2008); Nunes et al. (2008); Biuk-Aghai and Chan (2012)). Some tools analyze the contents of edits (or revisions), often abstracting edits into categories and visualizing temporal sequences of these categories (Wattenberg et al. (2007); Arazy et al. (2015)). However, we focus our attention on the product—the entire contents of the Wikipedia article—and its evolution over time.

Relevant to our investigation, Viegas and colleagues (Viégas et al. (2004); Viegas et al. (2007)) developed the HistoryFlow tool to track the evolution of the contents of an article, and used the tool to identify local patterns in the article's evolution. Namely, the tool was useful to detect large-scale vandalism, where the entire content of the article is deleted or when large text portions are added. In addition, this visualization was also useful in capturing simple negotiations (i.e., edit wars where the article goes back-and-forth between two states). Kittur and Kraut (2008) developed a metric to record the overall stability of an article in terms of changes to its contents, but did not capture the directionality of these changes.

2.3 Our proposed visualization: WikiAttractors

WikiAttractors (`https://mostafarezaei.shinyapps.io/wikiattractors`) was inspired by the theory of non-linear dynamic systems, which models how a system transitions between states. Unlike existing visualizations of Wikipedia that assume that the co-production process always proceeds in one direction, our visualization tracks the state of the co-produced artifact product (i.e., the article) as it moves in various directions—often revolving around attractors—until converging at the stable state, i.e., the fully-developed article (Featured Article). The attractors are points in the state space to which the article often returns: whether the empty article (reached when the entire article is deleted by vandals) or points

representing a particular viewpoint (such that the transition between regions around these points corresponds to a negotiation of opinions).

Our proposed visualization procedure takes as input the sequence s^0, s^1, s^2,..., s^F of representations of the contents of each version (or state) s^i of the article, starting from the intial version s^0 and ending with a final version s^F. The transition $s^{i-1} \rightarrow s^i$ from state s^{i-1} to state s^i represents a revision made by a particular contributor at a particular time.

We visualize the state of an article's revision on a reduced-dimension (2D) state chart, where the interactive simulation traverses the article's revisions and presents its position in this space. The position of each state s^i is determined by x_i, the distance between the current state s^i and the initial state s^0, and y_i, the distance between the current state s^i and the final state s^F. Each transition $s^{i-1} \rightarrow s^i$ is displayed as a line segment.

We adopt the following conventions and definitions:

1. The contents of an article, s^i, are represented as a binary 'bag-of-words': a vector indicating the absence (0) or presence (1) of each word in a dictionary that includes all words used in all revisions;

2. the distance between state s^i and s^j is the Hamming distance, which counts the number of words that appear in s_i but not in s_j or that appear in s_j but not in s_i;

3. the initial state s^0 is the empty article (0, 0,..., 0);

4. the final state s^F is the Feature Article version.

It follows that x_i is the number of words in state s^i, whereas y_i is $m_i + e_i$, where m_i is the number of words appearing in s^F but not in s^i (the *missing* words), and e_i is the number of words appearing in s^i but not in s^F (the *extra* words).

Also, from these definitions it follows that the final state s^F is mapped to point $(x_F, y_F) = (l, 0)$. The reason is that the final article has length $x^F = l$, t does not have any missing words ($m_F = 0$), and it does not contain any extra words ($e_F = 0$), which results in $y_F = m_F + e_F = 0$.

2.3.1 Example

We use a simple example to illustrate the construction of the transition diagram and its interpretation. The sequence of states in Table 2.1 results in the chart of Figure 2.1:

Table 2.1: States of the example article

State	Text
s^0	(blank)
s^1	Smoking is the main cause of cancer.
s^2	Smoking is the main cause of cancer. Children who smoke die early.
s^3	Smoking is the main cause of cancer.
s^4	Smoking is the main cause of cancer. Children who smoke are not healthy and die early.
s^5	Smoking is the main cause of cancer.
s^6	Smoking is the main cause of cancer. Studies show a relationship between smoking and lung cancer.
s^7	Studies indicate that there is a correlation between smoking cigarettes and lung cancer.
s^8	Studies indicate that there is a correlation between smoking cigarettes and lung cancer. My aunt died of it.
s^9	Studies indicate that there is a correlation between smoking cigarettes and lung cancer.
s^F	Studies indicate that there is a high correlation between smoking cigarettes from a young age and lung cancer.

From the definitions above, it is straightforward to interpret the directions along which the article moves, as follows. In a **North** (N) movement, relevant parts are removed and irrelevant ones are added. In a **North-East** (NE) movement, irrelevant parts are added. In an **East** (E) movement, some relevant and some irrelevant parts are added. In a **South-East** (SE) movement, relevant parts are added. In a **South** (S) movement, irrelevant parts are removed and relevant ones are added. In a **South-West** (SW) movement (e.g., $s^4 \rightarrow s^5$), irrelevant parts are removed. In a **West** (W) movement, some relevant and some irrelevant parts are removed. In a **North-West (NW)** movement, relevant parts are removed.

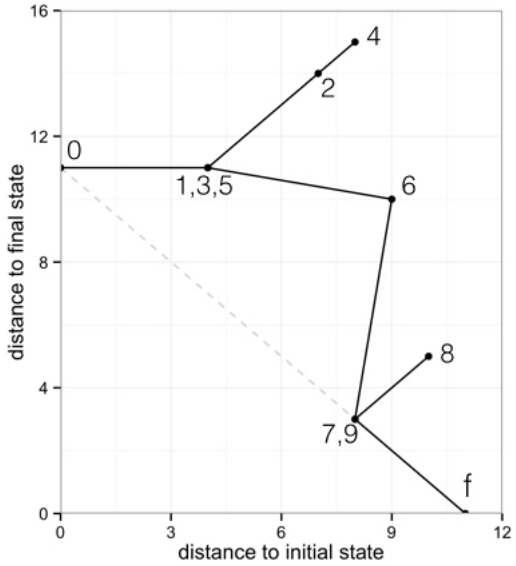

Figure 2.1: Trajectory for the example article.

2.4 Quantitive analysis of the visualization

Using quantitative methods, we analyze the state transition diagrams to detect attractors: local centers of gravity that attract the system state, as well as calculate metrics capturing the overall shape of the system's trajectory. We interpret the observed patterns as various patterns of Wikipedia's co-production. In order to help interpret the emerging patterns, we tracked the Wikipedia roles of contributors, coloring the diagrams based on the role of the contributor making the last revision. We applied the following scheme: unregistered contributors = green; registered = orange; administrators = red; and software robots (bots) = blue.

2.5 Detecting Vandalism

One phenomenon that is immediately observed in the transition diagrams is vandalism, where an editor deletes a large portion of the article or adds an irrelevant section. This behaviour

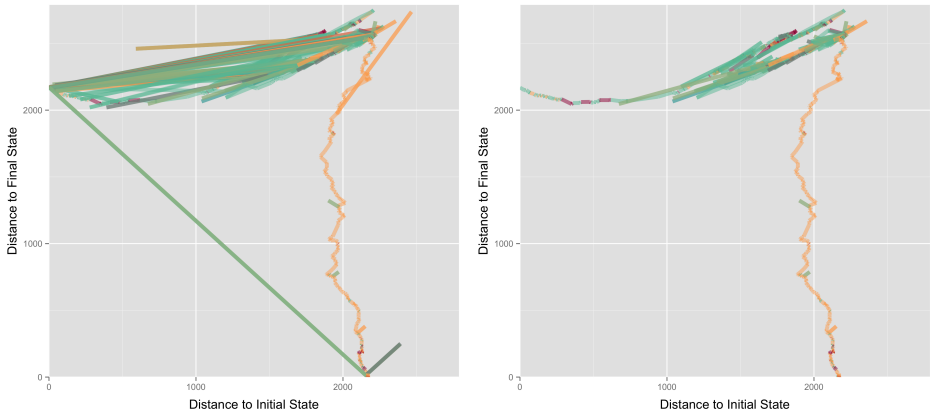

Figure 2.2: Article *Autism* before and after detection and removal of vandalism instances.

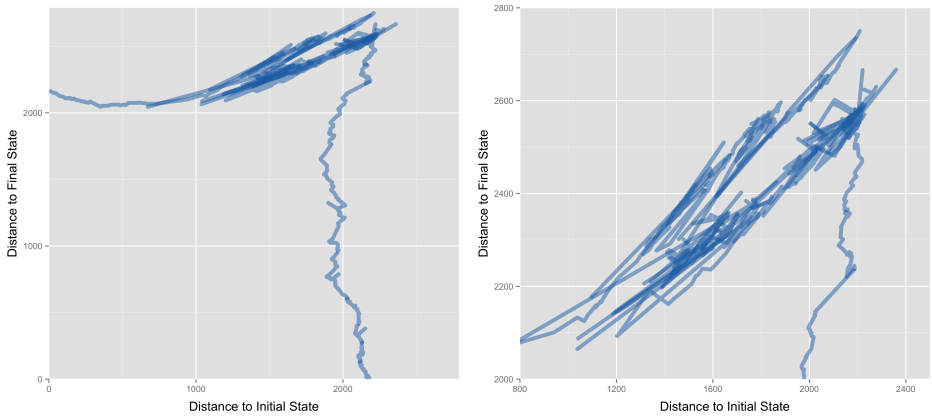

Figure 2.3: Patterns of negotiation are shown as dark blue (Autism article)

is represented as a deviation from the article's primary trajectory (e.g., when deleting all contents—a straight line from the current state the article to the initial state). In our example, the addition of the irrelevant text "My aunt died of it" represents vandalism, and is seen in Figure 2.1. To identify vandalism, we first detect all reversals to the original state and then remove single points that represent substantial deviations from the article's primary trajectory (see more details on smoothing in the sections below).

For example, an illustration of the Wikipedia article on Autism is presented in Figure 2.2 (inception date = December 31, 2001; Featured Article status achieved on September 1, 2007). The left panel shows the full trajectory of the article, whereas the right panel shows the article after applying our vandalism detection algorithm.

2.6 Detecting negotiations

"Negotation," or "conflict of opinion," is a legitimate conflict between alternative views of the topic. These can take the form of a simple "edit war" (two contributors adding and reverting content in turn) or a more complex tug of war between two camps pulling in opposite directions (but not necessarily reverting the article to the exact same point). Our algorithm for detecting such negotiations is:

A conflict of opinion is identified by two sets of states $A = \{a_1, ..., a_m\}$ and $B = \{b_1, ..., b_n\}$ such that:

1. $\max_{i,j} \text{dist}(a_i, a_j) < \min_{i,j} \text{dist}(a_i, b_j)$
2. $\max_{i,j} \text{dist}(b_i, b_j) < \min_{i,j} \text{dist}(a_i, b_j)$
3. $m \geq q$ and $n \geq q$
4. Between visits to state a_i and state a_{i+k}, there must be at least one visit to a state in B, and vice versa.
5. Any trajectory from a state in set to a state in the other set can traverse at most p states that are outside A and B,

where A and B are ordered according to the sequence in which the states are visited. Based on the above definition one can propose an algorithm with three parameters, q, p, and k for finding conflict of opinion. Figure 2.3 shows conflict of opinions in the Autism article (complete article trajectory on the left pane and a zoom-in on the right pane of 2.3). Our algorithm can identify such patterns in the article's trajectory.

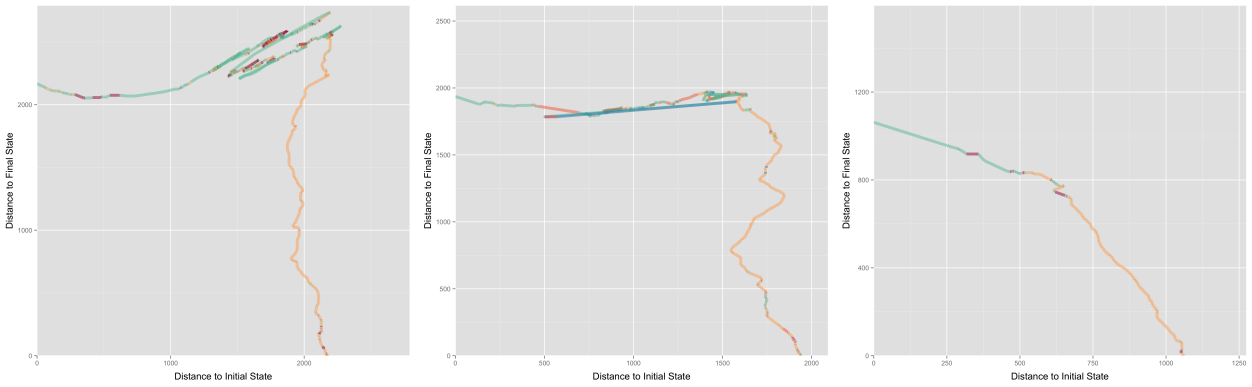

Figure 2.4: Global trajectory of three different Featured articles: *Autism*, *John Calvin*, and, *Irish Hunger Strike*

2.7 Measuring the Article's Global Trajectory

In order to make the article's global trajectory clearer, it is essential that local features are removed. We do this by smoothing, using a moving median with a window size of k:

$$(x_s^t, y_s^t) = (\text{median}(x^{t-\frac{k}{2}}, ..., x^t, ..., x^{t-\frac{k}{2}}), \text{median}(y^{t-\frac{k}{2}}, ..., y^t, ..., y^{t-\frac{k}{2}}))$$

The window size k controls the amount of smoothing. The left pane of Figure 2.4 shows the smoothed trajectory for the Autism article using a windows size of $k = 10$.

Different articles yield different global patterns, as evident in Figure 2.4 (showing the smoothed trajectories for articles: Autism; John Calvin; and 1981 Irish hunger strike). One of the most noticeable features of these trajectories is the rate at which the articles converge to the final state (i.e., Featured Article). To quantify this rate we use the following metric:

$$\frac{\text{actual distance travelled}}{\text{minimum required travelling distance}}$$

The values for this metric for the left, middle, and right trajectories in Figure 2.4 are 4.2, 3.7, and 1.1, respectively.

Smoothing can also be used to further remove vandalism by computing the following

measure:

$$M = |x_s^t - x^t| + |y_s^t - y^t|,$$

and removing a certain percentage (4% in our case) of states with the highest M.

2.8 Discussion and conclusion

In this paper we introduced a new visualization method—WikiAttractors—that is inspired by the theory of Complex Adaptive Systems. The key principles of our approach are: (a) representing the state of the co-produced artifact in a state-space; and (b) determining the position of a particular version of the artifact in the state space based on its distance to both its original and final states. Our operationalization employed simple linguistic methods for representing the state-space and defined the final state of the Wikipedia article as the version when it became a Featured Article. Of course, alternative realizations of our method are possible.

Building on prior work in the area, WikiAttractors is designed to identify behavioral patterns in the co-production process that are of interest to scholars. The most notable novelty of our approach is that it tracks the trajectory of the system as it traverses the state space, often revolving around attractors (in contrast to prior visualizations which represented the progressions as a one-directional process; e.g., (Kittur and Kraut, 2008; Arazy et al., 2015)). Attractors could represent stable (i.e., the end state) or unstable states (vandalism or conflict of opinions). For example, while prior visualizations were able to detect vandalism, they typically only captured simple delete of the entire page (and the following revert) (Viégas et al., 2004); in contrast, WikiAttractors is also able to detect the insertion of an irrelevant sentence. Similarly, prior tools were only able to capture relatively simple edit wars (Viégas et al., 2004), whereas the proposed method captures conflicts of opinions, often between two coalitions representing alternative viewpoints. Finally, an important contribution of this work

is the introduction of novel methods for representing—both visually and using quantitative metrics—the overall trajectory of an article's evolution and the rate at which it converges. Recording contributors' formal role in Wikipedia (in terms of access privileges) sheds new light on the observed patterns, illustrating: (I) the involvement of unregistered users in vandalism; (II) the responsibility of administrators for negotiations, often determining the direction that an article takes; and (III) the role of regular users in driving an article—once a direction has been set—to its final state.

Notwithstanding the novelty of our visualizations, it suffers from some limitations which we plan to address in the near future. First, we intend to explore the extent to which the observed patterns are sensitive to alternative operationalizations (for example, defining an end state other than that of a Featured Article). Second, we plan to refine our algorithms for detecting vandalism and negotiations. Third, we are interested in performing a large-scale analysis of the trajectories of many articles, and to identify prototypical patterns of convergence. Lastly, inspired by complexity theory we seek to explore whether the trajectories of Wikipedia articles present other characteristic traits of non-linear dynamic systems (e.g., similarity across scales).

Chapter 3

Feature selection for high-dimensional highly-correlated data: A clustering-based approach

3.1 Introduction

Classification is an important data mining problem whose objective is to explain how a set of independent variables (or attributes) determine the value of a categorical dependent variable. Classification techniques analyze a flat mining table (also called a 'data set') that consists of one row per observation and one column for each independent variable; the dependent variable Y also consists of one column.

One of the critical steps for classification is the construction of the data set. In a typical business environment, this step is performed manually beginning with a relational database, which is a set of tables that are related to one another. Typically, analysts use their domain knowledge and intuition to summarize the information from the database into the attributes that make up the flat mining table. Choosing the right attributes determines whether a satisfactory explanatory rule is found. For example, consider a database of customers and their

purchases and suppose that the most important attribute of a customer with a large lifetime value is the amount of dollars spent by the customer in cheap electronics. If the analyst does not include any attribute that carries information on past purchases in electronics, then the classification most likely will lead to unsatisfactory predictive performance and explanatory power, no matter which technique is used.

To automate the data construction process and to facilitate knowledge discovery, a number of attribute generation (AG) techniques have been proposed in the last two decades (Knobbe et al., 2001; Perlich and Provost, 2006; Samorani et al., 2011). AG techniques take a relational database as input, and "expand" a given 'Target table' by adding 'relational attributes' that summarize information from the other tables. This effort has lead to the development of a software system, Dataconda (Samorani, 2015), freely available for download from `www.dataconda.net`, which has the potential to generate thousands of attributes in a short time, including, for example, "the total number of client's past purchases in electronics" or "the average age of the clients who purchased the same products as the current client". We assume that important attributes in the classification model are among the thousands of generated attributes.

However, despite the breakthrough in the automatic generation of relational attributes, no progress has been made to design an effective attribute selection procedure that can effectively be used on these data sets. To this end, this paper attempts to fill this research gap by designing a method that is particularly effective on this type of data.

In the remainder of the introduction, we introduce a "running" example that is used throughout the paper, we briefly illustrate examples of automatically generated relational attributes and discuss their characteristics, and we argue that existing attribute selection methods are likely to perform poorly on relational attributes. In Section 3.2, we discuss related work. In Section 3.3, we illustrate our methodology and its variants. Then, in Section 3.4, we compare the performance of our method to other methods on both simulated and real-world data. Finally, we discuss limitations and future research directions.

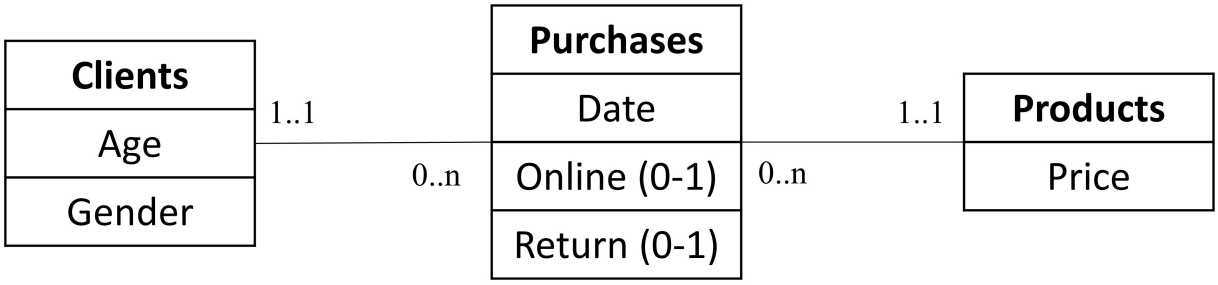

Figure 3.1: The Entity-Relationship diagram of the Returns database

3.1.1 Characteristics of Automatically Generated Relational Attributes

Let us illustrate some examples of automatically generated relational attributes by considering the running example of Figure 3.1, which is used throughout the paper. This database consists of three tables (Clients, Purchases, and Products) and contains information on products that have been purchased by clients of a fictitious retail store. More precisely, clients may make many purchases ("0...n", using the Entity-Relationship notation (Batini et al., 1989)). Each purchase is characterized by an attribute Date, an attribute Online (a binary indicator of whether the purchase was made online) and an attribute Return (a binary indicator of whether the purchased product was eventually returned back to the store by the client). Each purchase involves exactly one (1...1) product, which is characterized by a certain price, and one (1...1) client, who is characterized by an age and a gender.

The goal of this problem is to classify purchases by the attribute Return in order to find the important attributes (or predictors) of product returns. AG techniques generate new attributes which summarize information from the other tables, and add them to the target table Purchases. For example, the relational attributes generated by Dataconda include the following:

A1 The price of the product purchased;

A2 The total amount of money spent by the same client prior to the current purchase;

A3 The total amount of money spent by the same client prior to the current purchase on online purchases;

A4 The average age of the female clients who returned the same product prior to the current purchase.

It is immediately clear that these attributes could also be generated manually by simply joining the tables of the database and optionally aggregating one attribute. However, AG techniques do it automatically, thereby saving the analysts time and relieving them from the burden of formulating hypotheses about the possible drivers of product returns.

Automatically generated relational attributes have the following characteristics: (1) high dimensionality, (2) the presence of the complexity score of each attribute, and (3) multicollinearity, which make them difficult to model. We now describe these characteristics separately.

High dimensionality

The high dimensionality is caused by the ability of Dataconda to generate an arbitrarily large number of attributes. While this allows us to test a large number of hypotheses that may go untested in a manual analysis, it makes the problem challenging from a computational perspective. Also, an excessively large number of attributes being selected may lead to overfitting.

Complexity Score

Intuitively, the complexity score of an attribute measures how difficult it is to interpret the meaning of the attribute. For example, it can be argued that A1 is easier to interpret than A2, which is easier to interpret than A3, which is easier to interpret than A4. Dataconda measures the complexity score of each attribute as the number of tables joined to generate

the attribute plus 0.1 for each refinement (i.e., "where" clause) used to generate the attribute. For example, A1 has complexity 2, because it is generated by joining the table Purchases with the table Products (two tables); A2 has complexity 3.1, because it joins Purchases P1 to Clients C to Purchases P2 (three tables) and it uses the refinement *"where P1.Date > P2.Date"*; A3 has complexity 3.2 because it has one refinement more than A2 (*"where P2.Online = 1"*); finally, A4 has complexity 4.2. For more details on attribute generation, see Samorani (2015).

Note that the complexity score of each attribute is available without the need to look at the values of the attribute. According to the Occam's razor principle (Simon, 1979), an easier explanation should be preferred to a more complex one. Thus, our method prioritizes the selection of less complex attributes.

Multicollinearity

The multicollinearity is caused by the presence of generated attributes which are, by construction, highly correlated within groups. For example, A2 and A3 are likely to be correlated (assuming that clients randomly make purchases online or not). Assuming that most products cost less than $5,000, A2 would also be highly correlated with "The total amount of money spent by the same client prior to the current purchase among products costing less than $5,000".

For each group of highly correlated variables, it is desirable to select only a small number of them. To this end, our method uses clustering to detect the groups of correlated attributes (similar to Buhlmann et al. (2013)), in order to subsequently select only a small number of them from each group.

3.2 A Brief Review of Existing Methods

Our work is related to different research streams in machine learning, statistics, and marketing. Each of them also have their own nomenclature. Hence, in the paper, we do not distinguish between 'attribute', 'feature', 'predictor', and 'independent variable'.

First, our work is related to automatic attribute generation in the machine learning literature. Several attribute generation (AG) techniques have been proposed in the last two decades (Knobbe et al., 2001; Popescul and Ungar, 2003; Perlich and Provost, 2006; Samorani et al., 2011). While these developed methods to generate relational attributes, they do not develop any specific method to select the best predictors among them. Knobbe et al. (2001) used decision trees, Popescul and Ungar (2003) used logistic regression, Perlich and Provost (2006) used logistic regression and decision trees, and Samorani et al. (2011) used 10 classifiers implemented in Weka (Hall et al., 2009).

Second, our work is related to the literature in statistical methods for high-dimensional data, which we show is inferior to our proposed method. These methods are based on regression techniques, which can be defined by the following linear model:

$$Y = X\beta + \epsilon, \tag{3.1}$$

with Y a $n \times 1$ univariate response vector, X an $n \times p$ design matrix, β a $p \times 1$ true coefficient vector and ϵ an $n \times 1$ error vector. We can assume that the columns of X and the vector Y have been centred, so that we can omit the intercept term. Our focus is to perform feature selection (or variable screening) which has an important role in many fields (see Fan and LV (2010) for an overview of variable selection and variable screening). In order words, we would like to recover the set $S = \{j : \beta_j \neq 0, j = 1, ..., p\}$ with high probability using a particular procedure. In the common case where $p >> n$, Ordinary Least Squares (OLS) cannot be used to estimate β in eq.3.1 due to the issue of non-identifiability. Many procedures have been

introduced to estimate the set S. The Least Absolute shrinkage and selection operator (lasso, Expression 3.2) developed by Tibshirani (1996) uses a l_1 regularization of the coefficient vector as a way of feature selection.

$$\min_{\beta} \frac{1}{2n}\|Y - X\beta\|_2^2 + \lambda\|\beta\|_1 \tag{3.2}$$

where λ is called "regularization parameter". For the case of Logistic Regression, where $Y \in \{0, 1\}$ is a $n \times 1$ binary response vector, the corresponding lasso expression is as follows:

$$\min_{\beta} -\frac{1}{n}\sum_{i=1}^{n}[y_i.(x_i^T\beta) - \log(1 + e^{(x_i^T\beta)})] + \lambda\|\beta\|_1 \tag{3.3}$$

where x_i^T is the i^{th} row of the design matrix X. Lasso and its variants (Zou, 2006) have been very successful in applied settings. When the features are highly correlated with one another or where there is near linear dependence between a subset of features, lasso tends to select one feature from the highly correlated or near linear dependent group. Our method tends to achieve the same behavior, but we show that our method is better than Lasso at selecting the best feature of each group. The elastic net (Zou and Hastie, 2005), OSCAR (Bondell and Reich, 2008) and the cluster lasso (She, 2008) were introduced to allow for the selection of features that are highly correlated with one another. In addition to penalizing the l_1 norm vector of the coefficients, the elastic net also penalizes the squared l_2 norm of the vector (Expression 3.4).

$$\min_{\beta} \frac{1}{2n}\|Y - X\beta\|_2^2 + \lambda\|\beta\|_1 + (1 - \lambda)\|\beta\|_2^2 \tag{3.4}$$

Similarly for logistic regression:

$$\min_{\beta} -\frac{1}{n}\sum_{i=1}^{n}[y_i.(x_i^T\beta) - \log(1 + e^{(x_i^T\beta)})] + \lambda\|\beta\|_1 + (1 - \lambda)\|\beta\|_2^2 \tag{3.5}$$

The previous three methods do not taken into consideration the inherent correlation

structure between the features and they underperform when the features are close to perfect linear dependence. Huang et al. (2011) developed a sparse Laplacian shrinkage estimator which chooses features from a group of highly correlated features under certain regularity conditions. The restriction of this method is that the highly correlated features all have similar predictive performance. Buhlmann et al. (2013) do not require the features to have similar predictive performance and instead propose a two-step procedure. The method first clusters the variables and then uses sparse estimation techniques such as the group lasso (Yuan and Lin, 2007) Expression 3.6) to select the groups or clusters of features.

$$\min_{\beta} \frac{1}{2n}\|Y - X\beta\|_2^2 + \lambda \sum_{l=1}^{m} \sqrt{\rho_l}\|\beta^{(l)}\|_2 \tag{3.6}$$

where m is the number of groups and ρ_l is the number of variables in group l, and $\beta^{(l)}$ is the coefficient vector of the variables in group l. Similarly, for logistic regression:

$$\min_{\beta} -\frac{1}{n}\sum_{i=1}^{n}[y_i.(x_i^T\beta) - \log(1 + e^{(x_i^T\beta)})] + \lambda \sum_{l=1}^{m} \sqrt{\rho_l}\|\beta^{(l)}\|_2 \tag{3.7}$$

Our work is also related to previous work on clustering variables and then carrying out model fitting. Principal component regression (PCR), where principal component analysis (PCA) on the features is followed by regression was proposed by Kendall (1957). Hastie et al. (2001) proposed tree-harvesting where supervised learning methods are used on clusters of features that have been formed using hierarchical clustering. A method that jointly performs clustering and model fitting was developed by Dettling and Bühlmann (2004) and Bondell and Reich (2008). Buhlmann et al. (2013) carry on this line of work by proposing a method that first uses canonical correlation to cluster the features and then uses the group lasso to select the groups or clusters of features. All of the methods mentioned above can be used in the case where the response variable, Y, is binary (0/1), which is the focus of the work in this paper. The idea of variable selection has also arisen in the operations research literature (Shih et al., 2014; Hwang et al., 2016).

Lastly, our work is also related to the marketing literature on product returns. Some empirical research on product returns has focused on finding the predictors of product returns. These studies consider predictors such as price (Anderson et al., 2009), category of merchandise, reason for return (Hess and Mayhew, 1997), transaction costs (Janakiraman and Ordóñez, 2012; Mollenkopf et al., 2011; Hess et al., 1996), customer characteristics (e.g., income, married), and binary indicators for gift purchases, holiday purchases, purchases in new categories, purchases in new distribution channels, and products purchased on sale (Petersen and Kumar, 2009, 2015). However, because of the hypothesis-driven nature of these studies, the attributes used are set in advance by the authors, instead of being automatically generated from the data; moreover, these studies are executed on proprietary data, which we do not have access to.

In this paper, we introduce a feature selection method that builds upon previous work. It is specifically suitable to the setting where features are highly correlated. It is an attribute selection procedure that directly models automatically generated attributes. In contrast to the previous work on automatically generated relational attributes, in the current work we develop an a data-driven method to select the best predictors. Our method is data-driven because it exploits the peculiar nature of these automatically generated attributes (their large number, the presence of complexity information, and the correlation among them) to achieve a higher performance. Another original aspect of our procedure is that it also includes meta-information on the complexity of the features, an aspect no method before has taken into consideration. This unique combination allows our method to model the automatically generated attributes from databases and outperform existing state-of-the-art methods. We test our method on a real-world data set on product returns.

3.3 Methodology

Our method consists of executing a "Screening" procedure followed by a "Cluster-based feature selection" procedure. These procedures aim to address the high dimensionality of the data and the multicolinearity of the AG relational attributes, respectively. In our experiments, we assess the individual merit of these two procedures, by executing them independently or one after the other.

3.3.1 Screening Procedure

The high dimensionality problem is primarily addressed by only selecting variables up to a certain (optimal) level of complexity. This is made possible by using the complexity score provided by Dataconda for each generated attribute. The optimal complexity level is found, for example, by subsequently testing the out-of-sample accuracy obtained by including only variables up to a certain maximum complexity $C = 1, 2, 3, ...$ The accuracy of the model is expected to increase as C increases (because the data set contains more attributes), but only up to a certain point, after which the noise introduced by including a large number of excessively complex attributes causes a decrease in the out-of-sample performance because of overfitting.

The second mechanism used to address the high dimensionality issue is to apply screening procedures based on regularization (shrinking) methods that choose sparser models. Algorithm 1 illustrates the screening procedure. Note that we leave the options of (1) selecting either lasso or elastic net as shrinkage method, (2) consider the complexity scores or not, and (3) using the 1-sd method or the default to select the regularization parameter λ, to step 12.

The current algorithms for lasso (e.g. coordinate descend and least angle regression) evaluate the lasso expression for a sequence of λ values (the regularization parameter). Usually, for each value of the λ sequence, a ten-fold cross validation is performed and the optimal λ

Procedure 1 Screening Procedure

1: Select shrinkage method $M \in$ {lasso, elastic net}
2: Select whether to take into account the complexity score (*complexity=considered*) or not (*complexity=not considered*)
3: Select whether to choose the regularization parameter via the 1-sd rule (*1-sd = yes* or via cross validation (*1-sd = no*)
4: **if** complexity = *considered* **then**
5: **for** variable complexity $c = 1, \ldots, C$ **do**
6: Select variables with complexity smaller than or equal to c.
7: Compute the cross-validated accuracy of method M
8: Record the complexity c^* that results in the best cross-validated accuracy
9: **end for**
10: Select only all attributes up to the complexity score c^*
11: **else**
12: Select all attributes
13: **end if**
14: Execute shrinkage method M selecting the value of λ with the 1-sd rule if *1-sd = yes* or with cross validation if *1-sd = no*
15: Return the attributes selected in the previous step

value that results in the least cross validation error is selected. However, another criterion for selecting the optimal λ value of lasso is to select a value which is greater than the value that results in the least cross validation error, and still falls within one standard deviation of its error. This method for choosing the λ parameter is known as the "one-standard-error" rule (*1-sd*) and results in sparser models (Hastie et al., 2009; Friedman et al., 2010). Since the computation time for finding the higher value is constant, the choice of applying the "one-standard-error" rule or not has negligible effect on the overall computation time.

3.3.2 Cluster-based Feature Selection

As explained earlier, the multicollinearity is caused by the presence of high correlation within groups of the generated attributes. The multicollinearity problem is addressed by clustering the variables into groups of highly correlated variables. We use hierarchical clustering because it has the advantage of making it easier to visually confirm that the chosen number of clusters is correct when compared to non-hierarchical clustering algorithms. The optimal number of

clusters is determined using the mean silhouette score (Reynolds et al., 2006). We tried using other criteria, such as the Bayesian Information Criterion (BIC) score of a Gaussian Mixture Model, and obtained similar results. After clustering the variables into K clusters, we remove some of the clusters from the model by executing a group lasso (Yuan and Lin, 2007) where the group labels are the cluster labels. Group lasso has the attractive property of either shrinking all of the coefficients of a group of variables exactly to zero or none of them to zero. The number of clusters removed by the group lasso, L, depends on the penalizing parameter λ (equation 3.6). We select the value of λ that minimizes the average out-of-sample mean square error obtained in a 10-fold cross validation. After the group lasso, we are left with $K - L$ clusters of variables with high predictive performance.

Finally, in order to ensure that the final variables are uncorrelated with one another, we retain only one variable for each cluster: the variable with the largest individual predictive performance within each cluster. We measure the individual predictive performance of a variable by executing a 10-fold cross-validated simple logistic regression and averaging the area under the receiver operating characteristic curve (AUC) obtained in each fold. Thus, this procedure terminates with $K-L$ selected variables. Procedure 2 illustrates our cluster-based feature selection.

Procedure 2 Cluster-based feature selection

1: **Step 1:**
2: Select the number of clusters K using the mean silhouette score.
3: Cluster the variables into K clusters by hierarchical clustering.
4: **Step 2:**
5: Using the clustering labels as the labels for group lasso, eliminate L out of K clusters.
6: **Step 3:**
7: In each cluster, rank the variables according to their individual predictive performance.
8: **Step 4:**
9: Select the variable with the largest individual predictive performance within each of the $K - L$ remaining clusters.

Figure 3.2 illustrates the overall method when Procedures 1 and 2 are executed one after the other.

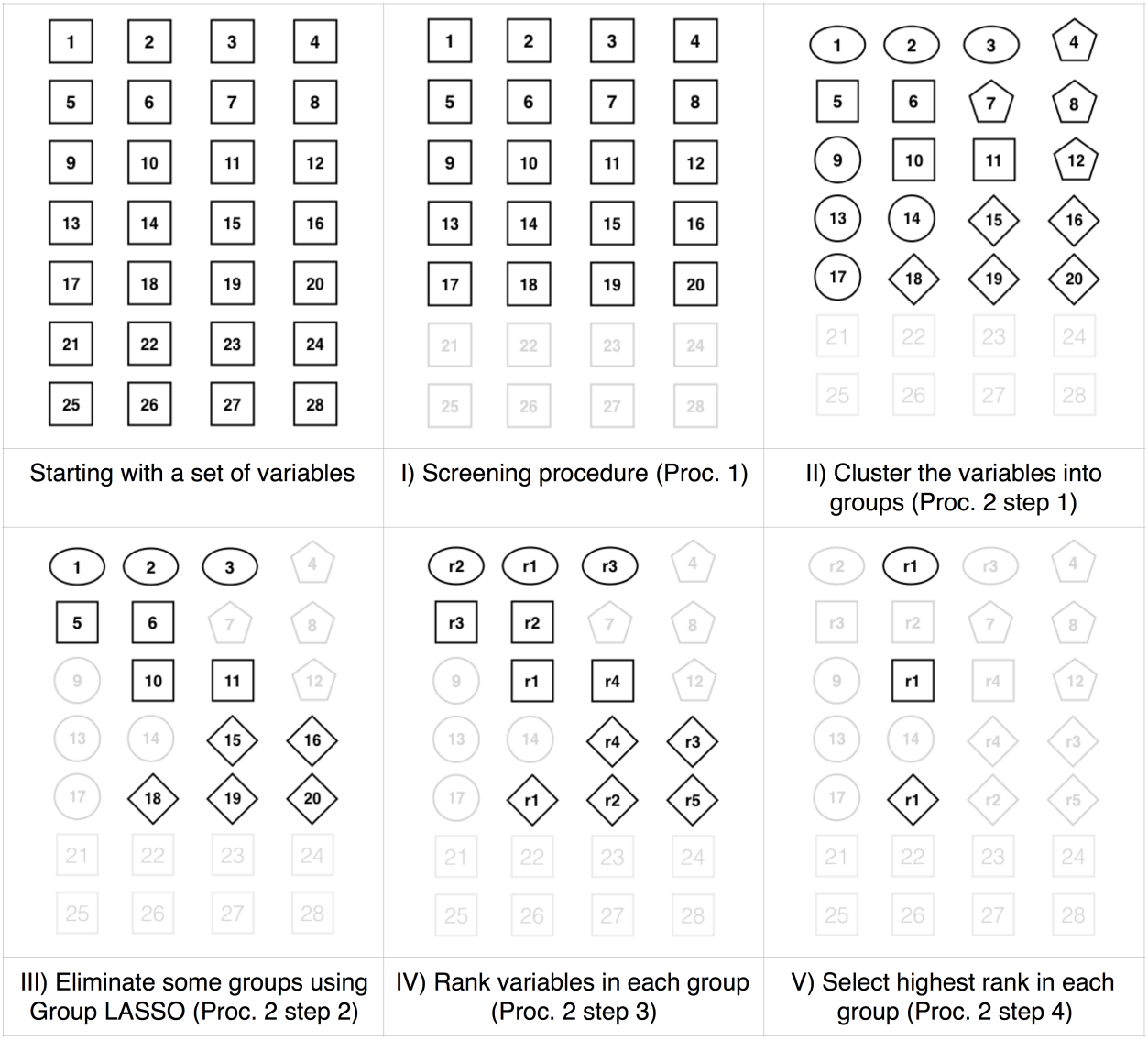

Figure 3.2: Illustration of the proposed feature selection method.

3.4 Results

In this section, we measure the performance of several variants of our proposed procedure on simulated data, with the goal of finding the best design choices.

3.4.1 Simulated Data

We carry out a simulation study to evaluate the performance of the proposed feature selection procedure. We generated 55 different instances of the product returns database, which have the same schema as the running example (Figure 3.1). The databases and the flattened tables can be downloaded from `http://www.dataconda.net/research-resources.html`.

Each of the 55 databases were generated as follows. First, the table Clients was populated with 30 random clients who have a 60% probability of being male and 40% probability of being female and whose age is uniformly distributed between 18 and 80 years. Second, the table Products was populated with 10 products having a price uniformly distributed between \$10 and \$500. Third, the table Purchases was populated by generating, for each client, a random number of purchases uniformly distributed between 10 and 25. The first purchase of each client was carried out on a random day between 1/1/2012 and 12/31/2013 and the time between two subsequent purchases was exponentially distributed with a mean of 15 days. Each purchase was made at a random time of the day, it had a 50% probability of being online, and it involved a random product.

After this procedure, all three tables were completely populated except for the class attribute Return of the table Purchases. The value of Return for each purchase was computed, using Equation 3.8, so that the probability of Return = 1 depends on the values of the following five variables, "True Variables":

X_1 = Price of the product.

X_2 = Age of the client.

X_3 = The proportion of products returned in the past by the same client.

X_4 = The age of the last client who purchased the same product online.

X_5 = The number of products costing more than \$400.00 among the client's past purchases.

(The complexity score of the variables X_1 to X_5 are 2, 2, 3, 4, and 4, respectively.)

$$Prob(Return) = Logit^{-1}(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5) \tag{3.8}$$

In an attempt to obtain a realistic database, we first note that only a small proportion of purchases were returned; hence, we set β_0 to a negative value. Second, variables X_1 and X_3 are known to impact the probability of returning a product. In particular, the empirical study by Anderson et al. (2009) shows that expensive items are more likely to be returned than cheap items; the empirical study by Petersen and Kumar (2009) shows that customers who returned more products in the past tend to return more products in the future. So, we set both β_1 and β_3 to have positive values.

It is less clear how to set the values of the other coefficients (β_2, β_4, and β_5) in equation 3.8, as there is no evidence that X_2, X_4, or X_5 impact product returns in reality. However, we choose to have them impact product returns in our simulated data experiments in order to measure the ability of our feature selection method to select not only the simplest true variables (those with the smallest complexity score) but also the most complex ones. We decide to set the values of β_2, β_4, and β_5 so that the proportion of returned purchases is between 10%-50%. A larger proportion would be unrealistic, whereas a smaller proportion would have two negative effects. First, X_3 would be irrelevant as its value would be close to 0 (or 1) for all purchases; second, the resulting data set would be excessively imbalanced, making it hard for any feature selection algorithm to perform well.

The task of achieving a predetermined proportion of returns is made more difficult by the fact that the magnitude of β_3 influences the future purchases of the same client. If β_3 is excessively large, then each client's behavior converges through simulated time to either always returning or never returning a purchase. If, for instance, a client's first purchases are

all returned, then the value of X_3 for the client's next purchase will be 1. If β_3 is large, then the probability that the next purchase is returned is very high; this, in turn, increases the probability that all of the client's future purchases are returned. Analogously, if a client's first purchases are not returned and β_3 is very large, then the client's probability of returning converges to 0. β_3 cannot be set to an excessively small value either, or else no feature selection algorithm is able to recognize it as true variable.

Driven by these considerations, we adopt a trial-and-error approach and set the values of the coefficients as follows:

$$Prob(Return) = Logit^{-1}(-3 + 0.02X_1 - 0.04X_2 + 2.4X_3 - 0.1X_4 + 0.5X_5) \tag{3.9}$$

After constructing the 55 relational databases, we use Dataconda (Samorani, 2015) to generate a flat mining table for each of them, with attributes up to complexity 5. The output is the 55 data sets whose characteristics (number of rows, columns, and percentage of returns) are reported in the first columns of Tables 1 and 2.

3.4.2 Results on Simulated Data

In this section, we evaluate the proposed algorithm in terms of its prediction performance and in terms of its ability to retrieve the "true variables", and we compare the results to those obtained by other feature selection algorithms.

To measure the effectiveness of Algorithm 1 in addressing high dimensionality, we consider different versions of the screening procedure. This can be obtained by setting M = lasso or M = elastic net as the shrinkage method, by taking into consideration the complexity scores of the attributes (*complexity = considered*) or not (*complexity = not considered*), and by selecting the regularization parameter via the 1-sd method (*1-sd = yes*) or the cross-validation method (*1-sd = no*).

To measure the effectiveness of Algorithm 2 in addressing multicollinearity, we compare

the results obtained by executing Algorithm 2 after Algorithm 1 (*cluster = yes*) or executing Algorithm 1 alone (*cluster = no*).

Although the options considered result in 16 combinations, we do not report the results of the four combinations with *1-sd = yes* and *cluster = yes*, because using the 1-sd method results in a very small number of attributes, which does not make sense to cluster. Thus, to summarize, we present the results relative to 12 combinations of our algorithm that can be obtained with the following parameter values:

- *From procedure 1:* $M \in \{$*lasso, elastic net*$\}$
- *From procedure 1: Complexity* $\in \{$*considered, not considered*$\}$
- *From procedure 1: 1-sd* $\in \{$*yes, no*$\}$
- *From procedure 2: cluster* $\in \{$*yes, no*$\}$
- *Combinations where 1-sd = yes and cluster = yes are excluded.*

Predictive performance

For each data set, we measure the performance of the feature selection procedures using a 10-fold cross validation, where for each fold we do the following:

1. Run the feature selection algorithm on the current training set
2. Train a logistic regression model with the selected variables on the training set
3. Record the AUC obtained on the current validation set

Tables 3.1-3.2 report the summary results obtained under all 12 parameter combinations on the 55 simulated data sets. The Tables report the results obtained by the six combinations with $M =$ *lasso*. We also executed our tests for the six combinations with $M =$ *elastic net*, but we exclude these results here because they are inferior to the combinations with $M =$

lasso (see Appendix A). The first column reports the data set number (1 to 55); the second, third, and fourth columns report the number of rows (i.e., Purchases), the number of columns (i.e., attributes generated by Dataconda), and the proportion of positive instances (*pos*) in each data set. The following columns report the average AUC and the number of selected features (n) obtained under each parameter combination throughout the cross validation procedure.

We also compare the performance of our methods to feature selection methods based on Random Forest, NaiveBayes, and Bagged Trees (Han et al., 2011) obtained by *caret* (Kuhn and Johnson, 2013). The last columns of Tables 3.1-3.2 report the performance obtained by Random Forest and Bagged Trees. NaiveBayes are excluded due to space limitations.

From Tables 3.1-3.2, the following findings emerge. First, considering the average and standard deviation of the AUC obtained (last two rows of Table 3.2), it is evident that all methods perform similarly. However, a paired t-test that compares the 55 cross-validated AUC values reveals that when *complexity = considered*, all three versions of our algorithms outperform ($p < 0.001$) random forest and Naive Bayes, whereas only 1-sd outpeforms Bagged Trees (the other versions are statistically similar).

However, if we consider the average number of selected features, it is immediately clear that all versions of our algorithm (and in particular the versions that use the clustering method of Algorithm 2) tend to select a significantly smaller number of features than the machine learning techniques.

Our experiments were executed on an Intel Xeon CPU E5-2670 v2 2.50 GHz, without using any parallel computation. The average time taken by the methods on each data set are shown in the last row of Table 3.2. Taking the complexity score into consideration reduces the computational time by half. Using clustering leads to a four-fold increase in computational time; this is mainly due to the various steps required to run Algorithm 2 (e.g. hierarchical clustering). Using the elastic net shrinkage method increases the computation time by almost 50% compared to using lasso as the shrinkage method. With the exception of bagged trees,

Table 3.1: Results obtained with M = *lasso* – part 1 of 2 – **The best performing combination is in bold**

			complexity = considered							complexity = not considered						random forest		bagged trees	
			1sd clust.	no no		yes no		**no yes**		no no		yes no		no yes					
	nrow	ncol	pos	AUC	n	AUC	n	**AUC**	**n**	AUC	n	AUC	n	AUC	n	AUC	n	AUC	n
1	478	2405	0.39	0.79	34	0.8	16	**0.79**	**6**	0.76	50	0.8	13	0.78	16	0.81	2	0.43	2405
2	533	2404	0.26	0.83	33	0.84	18	**0.83**	**6**	0.79	58	0.83	17	0.81	10	0.86	20	0.85	17
3	518	2396	0.21	0.84	30	0.85	13	**0.84**	**3**	0.81	46	0.84	14	0.8	6	0.84	20	0.86	20
4	500	2283	0.16	0.87	29	0.88	10	**0.88**	**4**	0.84	25	0.88	7	0.86	7	0.85	13	0.86	20
5	541	2392	0.16	0.88	22	0.88	12	**0.87**	**5**	0.86	47	0.88	24	0.87	18	0.91	18	0.92	6
6	535	2384	0.36	0.81	31	0.82	12	**0.82**	**6**	0.8	46	0.81	22	0.8	17	0.85	2	0.86	19
7	517	2402	0.37	0.78	24	0.8	13	**0.79**	**4**	0.77	60	0.78	22	0.77	4	0.73	19	0.74	18
8	543	2413	0.3	0.81	28	0.82	16	**0.8**	**8**	0.79	44	0.81	13	0.8	11	0.75	12	0.76	18
9	492	2351	0.24	0.82	34	0.83	16	**0.84**	**4**	0.78	34	0.83	13	0.84	7	0.85	8	0.62	2351
10	482	2395	0.42	0.78	32	0.78	21	**0.76**	**7**	0.74	64	0.77	33	0.76	26	0.81	2	0.83	20
11	548	2424	0.39	0.77	20	0.78	10	**0.78**	**6**	0.76	37	0.78	11	0.76	13	0.79	2	0.81	20
12	539	2416	0.41	0.78	40	0.78	21	**0.77**	**8**	0.75	50	0.78	17	0.75	12	0.76	18	0.77	12
13	496	2395	0.19	0.88	19	0.88	12	**0.87**	**4**	0.87	25	0.88	13	0.86	5	0.63	2395	0.91	12
14	549	2375	0.2	0.83	47	0.88	28	**0.79**	**5**	0.7	66	0.86	16	0.85	6	0.85	3	0.85	15
15	506	2414	0.23	0.83	26	0.83	11	**0.83**	**4**	0.8	49	0.81	14	0.83	6	0.84	8	0.86	11
16	482	2413	0.25	0.84	43	0.84	15	**0.78**	**3**	0.81	51	0.85	12	0.77	6	0.82	7	0.82	13
17	547	2375	0.17	0.88	41	0.87	18	**0.87**	**6**	0.82	44	0.84	8	0.85	5	0.85	11	0.84	5
18	553	2397	0.33	0.8	29	0.81	16	**0.8**	**4**	0.76	68	0.81	19	0.77	9	0.82	16	0.82	12
19	530	2387	0.41	0.77	36	0.77	14	**0.76**	**5**	0.75	43	0.78	11	0.76	9	0.75	7	0.82	19
20	500	2327	0.1	0.86	18	0.9	5	**0.91**	**3**	0.88	10	0.89	5	0.9	4	0.87	11	0.86	7
21	517	2392	0.36	0.79	34	0.81	16	**0.8**	**8**	0.7	62	0.8	12	0.79	38	0.52	2392	0.8	20
22	550	2354	0.26	0.86	25	0.86	14	**0.85**	**4**	0.85	21	0.85	13	0.84	7	0.53	2354	0.86	10
23	543	2394	0.36	0.77	39	0.77	17	**0.77**	**8**	0.75	47	0.77	19	0.78	14	0.75	6	0.77	17
24	540	2396	0.16	0.88	34	0.88	13	**0.75**	**2**	0.87	17	0.87	7	0.87	2	0.52	2396	0.6	2396
25	511	2417	0.52	0.73	36	0.73	18	**0.72**	**8**	0.68	77	0.72	31	0.72	29	0.74	16	0.74	16
26	504	2282	0.13	0.81	34	0.83	14	**0.81**	**8**	0.77	61	0.85	18	0.83	29	0.79	20	0.73	2282
27	516	2413	0.33	0.8	24	0.81	13	**0.81**	**7**	0.75	67	0.8	26	0.79	30	0.78	11	0.78	18
28	544	2428	0.17	0.88	28	0.88	16	**0.88**	**5**	0.83	52	0.86	24	0.87	11	0.89	18	0.86	11
29	536	2415	0.25	0.82	31	0.83	13	**0.78**	**4**	0.81	23	0.82	9	0.81	5	0.58	2415	0.84	18

Table 3.2: Results obtained with M = *lasso* – part 2 of 2 – **The best performing combination is in bold**

			complexity = considered							complexity = not considered						random forest		bagged trees	
			1sd clust.	no no		yes no		**no yes**		no no		yes no		no yes					
	nrow	ncol	pos	AUC	n	AUC	n	AUC	n	**AUC**	**n**	AUC	n	AUC	n	AUC	n	AUC	n
30	575	2399	0.54	0.71	30	0.72	13	**0.69**	**10**	0.68	67	0.72	25	0.68	22	0.69	11	0.69	11
31	578	2375	0.27	0.83	24	0.83	15	**0.74**	**3**	0.83	23	0.83	13	0.82	10	0.83	3	0.83	14
32	503	2319	0.17	0.85	47	0.85	28	**0.84**	**10**	0.82	29	0.84	5	0.83	14	0.76	7	0.76	4
33	540	2388	0.26	0.84	26	0.85	14	**0.85**	**6**	0.82	32	0.84	18	0.85	7	0.81	17	0.81	20
34	555	2315	0.12	0.84	22	0.85	11	**0.84**	**7**	0.81	41	0.81	13	0.82	19	0.81	14	0.87	12
35	517	2348	0.15	0.86	22	0.87	7	**0.87**	**4**	0.83	31	0.88	8	0.85	9	0.79	2	0.81	12
36	548	2410	0.16	0.85	28	0.88	12	**0.88**	**4**	0.78	52	0.87	10	0.87	6	0.77	7	0.79	12
37	492	2366	0.16	0.87	26	0.88	12	**0.85**	**4**	0.83	31	0.85	7	0.88	6	0.83	7	0.85	14
38	504	2410	0.63	0.67	29	0.68	9	**0.66**	**5**	0.64	78	0.67	13	0.64	12	0.67	20	0.67	14
39	500	2396	0.29	0.83	23	0.84	14	**0.82**	**4**	0.84	27	0.83	15	0.8	13	0.72	18	0.75	14
40	537	2421	0.24	0.83	21	0.83	11	**0.83**	**6**	0.8	41	0.83	15	0.82	6	0.58	2421	0.82	14
41	510	2418	0.33	0.79	37	0.8	15	**0.78**	**5**	0.76	68	0.8	30	0.79	38	0.5	2418	0.55	2418
42	535	2423	0.3	0.8	27	0.81	9	**0.79**	**6**	0.78	59	0.79	20	0.79	22	0.8	6	0.82	16
43	501	2418	0.19	0.86	40	0.88	18	**0.77**	**4**	0.86	17	0.86	7	0.84	2	0.85	4	0.87	17
44	536	2390	0.37	0.8	27	0.81	15	**0.81**	**6**	0.76	46	0.81	12	0.79	10	0.77	19	0.8	15
45	534	2397	0.42	0.77	34	0.77	15	**0.7**	**6**	0.74	49	0.77	14	0.76	27	0.81	9	0.8	17
46	531	2420	0.54	0.73	39	0.73	19	**0.73**	**8**	0.7	71	0.73	17	0.68	63	0.53	2420	0.72	16
47	540	2371	0.25	0.82	27	0.83	14	**0.83**	**4**	0.79	37	0.83	15	0.84	7	0.82	16	0.83	14
48	496	2412	0.58	0.7	38	0.7	18	**0.7**	**8**	0.65	77	0.7	28	0.68	37	0.68	12	0.67	18
49	505	2391	0.4	0.76	31	0.76	18	**0.76**	**11**	0.75	42	0.76	18	0.75	20	0.8	14	0.8	13
50	497	2300	0.24	0.86	29	0.87	14	**0.85**	**4**	0.84	27	0.86	13	0.85	4	0.4	2300	0.58	2300
51	527	2404	0.34	0.82	31	0.83	14	**0.82**	**6**	0.81	34	0.83	16	0.81	8	0.81	14	0.78	20
52	510	2383	0.2	0.86	34	0.88	20	**0.86**	**6**	0.78	56	0.88	26	0.85	12	0.57	2383	0.57	2383
53	577	2300	0.13	0.83	40	0.88	15	**0.86**	**4**	0.84	13	0.87	6	0.82	2	0.63	2300	0.94	20
54	528	2389	0.33	0.8	26	0.81	11	**0.8**	**6**	0.78	27	0.81	11	0.79	10	0.6	2389	0.84	10
55	494	2321	0.12	0.86	26	0.88	10	**0.86**	**4**	0.8	57	0.89	17	0.89	15	0.97	14	0.95	14
avg	524	2384	0.29	0.81	31	0.83	15	**0.81**	**6**	0.78	45	0.82	15	0.81	14	0.75	530	0.79	310
sd	24	37	0.13	0.048	7	0.051	4	**0.054**	**2**	0.055	17	0.049	7	0.056	11	0.12	980	0.1	780
computation time (sec.)				175		175		**655**		290		1086		67		63		826	

the other machine learning methods take a shorter time than any version of our procedure. Finally, note that models with (1-sd = yes) have the same computation time than those with (1-sd = no).

In summary, our methods have the potential to obtain a superior predictive performance with a smaller set of features than machine learning methods. We believe that the extra time taken by our method justifies the increase in performance: an analyst interested in uncovering new knowledge is willing to wait a few extra minutes to reduce the number of selected attributes by a few hundred.

Finding the True Variables

After considering the predictive performance, we now shift our focus to the model's ability to retrieve the true variables ($X1$-$X5$). We evaluate this ability through precision, recall and f1-score:

$$precision = \frac{number\ of\ true\ features\ retrieved}{total\ of\ features\ retrieved}$$

$$recall = \frac{number\ of\ true\ features\ retrieved}{number\ of\ true\ features}$$

$$f1\text{-}score = 2\frac{precision \times recall}{precision + recall}$$

For each data set, we measure the performance of the feature selection procedures through a 10-fold cross validation, where for each fold we carry out the following:

1. Run the feature selection algorithm on the current training set
2. Record the precision, recall and f1-score obtained

Figures 3.3 to 3.6 show the results of all the feature selection procedures on the 55 different simulated data sets.

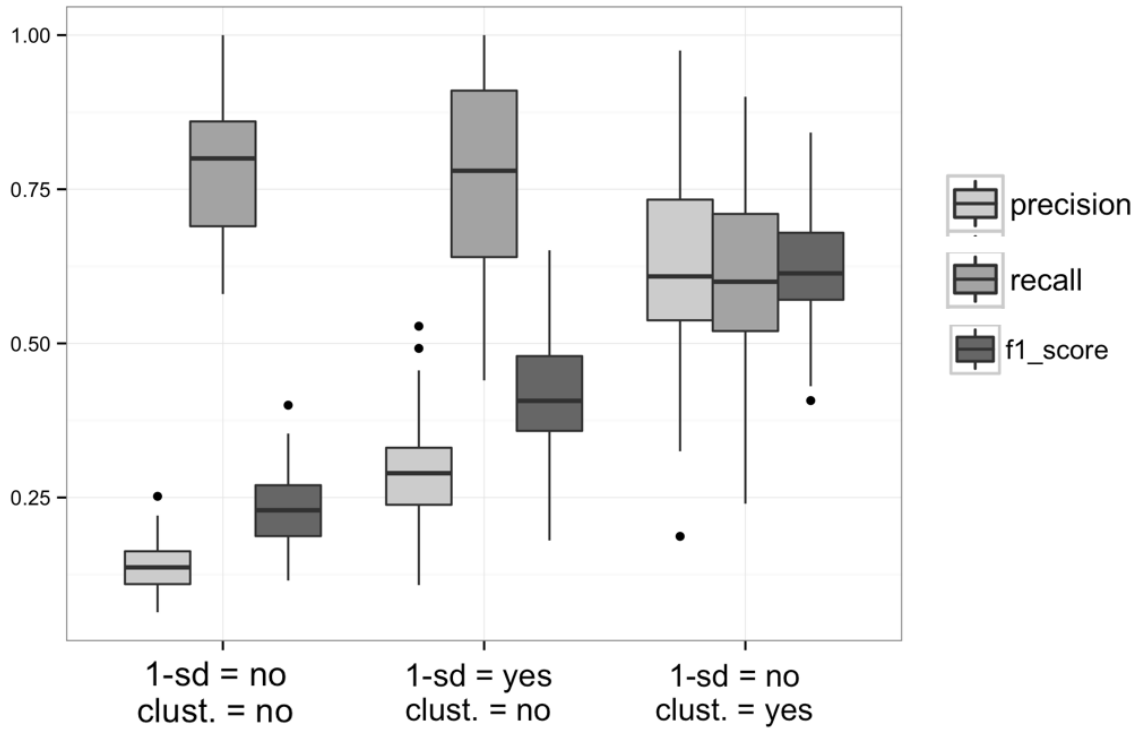

Figure 3.3: Performance for parameter combinations with $M = lasso$ and $complexity = considered$

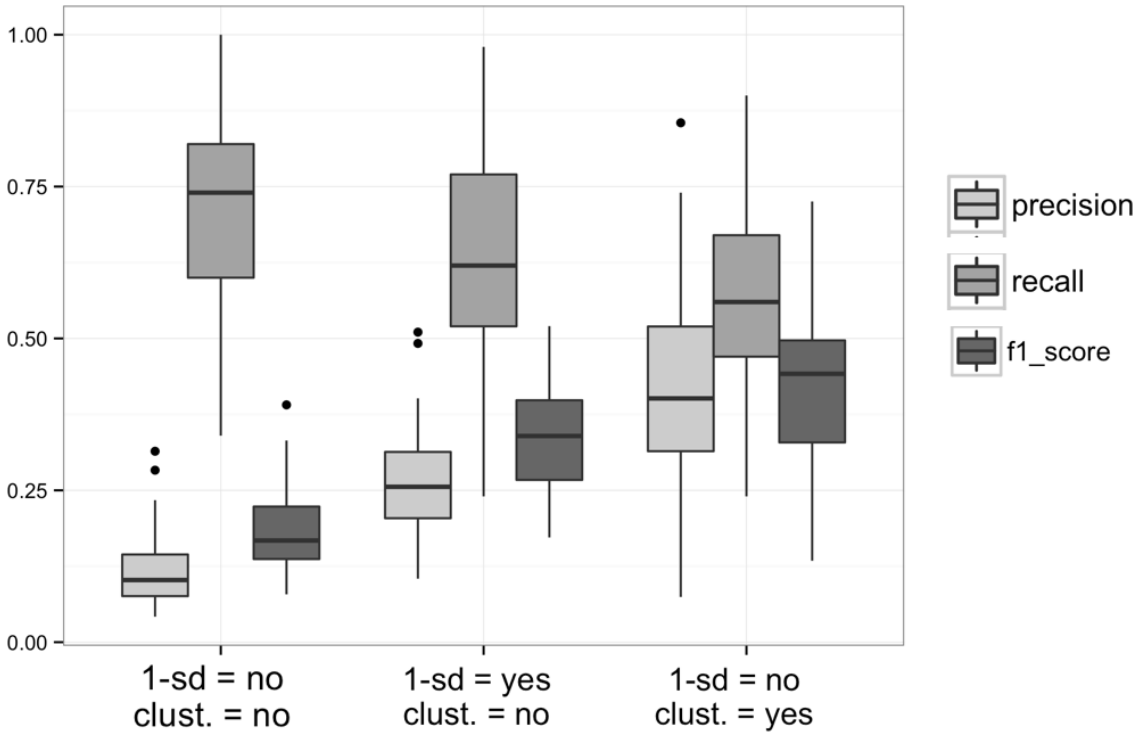

Figure 3.4: Performance for parameter combinations with $M = lasso$ and *complexity = not considered*

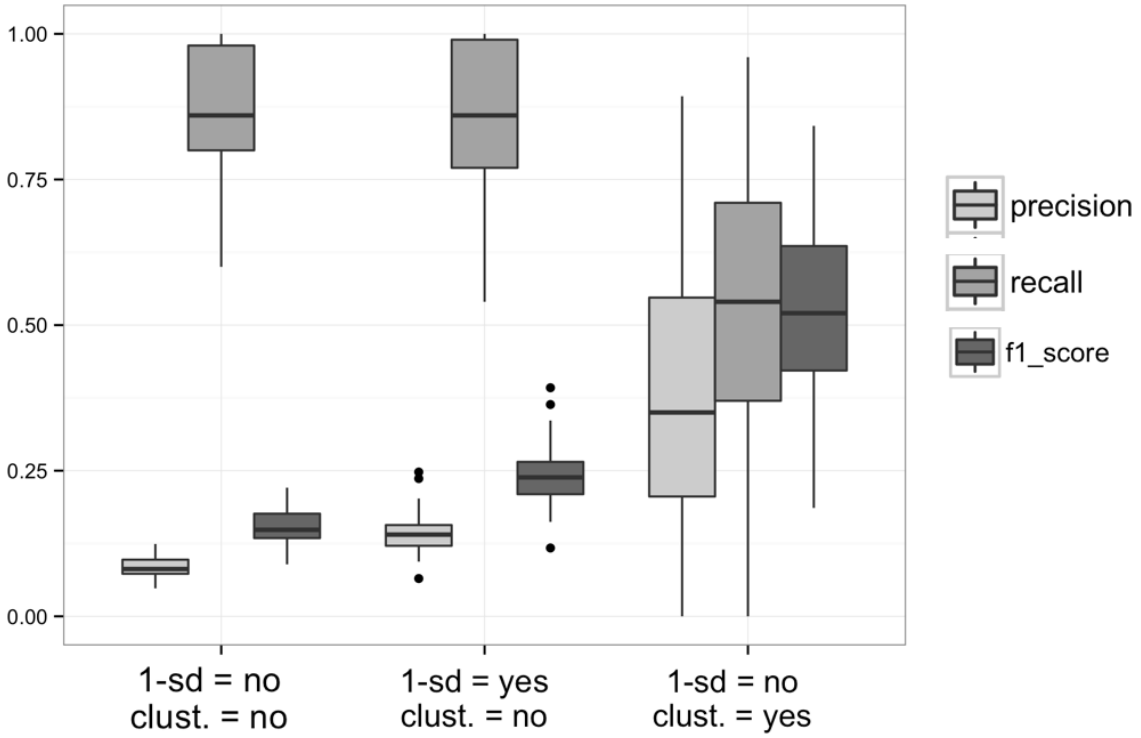

Figure 3.5: Performance for parameter combinations with $M = ENet$ and *complexity* = *considered*

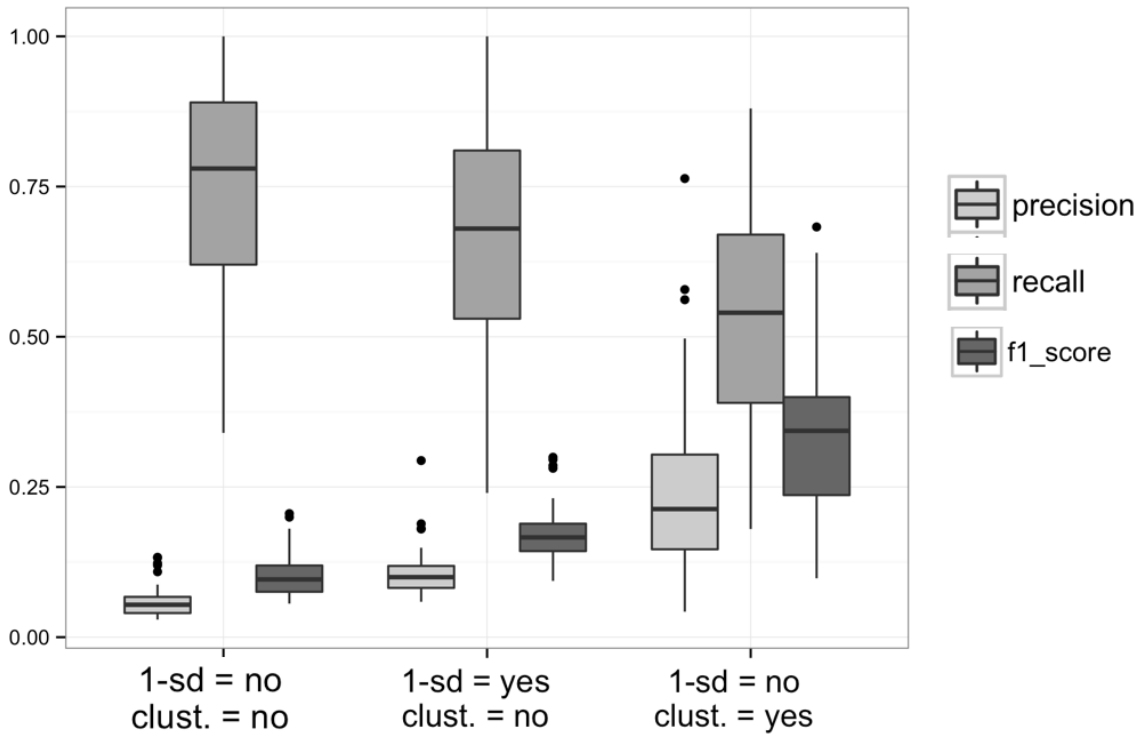

Figure 3.6: Performance for parameter combinations with $M = ENet$ and *complexity = not considered*

Each figure reports the results of three different parameter combinations and are illustrated using three boxplot charts, which report the distribution of the cross-validated precision, recall, and f1-score obtained in the 55 data sets.

In all cases, the clustering method has a higher f1-score than all the other methods. The reason lies in the superior ability of the clustering method to find a higher-quality balance between precision and recall, whereas the other parameter combinations tend to obtain a high recall but a low precision. Also, comparing figure 3.3 with figure 3.4 and figure 3.5 with figure 3.6, one can conclude that taking the complexity score into consideration improves the performance of retrieving the true variables. Finally, comparing figure 3.3 with figure 3.5 and figure 3.4 with figure 3.6, it is clear that using lasso as a shrinkage method results in a larger f1-score than using elastic net.

Thus, the best performing method consists of the following design choices: M = lasso, complexity = considered, 1-sd = no, cluster = yes. We executed a t-test to show that the f1-

score obtained by using the best design choice combination, adjust for multiple comparisons here too for the p-values, is significantly higher than the f1-score obtained by using the other combinations. The results of this t-test are reported in Table 3.3, where each row reports a combination, the performance obtained averaged across the 55 data sets, and the p-value indicates whether the best combination (reported in the first row) performs significantly better than the best performing combination. It is clear from the table that the f1-score of the best performing method is statistically higher than other models with different design choices.

Table 3.3: Performance obtained by the various configurations of design choices.

	M	complexity	1-sd	clust.	recall	precision	f1_score	p-value$^+$
*	**lasso**	**considered**	**no**	**yes**	**0.60**	**0.62**	**0.58**	**NA**
	lasso	considered	no	no	0.78	0.14	0.24	$<$ 1e-09
	lasso	considered	yes	no	0.77	0.29	0.42	$<$ 1e-09
	lasso	not considered	no	yes	0.56	0.42	0.41	$<$ 1e-09
	lasso	not considered	no	no	0.71	0.12	0.19	$<$ 1e-09
	lasso	not considered	yes	no	0.64	0.26	0.34	$<$ 1e-09
	ENet	considered	no	yes	0.54	0.38	0.42	$<$ 1e-09
	ENet	considered	no	no	0.87	0.08	0.15	$<$ 1e-09
	ENet	considered	yes	no	0.85	0.14	0.24	$<$ 1e-09
	ENet	not considered	no	yes	0.52	0.25	0.28	$<$ 1e-09
	ENet	not considered	no	no	0.75	0.06	0.10	$<$ 1e-09
	ENet	not considered	yes	no	0.68	0.11	0.17	$<$ 1e-09

***** The first row of the table contains the configuration of the best performing model.
$+$ p-value of a comparison between each model with the best performing model.

In order to further evaluate the proposed feature selection method, we undergo a computational experiment where we compare the performance of our method under increasing sparsity levels with the following methods: lasso, support vector machines with l_1 regularization (SVMs), and random forests (RFs). SVMs are considered to be one of the best machine learning algorithms for regression and classification problems (Hastie et al., 2009). Similar to the expressions introduced in Section 2, the loss function of SVMs can be combined with the l_1 norm of the coefficient vector in order to achieve sparsity. Specifically, in the case of classification, suppose y_i represents the response variable $y_i \in \{-1, +1\}, i = 1, ..., n$, and x_i

represents the predictor variables for observation i, then the l_1-regularized l_2-loss SVM solves the following expression (Fan et al., 2008):

$$\min_{\beta} \sum_{i=1}^{n} (max(0, 1 - y_i \beta^T x_i))^2 + \lambda \|\beta\|_1 \tag{3.10}$$

RFs are also widely used in practice for feature selection. RFs begin with a bootstrap sample of the data (equal in size to the original data), then builds a classification tree where instead of using all of the variables, in each node, it uses a subset of the variables. This procedure is repeated a large number of times, denoted by N. Given new data, it predicts the response variable as the majority vote of the N previously trained trees. During the training process of RFs, it computes a measure of importance for each variable. This measure is computed as follows. For calculating the importance of the n^{th} variable, RF randomly permutes the values of that variable in the left out cases of the k^{th} tree, and then computes the error rate of that tree. The difference of this error rate with the unpermuted case is calculated as the importance measure of variable n (Breiman, 2001).

For each sparsity level $s = 1, ..., 17$, we calculate the true-positive (the number of true variables retrieved) of each feature selection method for all 55 simulated data sets. The constraint to select s variables is implemented as follows. For our method, lasso, and SVMs, we find the value of λ that results in selecting s variables on the current data set. For RFs, we train them on the current data set and then select the s most important variables (the importance measure from the output of random forests). Table 3.4 shows the average true-positive of all four feature selection methods.

3.4.3 Real-World Data

In this section, we show the results of the proposed feature selection method on a real dataset. The database used is the ISMS Durable Goods data set from Ni et al. (2012), a real world database that contains the transactions made at a large US consumer electronics retailer.

Table 3.4: Average number of true variables (out of five) retrieved for different sparsity levels

s	proposed method	lasso	SVM	RF
1	0.98	0.78	0.00	0.67
2	1.79	1.06	0 .00	0.87
3	2.39	1.35	0.10	1.04
4	2.91	1.64	0.62	1.18
5	3.10	1.83	0.79	1.24
6	3.26	2.29	0.79	1.27
7	3.26	2.58	1.00	1.31
8	3.65	2.54	0.96	1.33
9	3.54	2.91	1.00	1.38
10	3.38	3.16	1.06	1.40
11	3.30	3.12	1.11	1.49
12	3.54	3.40	1.09	1.51
13	3.61	3.44	1.59	1.55
14	3.54	3.48	1.71	1.58
17	3.64	3.65	2.21	1.62

The ISMS Durable Goods dataset, which can be purchased from the INFORMS Marketing Science Society website[1], has one row per transaction (a purchase or a return) and columns indicating the client_id, the store_id, the product_id and its category and brand.

We first normalize the data into the following tables: *Purchases* (which contains the class "Returned"), *Clients*, *Brands*, *Locations*, *Product Categories*. Then, we use Dataconda to generate a flat mining table of purchases and we execute our feature selection methods on the generated flat table. The resulting flat table has 6305 rows and 1330 attributes.

Our first comparison is between the predictive performance of our method to that obtained by a fictitious human analyst who, driven by findings from the marketing literature, manually builds attributes that should be correlated to product returns. The fictitious human analyst uses the following 10 attributes to characterize each purchase:

B1 Online purchase (0/1)

B2 Quantity purchased

[1] `https://www.informs.org/Community/ISMS/ISMS-Research-Datasets`

B3 Unit price ($)

B4 Presence of children at home (0/1)

B5 Client's gender

B6 Client's age

B7 Client's income

B8 Total products (in $) purchased and kept (i.e., not returned) by the client in the past

B9 Total products (in $) returned by the client in the past

B10 Proportion of purchases returned by the client in the past

Attributes B1-B7 are included because they can be obtained easily, either because they are readily available from the Purchases table or because they can be obtained by simply joining Purchases with Clients. Attribute B3 is included because it is known to be positively correlated with the return probability (Anderson et al., 2009). Attributes B8-B10 are included because the client's past return behaviour (either in terms of dollars purchased and returned or in terms of proportion of returned purchases) is known to impact future return behaviour (Petersen and Kumar, 2009).

Table 3.5 compares the predictive performance obtained by the fictitious human analyst (obtained by logistic regression) to that obtained by the variants of our algorithm on the real data. In these experiments, we set M = lasso and *1-sd* = *no*. To have a fair comparison, all the other four models were constrained to select 10 features. This is possible either by using the regularization parameter λ that results in 10 features (if *cluster* = *no*) or by setting the number of clusters to 10 (if *cluster* = *yes*).

The results indicate that all the designed models outperform the model built by the fictitious human analyst; however, there is no statistical difference among the designed models.

If we remove the constraint to select 10 features, our method obtains a slightly higher AUC at the cost of selecting more attributes. The results are reported in Table 3.6.

Table 3.5: Predictive performance on real-world data, with the constraint of selecting 10 features

	complexity = considered		complexity = not considered		
	cluster = no AUC(%)	cluster = yes AUC(%)	cluster = no AUC(%)	cluster = yes AUC(%)	manual AUC(%)
fold 1	54	52	52	53	47
fold 2	56	55	55	54	51
fold 3	62	60	62	59	54
fold 4	59	59	60	59	49
fold 5	53	54	53	54	48
fold 6	54	53	55	53	52
fold 7	56	57	56	54	51
fold 8	58	57	56	57	55
fold 9	53	56	56	57	49
fold 10	59	57	61	58	55
avg.	56	56	57	56	51
sd	2.9	2.6	3.1	2.5	3

Our second computational experiment consists of comparing the performance obtained under increasing sparsity levels by the following methods: our proposed method, lasso, support vector machines with l_1 regularization (SVMs), and random forests (RFs). For each desired sparsity level, we record two performance measures: the out-of-sample AUC and the average complexity (as measured by Dataconda) of the selected attributes. We cannot record precision or recall because the true variables are unknown in the real data. Our goal is to show that our method obtains a similar predictive performance than other methods by using less complex attributes.

For each sparsity level $s = 1, ..., 34$, we perform 100 out-of-sample tests obtained by randomly splitting the data set into training set (70%) and test set (30%) using different random seeds. We execute each competing method with the constraint to select s variables, and we record the average complexity of the selected variables and the out-of-sample AUC obtained by a logistic regression built using these s variables. Table 3.7 reports our results.

The results in Table 3.7 show that the methods considered achieve a similar predictive performance for any given sparsity level. However, our method selects variables that are less

Table 3.6: Performance on the real-world data, without the constraint on the number of features

	cluster = no		cluster = yes	
	AUC(%)	n. att.	AUC(%)	n. att.
fold 1	54	26	56	17
fold 2	56	32	55	24
fold 3	62	56	61	36
fold 4	62	38	60	28
fold 5	57	49	57	34
fold 6	58	28	56	19
fold 7	56	44	55	29
fold 8	58	51	59	37
fold 9	60	51	58	1
fold 10	61	35	58	25
avg.	58	41	58	25
sd	2.7	11	2.1	11

complex than other methods, which facilitates their interpretation and allows new knowledge to emerge.

3.5 Conclusion

Attribute generation techniques are a powerful tool to find patterns in relational data. By generating a large number of attributes automatically, these techniques have the potential to uncover information that a human analyst might not take into consideration. However, because of the characteristics of these attributes (high dimensionality and large correlation), regular feature selection algorithms struggle to find the determinant attributes.

This paper proposes a new multi-step feature selection procedure that proves to be effective on this type of data. It does so by (1) considering only the attributes up to a certain complexity score (which is information given by attribute generation software packages) and (2) clustering the attributes to find the groups of correlated features.

On simulated data, where the true variables are known, our clustering-based method

is more effective in retrieving the true variables than existing machine learning techniques. This ability results in a significantly smaller set of features and a similar or better predictive performance.

On a real-world retail data set, where the true variables are unknown, our clustering-based method results in both a better predictive performance and a sparser model than a model built using information from the marketing literature. When fixing a sparsity level, our method results in similar predictive performance but less complex variables than other competing methods.

To the best of our knowledge, the current work is the first to propose a data-driven method to select the best predictors among automatically generated relational attributes. Hence, there are several different research opportunities for future work in this area. One possibility is to modify the objective function of the lasso to take into account both multicollinearity and complexity scores.

Another possible research avenue is to perform the attribute selection at the same time as the attribute generation. That is, instead of executing an attribute selection procedure after generating the flat table, one could analyze the attributes while they are being generated. This would make it possible to deem an attribute to be a good or bad predictor after computing its value for only a few instances (as opposed to all instances). This may turn out to be particularly valuable for larger data sets.

In this paper feature selection methods are evaluated by the extent to which they are able to select the exact true variable. This means that a feature selection method would not be given any points if it selects a variable close to the true variable instead of the exact variable. The main complexity of this alternative approach is that it is generally hard to come up with an explicit definition for a "near-miss". In our opinion, this problem is worth exploring and could be the focus of future studies.

Lastly, in this paper we applied our method on attributes generated by Dataconda because it is the only freely available relational attribute generation software; however, our method

can be applied to attributes generated by other attribute generation techniques, such as the ones developed by Knobbe et al. (2001), Popescul and Ungar (2003), Perlich and Provost (2006), Samorani et al. (2011). An interesting future development is to compare the predictive performances obtained when applying our attribute selection method to the attributes generated by different attribute generation techniques.

Table 3.7: Performance vs sparsity table on the ISMS data set

	Our best method		lasso		SVM		RF	
s	AUC	cmplx	AUC	cmplx	AUC	cmplx	AUC	cmplx
1	0.55	3.20	0.55	3.20	0.52	3.15	0.52	2.98
2	0.56	3.15	0.55	3.25	0.51	3.19	0.52	3.09
3	0.55	3.15	0.55	3.27	0.52	3.18	0.53	3.20
4	0.55	3.15	0.55	3.35	0.51	3.18	0.53	3.21
5	0.55	3.13	0.55	3.32	0.51	3.19	0.53	3.26
6	0.56	3.12	0.55	3.39	0.52	3.19	0.54	3.31
7	0.56	3.12	0.56	3.34	0.52	3.20	0.54	3.32
8	0.56	3.13	0.55	3.38	0.52	3.20	0.54	3.34
9	0.56	3.13	0.56	3.36	0.52	3.20	0.54	3.34
10	0.56	3.14	0.56	3.33	0.53	3.20	0.54	3.34
11	0.56	3.14	0.56	3.36	0.53	3.20	0.54	3.36
12	0.56	3.13	0.57	3.42	0.53	3.20	0.54	3.38
13	0.56	3.14	0.56	3.32	0.53	3.20	0.54	3.39
14	0.56	3.13	0.56	3.39	0.53	3.20	0.54	3.37
15	0.57	3.13	0.57	3.37	0.53	3.20	0.54	3.39
16	0.57	3.14	0.56	3.39	0.52	3.20	0.55	3.41
17	0.56	3.14	0.55	3.39	0.54	3.20	0.54	3.41
18	0.57	3.14	0.56	3.39	0.54	3.20	0.54	3.38
19	0.57	3.13	0.56	3.36	0.54	3.20	0.54	3.38
20	0.57	3.14	0.56	3.35	0.55	3.20	0.55	3.39
21	0.57	3.13	0.57	3.37	0.55	3.20	0.55	3.39
22	0.57	3.13	0.57	3.46	0.56	3.20	0.55	3.38
23	0.56	3.13	0.57	3.36	0.55	3.20	0.55	3.40
24	0.57	3.13	0.57	3.35	0.58	3.20	0.55	3.38
25	0.57	3.13	0.57	3.35	0.56	3.20	0.55	3.41
26	0.57	3.13	0.57	3.40	0.57	3.20	0.55	3.41
27	0.57	3.13	0.56	3.33	0.57	3.20	0.55	3.40
28	0.57	3.13	0.57	3.38	0.58	3.20	0.56	3.42
29	0.57	3.14	0.57	3.50	0.55	3.20	0.55	3.43
30	0.57	3.14	0.58	3.35	0.56	3.20	0.55	3.46
31	0.57	3.14	0.56	3.46	0.59	3.20	0.55	3.50
32	0.56	3.14	0.56	3.38	0.57	3.20	0.55	3.40
33	0.56	3.13	0.55	3.38	0.55	3.20	0.55	3.43
34	0.57	3.13	0.58	3.43	0.56	3.20	0.55	3.43
average	0.56	3.14	0.56	3.37	0.54	3.20	0.54	3.36

Appendix A: Results for the six combinations with M = *elastic net* on the simulated data set

Table 3.8: Results obtained with M = *elastic net* – part 1 of 2

				complexity = considered						complexity = not considered						random forest		bagged trees	
			1sd clust.	no no		yes no		no yes		no no		yes no		no yes					
	nrow	ncol	pos	AUC	n	AUC	n	AUC	n	AUC	n	AUC	n	AUC	n	AUC	n	AUC	n
1	478	2405	0.39	0.74	63	0.79	42	0.79	8	0.73	110	0.79	38	0.75	22	0.81	2	0.43	2405
2	533	2404	0.26	0.81	50	0.83	34	0.83	8	0.73	89	0.83	39	0.81	22	0.86	20	0.85	17
3	518	2396	0.21	0.85	54	0.85	34	0.76	4	0.7	67	0.84	31	0.78	8	0.84	20	0.86	20
4	500	2283	0.16	0.85	50	0.88	24	0.86	6	0.83	57	0.88	23	0.84	10	0.85	13	0.86	20
5	541	2392	0.16	0.84	48	0.88	24	0.84	6	0.74	82	0.86	44	0.84	14	0.91	18	0.92	6
6	535	2384	0.36	0.8	56	0.81	37	0.73	6	0.74	100	0.75	59	0.79	7	0.85	2	0.86	19
7	517	2402	0.37	0.78	53	0.79	28	0.79	6	0.68	120	0.77	53	0.78	4	0.73	19	0.74	18
8	543	2413	0.3	0.8	50	0.82	29	0.76	8	0.75	80	0.81	33	0.8	28	0.75	12	0.76	18
9	492	2351	0.24	0.8	51	0.84	32	0.75	5	0.8	69	0.82	34	0.81	16	0.85	8	0.62	2351
10	482	2395	0.42	0.76	56	0.77	42	0.78	12	0.68	140	0.74	76	0.73	45	0.81	2	0.83	20
11	548	2424	0.39	0.76	43	0.78	25	0.74	8	0.71	85	0.77	34	0.73	24	0.79	2	0.81	20
12	539	2416	0.41	0.77	62	0.78	45	0.78	23	0.71	100	0.77	47	0.75	31	0.76	18	0.77	12
13	496	2395	0.19	0.87	40	0.87	26	0.79	5	0.84	52	0.85	25	0.8	6	0.63	2395	0.91	12
14	549	2375	0.2	0.78	86	0.85	51	0.83	9	0.79	100	0.84	35	0.85	11	0.85	3	0.85	15
15	506	2414	0.23	0.82	39	0.84	17	0.83	5	0.76	92	0.8	30	0.74	5	0.84	8	0.86	11
16	482	2413	0.25	0.8	68	0.84	42	0.67	2	0.78	120	0.82	34	0.83	13	0.82	7	0.82	13
17	547	2375	0.17	0.83	62	0.89	35	0.89	7	0.79	54	0.85	19	0.85	4	0.85	11	0.84	5
18	553	2397	0.33	0.8	56	0.81	33	0.81	6	0.73	140	0.8	44	0.79	10	0.82	16	0.82	12
19	530	2387	0.41	0.76	61	0.76	40	0.73	11	0.72	100	0.77	43	0.76	36	0.75	7	0.82	19
20	500	2327	0.1	0.83	44	0.91	16	0.9	8	0.87	18	0.87	14	0.89	6	0.87	11	0.86	7
21	517	2392	0.36	0.75	62	0.8	39	0.75	10	0.67	120	0.8	38	0.77	32	0.52	2392	0.8	20
22	550	2354	0.26	0.84	42	0.86	28	0.85	5	0.83	49	0.85	32	0.85	7	0.53	2354	0.86	10
23	543	2394	0.36	0.77	56	0.78	34	0.78	9	0.73	87	0.76	37	0.76	27	0.75	6	0.77	17
24	540	2396	0.16	0.82	56	0.88	22	0.84	3	0.85	37	0.87	21	0.87	2	0.52	2396	0.6	2396
25	511	2417	0.52	0.71	60	0.73	43	0.65	6	0.69	140	0.72	78	0.71	21	0.74	16	0.74	16
26	504	2282	0.13	0.77	54	0.83	29	0.83	8	0.67	92	0.84	32	0.77	16	0.79	20	0.73	2300
27	516	2413	0.33	0.79	40	0.8	27	0.8	16	0.73	120	0.79	54	0.79	34	0.78	11	0.78	18
28	544	2428	0.17	0.86	43	0.88	31	0.84	5	0.78	99	0.82	47	0.83	8	0.89	18	0.86	11
29	536	2415	0.25	0.81	52	0.83	26	0.78	13	0.76	76	0.83	26	0.79	26	0.58	2415	0.84	18

Table 3.9: Results obtained with M = *elastic net* – part 2 of 2

			complexity = considered							complexity = not considered						random forest		bagged trees	
			1sd clust.	no no		yes no		no yes		no no		yes no		no yes					
	nrow	ncol	pos	AUC	n	AUC	n	AUC	n	AUC	n	AUC	n	AUC	n	AUC	n	AUC	n
30	575	2399	0.54	0.71	56	0.72	36	0.7	11	0.66	180	0.68	75	0.68	65	0.69	11	0.69	11
31	578	2375	0.27	0.82	45	0.83	27	0.77	7	0.81	33	0.82	18	0.82	8	0.83	3	0.83	14
32	503	2319	0.17	0.83	67	0.84	48	0.85	4	0.81	48	0.83	11	0.83	11	0.76	7	0.76	4
33	540	2388	0.26	0.83	52	0.85	30	0.84	10	0.76	84	0.83	49	0.83	16	0.81	17	0.81	20
34	555	2315	0.12	0.8	35	0.86	20	0.85	7	0.77	60	0.8	21	0.82	25	0.81	14	0.87	12
35	517	2348	0.15	0.86	40	0.87	22	0.84	4	0.75	65	0.87	25	0.85	13	0.79	2	0.81	12
36	548	2410	0.16	0.84	48	0.86	26	0.88	6	0.77	85	0.87	22	0.86	6	0.77	7	0.79	12
37	492	2366	0.16	0.83	50	0.88	26	0.86	8	0.8	58	0.86	17	0.86	9	0.83	7	0.85	14
38	504	2410	0.63	0.67	64	0.67	33	0.67	11	0.64	160	0.67	44	0.65	32	0.67	20	0.67	14
39	500	2396	0.29	0.82	48	0.83	30	0.81	5	0.8	48	0.82	29	0.83	22	0.72	18	0.75	14
40	537	2421	0.24	0.82	44	0.83	23	0.76	4	0.72	90	0.82	28	0.77	11	0.58	2421	0.82	14
41	510	2418	0.33	0.78	58	0.79	32	0.74	5	0.76	140	0.75	62	0.74	16	0.5	2418	0.55	2418
42	535	2423	0.3	0.8	52	0.79	32	0.8	6	0.75	100	0.79	48	0.78	33	0.8	6	0.82	16
43	501	2418	0.19	0.8	60	0.88	34	0.82	6	0.83	46	0.83	19	0.85	10	0.85	4	0.87	17
44	536	2390	0.37	0.8	55	0.81	38	0.67	4	0.78	92	0.8	38	0.76	9	0.77	19	0.8	15
45	534	2397	0.42	0.76	62	0.78	30	0.71	6	0.71	98	0.77	42	0.73	25	0.81	9	0.8	17
46	531	2420	0.54	0.7	75	0.73	49	0.71	21	0.67	170	0.71	62	0.69	74	0.53	2420	0.72	16
47	540	2371	0.25	0.82	45	0.83	30	0.77	4	0.8	60	0.82	37	0.8	8	0.82	16	0.83	14
48	496	2412	0.58	0.68	70	0.7	48	0.7	18	0.66	140	0.68	70	0.6	70	0.68	12	0.67	18
49	505	2391	0.4	0.76	47	0.76	33	0.76	9	0.74	79	0.76	44	0.76	26	0.8	14	0.8	13
50	497	2300	0.24	0.84	49	0.87	24	0.83	6	0.82	56	0.87	32	0.85	13	0.4	2300	0.58	2300
51	527	2404	0.34	0.78	58	0.82	42	0.82	12	0.77	110	0.82	44	0.76	14	0.81	14	0.78	20
52	510	2383	0.2	0.82	53	0.89	36	0.79	6	0.78	94	0.86	44	0.87	14	0.57	2383	0.57	2383
53	577	2300	0.13	0.78	62	0.89	28	0.69	5	0.85	26	0.86	20	0.88	4	0.63	2300	0.94	20
54	528	2389	0.33	0.8	49	0.81	30	0.71	8	0.73	82	0.81	40	0.75	11	0.6	2389	0.84	10
55	494	2321	0.12	0.83	54	0.88	22	0.79	4	0.75	110	0.89	30	0.85	10	0.97	14	0.95	14
avg	524	2384	0.29	0.8	54	0.82	32	0.78	8	0.75	89	0.81	38	0.79	19	0.75	530	0.79	310
sd	24	37	0.13	0.044	10	0.052	8	0.06	4	0.055	35	0.052	16	0.058	16	0.12	980	0.1	780
computation time (sec.)				236		236		885		430		1614		67		63		826	

Chapter 4

The tradeoff between forecast accuracy and spatiotemporal resolution for EMS call volumes

4.1 Introduction

Optimal management of an Emergency Medical Services (EMS) system is of paramount importance for the wellbeing of the citizens. With increasing demand for such services in most cities, EMS managers are finding it increasingly difficult to provide high-quality healthcare services with limited resources (Lowthian et al., 2011; Munjal et al., 2011). In a well-planned EMS system, the location of the ambulance stations, the staff and fleet size of stations, and the strategies for vehicle deployment have all been optimized in order to meet its performance targets. Such decisions require reliable demand forecasts (Aringhieri et al., 2017; Reuter-Oppermann et al., 2017; Ingolfsson, 2013). In this study, we propose methods for forecasting EMS call volumes in space and time. In evaluating forecasting methods, we pay special attention to three challenges.

First, for greater generalizability, we would like to test the performance of the methods we

employ on data from multiple cities. This is important because spatio-temporal patterns of EMS calls vary with city characteristics such as its geography, its infrastructural layout, and the size and distribution of the population (Zhou et al., 2016). To address this challenge, we have obtained five years of data on EMS calls for three cities in Alberta, Canada: Calgary, Edmonton, and Grande Prairie—a larger and more diverse data set than has been used in past research. Table 4.1 compares the three cities. Calgary and Edmonton are the two largest cities in Alberta, each with a population exceeding 900 thousand. Grande Prairie is smaller, with a population of around 63 thousand. Calgary and Edmonton are geographically similar in that both have a river that runs through the city center (Figure 4.1). This diverse data set allows us to investigate such questions as whether a method that performs well in one city also performs well in other cities with similar or different characteristics.

Table 4.1: Population, area, and call volume for the cities in our sample. All values are for 2016.

City	Population	Area (km^2)	Pop. density (per km^2)	Total calls	Calls per capita	Number of stations
Calgary	1,239,220	825.56	1,501	99,104	1/12.5	27
Edmonton	932,550	685.25	1,361	89,286	1/10.4	16
Grande Prairie	63,166	132.72	476	4,573	1/13.8	1

Second, a critical decision when designing an EMS forecasting system is choosing the appropriate level of temporal and spatial granularity. From a technical point of view, increasing resolution, by predicting for shorter time intervals and smaller regions, results in fewer calls per interval/region which can lead to noisier forecasts. Decreasing resolution, on the other hand, could result in missing spatio-temporal patterns present at higher resolutions and would also result in aggregation error (Francis et al., 2009; Micheletti et al., 2010). In order to investigate this trade-off, we use three levels of temporal granularity: 4-hour, 8-hour, and 24-hour periods, and three levels of spatial granularity: station catchment areas, intermediate regions, and the whole city. This results in nine combinations, which we will use to train different models. These values were selected after interviewing EMS system managers

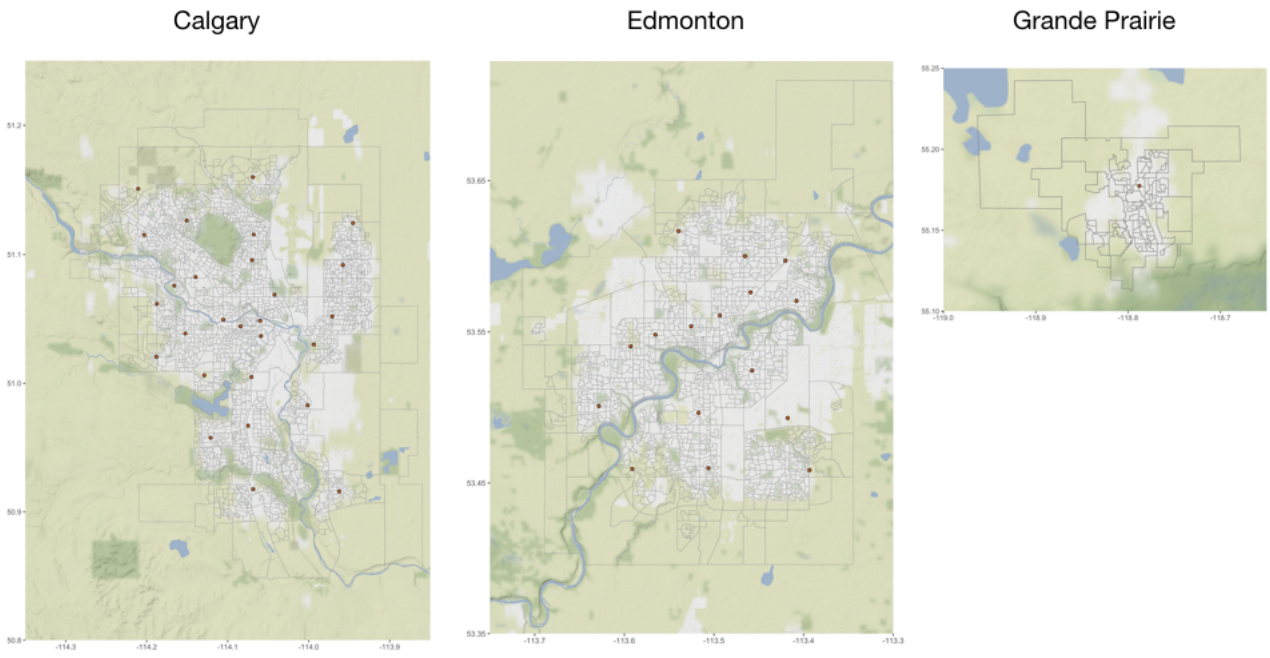

Figure 4.1: Map of Calgary, Edmonton, and Grande Prairie. Ambulance stations are shown as red circles. The polygons are dissemination areas with each having a population of 400 to 700 persons.

and discussing the ways in which forecasts might be used in practice. For example, in order to optimize the fleet size of each ambulance station within the city, we would need to know the level of demand for each station's catchment area. Having a higher spatial granularity, than catchment area-level, does not give us any further advantage in this case. Similarly, knowing the demand at 4-hour intervals would provide such managers sufficient information for making better decisions regarding staffing. Although some past research have used finer level of spatial granularity (Setzler et al., 2009), it is not clear how this finer granularity is useful for station-level deployment, although it might be useful for selecting station locations. The focus of this paper is on tactical decisions faced by EMS managers (e.g. staffing and ambulance relocation) as opposed to strategic decisions (e.g. station location) and therefore we do not consider finer levels of spatial granularity than station catchment areas.

Third, we will investigate the value that can be obtained from exploiting spatial autocorrelations, by jointly forecasting the call volumes of multiple regions within a city. Ye et al. (2019) have shown that, in the presence of inter-stream lag dependence, it is beneficial to

jointly forecast multiple time-series. After testing for such dependence, following a procedure proposed by the same authors, we simultaneously forecast call volumes using multivariate models and compare their performance against their univariate version.

Past research on EMS call volumes has mostly focused on forecasting volumes over time (Channouf et al., 2007; Matteson et al., 2011; Vile et al., 2012) and forecasting the location of calls has been relatively less studied. Setzler et al. (2009) use artificial neural networks to forecast call volumes at various spatial resolutions, using regions as small as 2×2 sq. mile regions. They observe that with high spatial resolution, many regions have zero calls in most time intervals, which complicates forecasting. Zhou et al. (2015) show that call volumes from Toronto, Canada exhibit spatial autocorrelation and propose a time-varying Gaussian mixture model for spatio-temporal forecasting. They claim that their proposed model is more suitable for sparse data such as EMS call volumes. Zhou et al. (2016) use a kernel warping method for forecasting EMS calls in Melbourne, Australia, which outperforms a time-varying Gaussian mixture model on their data set.

Regarding the trade-off between forecast resolution and accuracy, Huddleston et al. (2015) demonstrated that "top down" forecasting methods, which first forecast on a lower spatial resolution and then split the forecasts into ones for smaller regions, outperform "bottom up" methods that aggregate forecasts made at higher resolutions. The different levels of spatial and temporal resolution used in this paper allow us to conduct similar experiments. In some cases we employ methods that are hybrid, in the sense that forecasting is top down with respect to space but bottom up with respect to time, or vice versa.

Past studies have shown that multivariate exponential smoothing models are suited for simultaneous forecast of multiple time series (Hyndman et al., 2008; De Silva et al., 2010; Pfeffermann and Allon, 1989). While some advocate the use of multivariate models by showing an improvement in forecast accuracy (De Silva et al., 2010; Corberán-Vallet et al., 2011; Athanasopoulos and de Silva, 2012) others argue that multivariate models do not add value over univariate models (Du Preez and Witt, 2003; Feng and Shi, 2018). Other reasons,

beside forecasting accuracy, for using multivariate models include less computation time and ease of use (Bermúdez et al., 2009). The models of the mentioned papers, however, were trained on tourism data or economic data and to the best of our knowledge there has not been any empirical research on the suitability of multivariate exponential smoothing models for forecasting EMS call volumes.

4.2 Data description and preprocessing

We have data on the time and anonymized location of every EMS call received over 5 years, for Calgary and Edmonton, and over 4 years and 2 months, for Grande Prairie. Table 4.2 lists the number of calls received each year in each city.

Table 4.2: EMS calls by city and year

City	2012 Apr-Dec	2013 Jan-Dec	2014 Jan-Dec	2015 Jan-Dec	2016 Jan-Dec	2017 Jan-Mar	Total
Calgary	64,744	89,551	93,994	95,843	99,104	25,199	468,435
Edmonton	58,283	80,359	84,263	87,754	89,286	22,613	422,558
Grande Prairie		Feb-Dec 4,005	4,629	4,777	4,573	1,253	19,237

4.2.1 Aggregating the data

We train models with three levels of temporal resolution: 4, 8, and 24 hour intervals and three levels of spatial resolution: station catchment area, intermediate regions, and city-wide. Grande Prairie has only one ambulance station, and therefore we use only one spatial resolution for that city.

For a temporal resolution δ, we divide the day into equally spaced intervals, starting at midnight, of length δ and count the number of calls in each interval. For example, for an 8-hr temporal resolution, we have the following intervals: [00:00, 8:00), [8:00, 16:00), [16:00, 24:00).

For spatial aggregation of calls, we first identify the dissemination areas (DAs; the smallest geographical areas in Canada for which full census data is available, Statistics Canada, 2018) within a region and then aggregate the call volumes of DAs.

4.2.2 Constructing station catchment areas

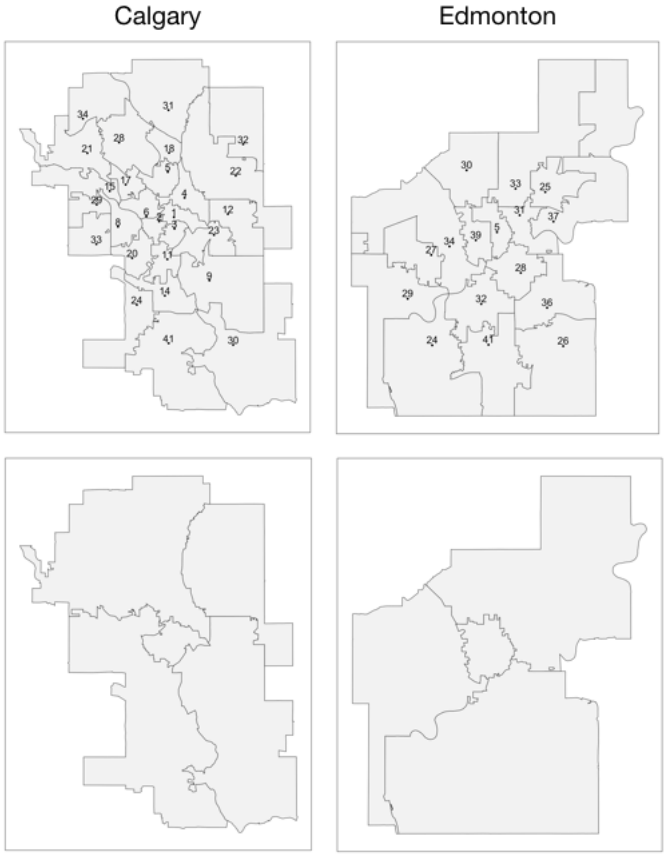

Figure 4.2: Top: Ambulance station catchment areas (numbers indicate station locations); Bottom: Intermediate regions

We construct station catchment areas (Figure 4.2, top) by assigning each DA to an ambulance station. We calculate the center of a DA as the population-weighted average of the dissemination blocks within that DA (A dissemination block is the smallest geographic area in Canada for which population figures are disseminated, Statistics Canada, 2018.) We use the Google Maps API to calculate the driving distance from all ambulance stations to

the DA's center and assign the DA to the closest ambulance station.

The intermediate regions (Figure 4.2, bottom) are constructed by combining station catchment areas, using geographical barriers and main roads and highways as boundaries.

4.3 Notation and preliminary data analysis

4.3.1 Notation

Let $y_{t,r}^{\delta,\alpha}$ be the call volume in Period t and Region r, using temporal resolution δ and spatial resolution α, where $t \in \{1, 2, \dots\}$ and $r \in R(\alpha)$. We focus on temporal resolutions of $\delta = 4, 8,$ and 24 hours, but we also use $\delta = 1$ and 168 in this section, to describe weekly patterns. The spatial resolution can be $\alpha = h, m,$ or l, short for *high* (station catchment areas), *medium* (intermediate regions), or *low* (whole city). The set $R(m)$, as an example, for Edmonton, is { Region 1, ..., Region 5 }. We omit the superscripts δ and α when they are not needed, so that the notation simplifies to $y_{t,r}$. We denote the call volume forecast for time $t + h$ made at time t as $\hat{y}_{t+h,r|t}$.

In order to transform forecasts at one resolution to forecasts at a higher or lower resolution we define three functions:

- $f(t, \delta)$ gives the time interval of t, for example, $f(1, 8) = [0, 8)$ week 1.
- $g(t, \delta)$ for $\delta \in \{4, 8\}$, gives a number from 1 to 21, specifying which 8-hour interval of the week t belongs to, with $1 = [0, 8)$, $2 = [8, 16)$, ..., $21 = [160, 168)$.
- $\omega(t, \delta)$ for $\delta \in \{4, 8, 24\}$ specifies the day of the week that t belongs to.

We use bold lower case for vectors and bold upper case for matrices, for example, $\mathbf{y}_t = (y_{t,r}, r \in R)$. We use a . subscript to indicate summation, that is, $y_{t.} = \sum_{r \in R} y_{t,r}$ and $y_{.r} = \sum_t y_{tr}$.

Finally, n is the total number of weeks in the smallest training set (fold 1) and $m(\alpha)$ is the number of regions at the α spatial resolution level.

4.3.2 Call volumes in time

Figure 4.3 shows the percentage of calls for the 168 hours of the week, starting at 0 am on Monday, for the three cities. The black curve represents the average among all weeks in the data and the grey shades represent 95% confidence intervals. The procedure for creating these plots is:

- Calculate the proportion of calls in hour h of week j:

$$p_{j,h} = \frac{y^{1,l}_{168(j-1)+h.}}{y^{168,l}_{j.}}, \quad h = 1, ..., 168, \quad j = 1, ..., n$$

- Calculate the sample mean and sample variance of the $p_{j,h}$s for each hour of the week:

$$\mu_h = \frac{1}{n}\sum_{j=1}^{n} p_{j,h}; \quad \sigma_h^2 = \frac{1}{n-1}\sum_{j=1}^{n}(p_{j,h} - \mu_h)^2$$

- Calculate confidence interval half-width for each hour of the week: $t_{n-1}\sqrt{\frac{\sigma_h^2}{n}}$, where t_{n-1} is the t-score, with $n-1$ degrees of freedom, at 95% confidence level.

Grande Prairie, being a smaller city, has a noisier curve with wider confidence intervals than the other two cities. Calgary and Edmonton are similar with the former having slightly higher peaks and lower valleys.

Call volumes start to increase around 7 am and stay high until evening. They then fall and reach their minimum around 2 am. The temporal patterns for the first four days of the week (Monday to Thursday) are quite similar and different from the weekends. In particular, call volumes on Fridays and Saturdays stay high for a longer period of time. This pattern is typical for EMS call volumes (Ingolfsson, 2013).

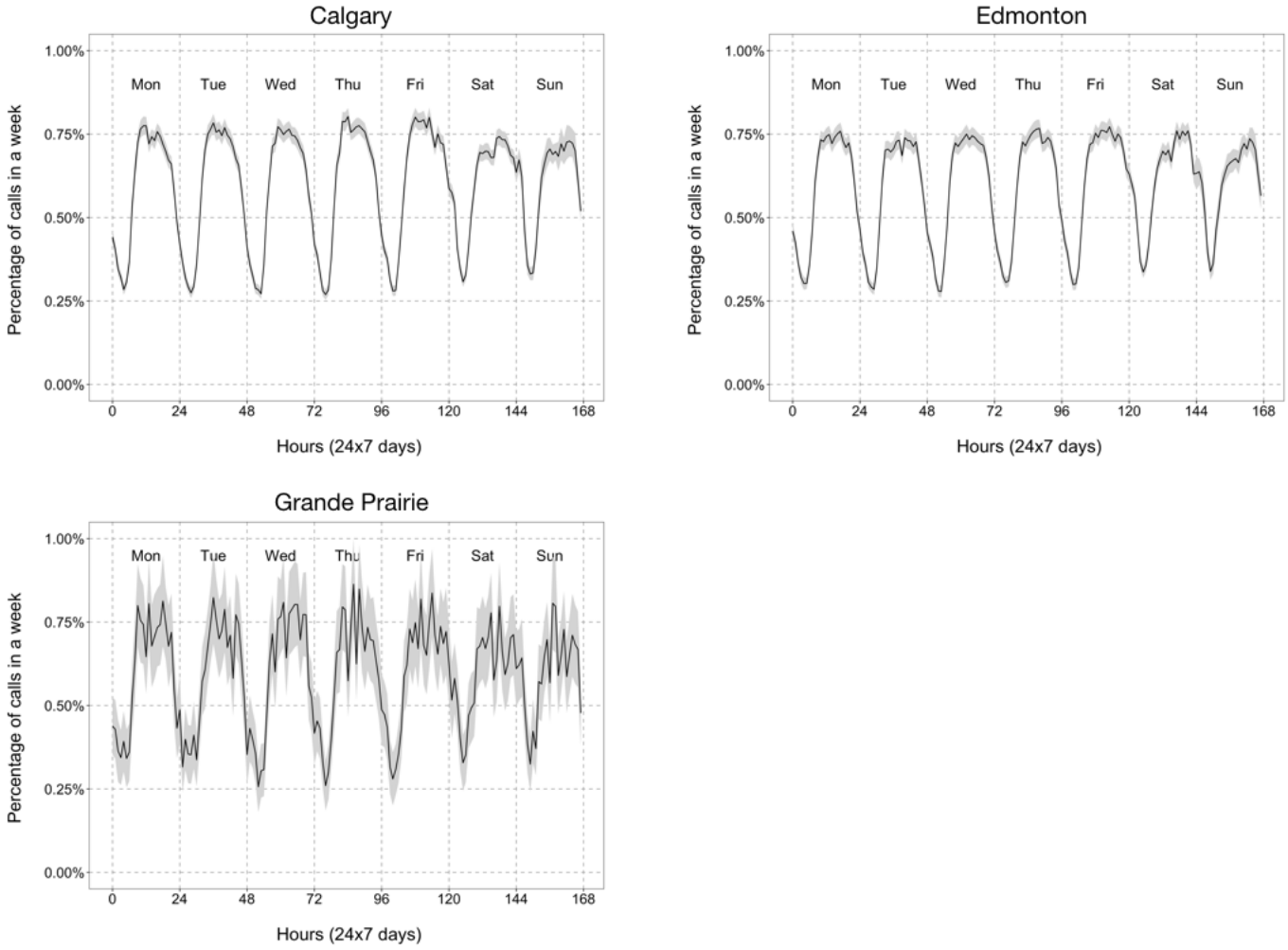

Figure 4.3: Temporal pattern of call volumes in a week.

4.3.3 Call volumes in space

Figure 4.4 shows the spatial distribution of calls, with each DA coloured based on the number of calls per km^2 in 2016. The central business district has by far the highest concentration of calls in all three cities.

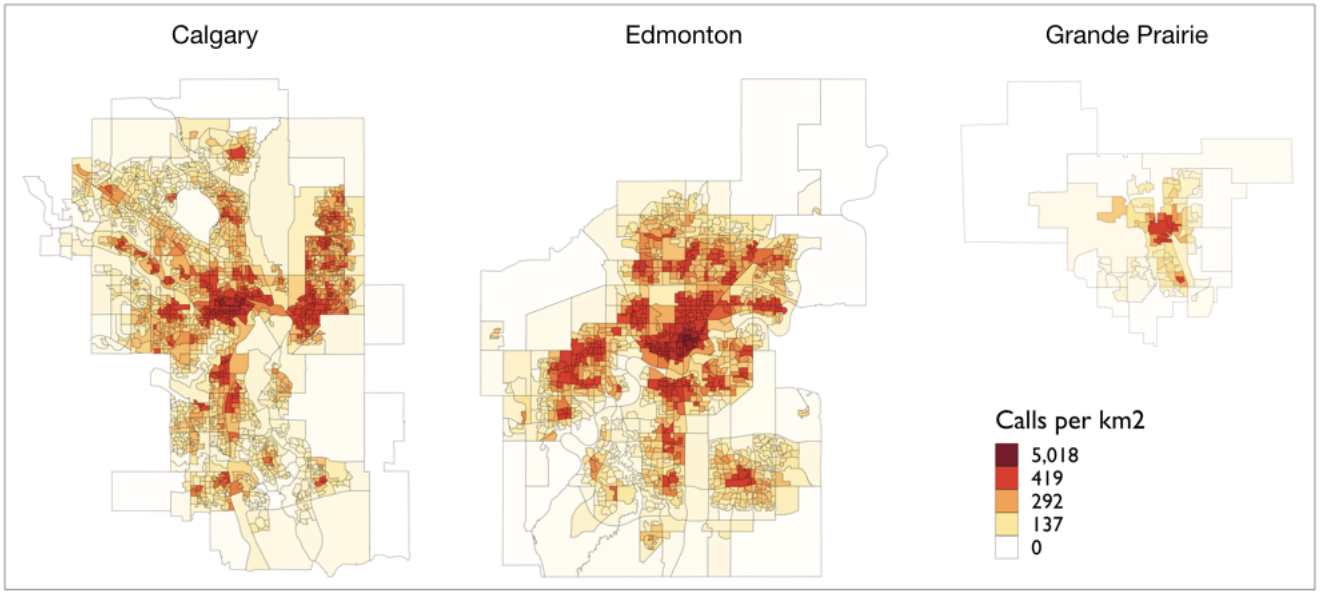

Figure 4.4: Spatial pattern of call volumes for 2016.

4.3.4 Call volumes in time and space

The spatial patterns and the temporal patterns interact, as we illustrate in Figure 4.5, for Calgary. Although the intermediate regions have similar weekly patterns, we observe some systematic differences. The most pronounced difference is the elevated call volumes after midnight on Friday and Saturday in Region 1 (the central business district). We also see an earlier and higher peak on weekdays in Region 5 than for the city as a whole.

4.3.5 Test for inter-stream lag dependence

Ye et al. (2019) show that it is beneficial to jointly forecast multiple time series when there is inter-stream lag dependence and propose a statistical test for such dependence. Following their proposed procedure, we take the following steps to test for the existence of inter-stream lag dependence among call volumes of a city's regions:

1. Calculate the daily square-root transformed call volume of Region r: $q_{t,r} = \sqrt{y_{t,r}^{24,m}}$

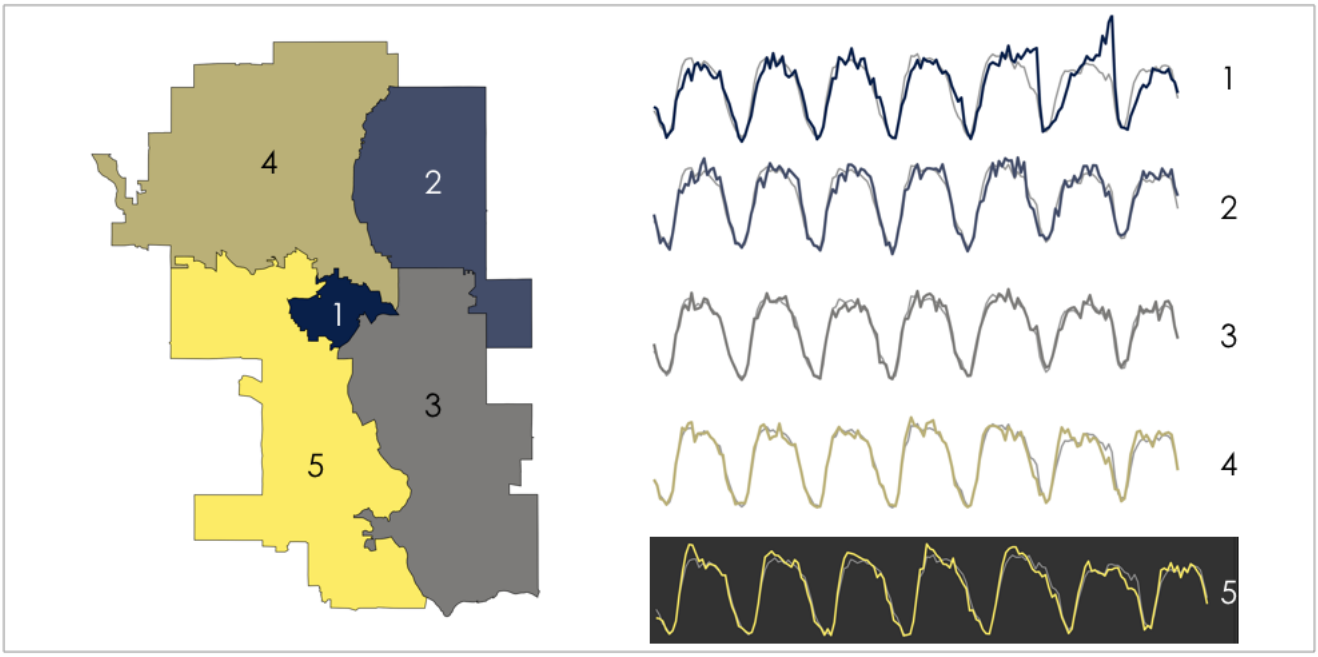

Figure 4.5: Interaction between spatial and temporal patterns, for Calgary. The graphs on the right show the weekly temporal pattern for each intermediate region (bold lines), superimposed on the weekly temporal pattern for the city as a whole (gray lines).

2. Calculate the average daily call volume for each day of the week: $\hat{\beta}_{d,r} = \frac{1}{|\{t:\omega(t,24)=d\}|} \sum_{t:\omega(t,24)=d} q_{t,r}$, $d = 1, \ldots, 7$.

3. Define the vectors $\mathbf{q}_t = (q_{t,r}, r \in R(m))$ and $\hat{\beta}_t = (\hat{\beta}_{\omega(t,24),r}, r \in R(m))$.

4. Fit the following multivariate time series model using the R package *vars* (Pfaff, 2008) and estimate the coefficient matrix $\mathbf{D}$, and the variance-covariance matrix $\tilde{\mathbf{\Omega}}$:

$$\mathbf{q}_t - \hat{\beta}_t = \mathbf{D}(\mathbf{q}_{t-1} - \hat{\beta}_{t-1}) + \mathbf{v}_t, \quad \mathbf{v}_t \sim N(0, \tilde{\mathbf{\Omega}}) \tag{4.1}$$

5. Perform a hypothesis test for the parameters in the coefficient matrix $\mathbf{D} = (d_{ij})$ (Tsay, 2013).

A statistically significant coefficient d_{ij} indicates the existence of inter-stream one-day lag dependence of Region i on Region j. Tables 4.3 and 4.4 show the estimated values of the coefficent matrix $\mathbf{D}$ for Calgary and Edmonton, respectively, at the intermediate region

spatial resolution. The majority of the coefficients are statistically significant at the 0.05 level (19 out of 25 for Calgary, 15 out of 16 for Edmonton) after accounting for multiple comparisons by using a Bonferroni correction. All of the estimated coefficients are positive, indicating that the call volume on Day t in Region r has a positive association with the call volumes on Day $t-1$ in most of the regions in the city. A multivariate Jarque-Bera test and multivariate skewness and kurtosis tests for normality of the residuals in Equation 4.1 was performed. The results indicate that the residuals of both the Calgary and Edmonton models follow a normal distribution. The same inter-stream one-day lag dependence test was conducted for the station catchment area spatial resolution level and the results can be seen in appendix 4.10.2. The number of statistically significant coefficients at this spatial resolution is much lower, with only 17 out of 729 and 8 out of 256 coefficients being significant at the 0.05 Bonferroni-adjusted level for Calgary and Edmonton, respectively. All of the significant coefficients are positive, and about half of them (8 out of 17 for Calgary, 4 out of 8 for Edmonton) correspond to the call volume in a station catchment area on one day being positively associated with the call volume in the same station catchment area on the previous day.

Table 4.3: Estimated d_{ij} coefficients for Calgary—$\alpha = m$. 19 out of 25 coefficients are statistically significant at the 0.05 level after accounting for multiple comparisons by using Bonferroni correction.

	Region 1	Region 2	Region 3	Region 4	Region 5
Region 1	0.267 *	0.094 *	0.018	0.082 *	0.079
Region 2	0.075 *	0.076 *	0.076 *	0.109 *	0.095 *
Region 3	0.042	0.046	0.088 *	0.104 *	0.114 *
Region 4	0.048	0.088 *	0.119 *	0.153 *	0.121 *
Region 5	0.087 *	0.082 *	0.059	0.138 *	0.170 *

* $p < 0.002$ (Bonferroni adjusted significance level)

Table 4.4: Estimated d_{ij} coefficients for Edmonton—$\alpha = m$. 15 out of 16 coefficients are statistically significant at the 0.05 level after accounting for multiple comparisons by using Bonferroni correction.

	Region 1	Region 2	Region 3	Region 4
Region 1	0.230 *	0.078 *	0.080 *	0.091 *
Region 2	0.074 *	0.195 *	0.105 *	0.157 *
Region 3	0.069 *	0.068 *	0.075 *	0.105 *
Region 4	0.106 *	0.177 *	0.049	0.200 *

* $p < 0.0031$ (Bonferroni adjusted significance level)

4.4 Models, Methods, and Estimation

All data analysis and model building was done in the R statistical language (R Core Team, 2013).

4.4.1 Univariate models

In this subsection we will review the univariate exponential smoothing methods that we use, and a generalization of exponential smoothing called TBATS. Innovations state-space models provide a unified way to formulate models corresponding to all of the univariate forecasting methods that we compare. Following Hyndman et al. (2008) we distinguish between models and methods, using *model* for equations that provide an entire distribution of forecasts and *method* for equations or algorithms that only give point forecasts.

Innovations state-space models and exponential smoothing:

Let $\mathbf{x}_{t,r}$ be a $k \times 1$ state vector for Region r at time t. The governing equations of a linear innovations state-space model are:

$$y_{t,r} = \mathbf{w}_r^T \mathbf{x}_{t-1,r} + \epsilon_{t,r}, \tag{4.2}$$

$$\mathbf{x}_{t,r} = \mathbf{F}_r \mathbf{x}_{t-1,r} + \mathbf{g}_r \epsilon_{t,r}, \tag{4.3}$$

where $\epsilon_{t,r}$ is white noise; $\mathbf{x}_{t,r}$ is the $k \times 1$ state vector; and the $k \times k$ matrix $\mathbf{F}_r$ and the $k \times 1$ vectors $\mathbf{g}_r$ and $\mathbf{w}_r$ contain coefficients. The *measurement equation* (4.2) specifies how the state vector translates into a call volume. The *transition equation* (4.3) describes how the state vector evolves. Most exponential smoothing methods can be derived from one or more (linear or nonlinear) innovations state space models. For example, consider the additive-seasonality variant of the Holt-Winters (HW-A for short) method (Holt, 1957; Winters, 1960), whose point forecasts are generated as follows:

$$\begin{aligned} l_{t,r} &= \alpha_r(y_{t,r} - s_{t-m,r}) + (1-\alpha_r)(l_{t-1,r} + b_{t-1,r}), \\ b_{t,r} &= \beta_r(l_{t,r} - l_{t-1,r}) + (1-\beta_r)b_{t-1,r}, \\ s_{t,r} &= \gamma_r(y_{t,r} - l_{t-1,r} - b_{t-1,r}) + (1-\gamma_r)s_{t-m,r}, \\ \hat{y}_{t+h,r|t} &= l_{t,r} + b_{t,r}h + s_{t-m+h_m^+,r}, \end{aligned} \tag{4.4}$$

where $l_{t,r}$, $b_{t,r}$, and $s_{t,r}$ are the level, trend, and seasonality at time t; m is the length of the seasonal cycle; $h_m^+ = [(h-1) \mod m]+1$; and α_r, β_r, and γ_r are smoothing parameters. The HW-A forecasting equations can be derived from a linear innovations state space model of the form shown in (1)-(2) and also from a nonlinear innovations state space model in which the error term ϵ_t is multiplied with the state vector $\mathbf{x}_{t-1}$.

Hyndman et al. (2008) introduced a unified notation for innovations state-space models associated with exponential smoothing methods, where the triple (E,T,S) identifies the type

of error, trend, and seasonality. Table 4.5 uses ETS notation to list the innovations state space models for the 15 univariate exponential smoothing methods that we compare. We used the smooth package (Svetunkov, 2018) in R to estimate these models. We limit ourselves to models with additive error. All 15 models have variants with multiplicative error, but these models result in equations of identical form for computing point forecasts. Given that we use out-of-sample forecast errors to compare methods (rather than in-sample methods that require likelihood values), we gain nothing by including multiplicative error models. Furthermore, multiplicative error models are numerically unstable when the data contain zeros. The class of innovations state-space models includes models for all exponential smoothing methods and models equivalent to all ARIMA models (Hyndman et al., 2008).

Table 4.5: Exponential smoothing models that we compare. N = none, A = additive, M = multiplicative, d = damped.

	Seasonal		
Trend	N	A	M
N	ETS(A,N,N)	ETS(A,N,A)	ETS(A,N,M)
A	ETS(A,A,N)	ETS(A,A,A)	ETS(A,A,M)
A_d	ETS(A,A_d,N)	ETS(A,A_d,A)	ETS(A,A_d,M)
M	ETS(A,M,N)	ETS(A,M,A)	ETS(A,M,M)
M_d	ETS(A,M_d,N)	ETS(A,M_d,A)	ETS(A,M_d,M)

TBATS:

De Livera et al. (2011) proposed an innovations state-space model for complex seasonal time series called TBATS, which generalizes the ETS(A,A_d,A) model. TBATS is an acronym for the main features in the model: Trigonometric, Box-Cox transform, ARMA errors, Trend, and Seasonal components. The seasonal component is modeled as a sum of trigonometric functions, which reduces the number of parameters needed to represent complex seasonal patterns. The Box-Cox transformation includes the square-root transformation recommended by Ye et al. (2019). ARMA errors allow the model to capture more complex dependence among the values in the time series. We used the *forecast* package in R (Hyndman and

Khandakar, 2008) to estimate the TBATS model.

4.4.2 Multivariate models

Vector exponential smoothing (VES): The general form of a linear innovations state-space model for a vector time series $\mathbf{y}_t$ is (Hyndman et al., 2008; De Silva et al., 2010):

$$\mathbf{y}_t = \mathbf{W}\mathbf{x}_{t-1} + \epsilon_t, \tag{4.5}$$

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{G}\epsilon_t, \tag{4.6}$$

where $\mathbf{y}_t$ is an $m(\alpha)$-vector of observations of the series at time t, $\mathbf{x}_t$ is the state vector, ϵ_t is an $m(\alpha)$-vector of error terms (one for each series), $\mathbf{W}$ and $\mathbf{F}$ are matrices with coefficients that are typically known, from the assumed model structure, and $\mathbf{G}$ is a matrix with entries that typically need to be estimated.

We compare two VES methods; one that can be derived from a linear innovations state space model and one that can be derived from a nonlinear innovations state space model. The set of equations corresponding to the linear innovations state space model (LVES) model is shown in the set of equations in (4.7). In this formulation of VES, series do not share any state value. Furthermore, when the off-diagonal elements of $\mathbf{G}$ are zero the innovations of each series would only have an effect on its own process but with non-zero off-diagonal

elements the innovation of a series would also affect other series.

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{x}_{t,1} \\ \vdots \\ \mathbf{x}_{t,m(\alpha)} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \mathbf{w}_1^T & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{w}_{m(\alpha)}^T \end{bmatrix}, \tag{4.7}$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{F}_{m(\alpha)} \end{bmatrix}, \quad \text{and} \quad \mathbf{G} = \begin{bmatrix} g_{11} & \cdots & g_{1m(\alpha)} \\ g_{21} & \cdots & g_{2m(\alpha)} \\ & \cdots & \\ g_{m(\alpha)1} & \cdots & g_{m(\alpha)m(\alpha)} \end{bmatrix}.$$

Here $\mathbf{x}_{t,1}$ is a $k \times 1$ vector of state values for region 1 and, thus, $\mathbf{x}_t$ is a $(m(\alpha)k) \times 1$ vector of state values. $\mathbf{w}_1$ is a $k \times 1$ vector and $\mathbf{W}$ is an $(m(\alpha)k) \times (m(\alpha)k)$ matrix. $\mathbf{F}_1$ is $k \times k$ transition matrix for region 1 and $\mathbf{F}$ is an $(m(\alpha)k) \times (m(\alpha)k)$ matrix. We used the *smooth* package (Svetunkov, 2018) in R to estimate the LVES model.

Proportional exponential smoothing:

We propose an exponential smoothing model suitable for simultaneous forecast of multiple time series. We assume that at each time period, t, there exists a latent state value, l_t, which defines the aggregate level of calls and b_t defines the growth of this state variable. The level of a single stream, r, at time t is determined as a proportion, $p_{t,r}$, of the aggregate level l_t. Each series can furthermore have its own seasonality pattern. These assumptions are suitable, for example, for competitive products with different market shares, which can change with time. In the case of EMS call volumes, it is clearly the case that the city level affects the individual levels of the regions. Furthermore, because of city developments and long-term seasonal patterns the level of each region, as a proportion of the city level, could change over time. The state-space equations of the nonlinear innovations state space model

(NVES) corresponding to the proportional exponential smoothing model are:

$$
\begin{aligned}
y_{t,r} &= p_{t-1,r}(l_{t-1} + b_{t-1}) + s_{t-m,r} + (l_{t-1} + b_{t-1})\epsilon_t, && \text{for } r = 1, 2, \ldots, m(\alpha) \qquad (4.8)\\
l_t &= l_{t-1} + b_{t-1} + \alpha\epsilon_t, \\
b_t &= b_{t-1} + \beta\epsilon_t, \\
p_{t,r} &= p_{t-1,r} + \lambda\epsilon_t, && \text{for } r = 1, 2, \ldots, m(\alpha) \\
s_{t,r} &= s_{t-m,r} + \gamma\epsilon_t, && \text{for } r = 1, 2, \ldots, m(\alpha) \\
\sum_{r=1}^{m(\alpha)} p_{t,r} &= 1
\end{aligned}
$$

We can obtain update equations by substituting for the error term:

$$
\begin{aligned}
l_t &= \left(\frac{\alpha}{n(l_{t-1} + b_{t-1})}\right)(y_{t,\cdot} - s_{t-m,\cdot}) + \left(1 - \frac{\alpha}{n(l_{t-1} + b_{t-1})}\right)(l_{t-1} + b_{t-1}), && (4.9)\\
b_t &= \left(\frac{\beta}{\alpha}\right)(l_t - l_{t-1}) + \left(1 - \frac{\beta}{\alpha}\right)(b_{t-1}), \\
p_{t,r} &= (\lambda)\left(\frac{y_{t,r} - s_{t-m,r}}{l_{t-1} + b_{t-1}}\right) + (1 - \lambda)(p_{t-1,r}), && \text{for } r = 1, 2, \ldots, m(\alpha) \\
s_{t,r} &= \left(\frac{\gamma}{l_{t-1} + b_{t-1}}\right)(y_{t,r} - p_{t-1,r}(l_{t-1} + b_{t-1})) + \left(1 - \frac{\gamma}{l_{t-1} + b_{t-1}}\right)(s_{t-m,r}), && \text{for } r = 1, 2, \ldots, m(\alpha)
\end{aligned}
$$

where we have imposed the condition $\sum_{r=1}^{m(\alpha)} p_{t,r} = 1$ by using the equation:

$$y_{t,\cdot} = l_{t-1} + b_{t-1} + s_{t-m,\cdot} + n(l_{t-1} + b_{t-1})\epsilon_t, \qquad (4.10)$$

in (4.9).

We can write the nonlinear innovations state-space models in the form:

$$\mathbf{y}_t = w(\mathbf{x}_{t-1}) + r(\mathbf{x}_{t-1})\epsilon_t, \qquad (4.11)$$

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{g}\epsilon_t, \qquad (4.12)$$

where:

- $\mathbf{y}_t = (y_{t,1}, \ldots, y_{t,m(\alpha)})^T$ is an $m(\alpha)$-vector of observations,
- $\mathbf{x}_t = (l_t, b_t, p_{t,1}, \ldots, p_{t,m(\alpha)}, s_{t,1}, \ldots, s_{t-m+1,1}, s_{t,2}, \ldots, s_{t-m+1,m(\alpha)})^T$ is a $(2 + m(\alpha)(1 + m))$-vector of state values,
- ϵ_t is a scalar value for the error at time t,
- $w(\mathbf{x}_{t-1})$ is a nonlinear function of $\mathbf{x}_{t-1}$,
- $r(\mathbf{x}_{t-1})$ and $f(\mathbf{x}_{t-1})$ are linear functions of $\mathbf{x}_{t-1}$, and
- $\mathbf{g}$ is a vector containing the smoothing parameters.

The first author developed R code (available upon request) to formulate and estimate the PES using the *rcpp* package in R (Eddelbuettel and François, 2011). Optimization was done by choosing selecting smoothing parameters, from a grid of 10 equally-spaced values from 0 to 0.2, that minimize the one-step mean squared error.

4.5 Model evaluation

4.5.1 Evaluation metrics

We use the Mean Absolute Error (MAE) and Mean Absolute Scaled Error (MASE) as measures of forecast performance. Given a sequence of out-of-sample errors, $e_1, e_2, \ldots, e_h$, the MAE is:

$$\text{MAE} = \frac{\sum_{i=1}^{h} |e_i|}{h}.$$

The MASE was proposed by Hyndman and Koehler (2006) as a scale-free measure of error that is defined even in the presence of zero values (unlike the Mean Absolute Percentage

Error metric.) In order to calculate the MASE we first calculate the in-sample error of a naïve forecast method that forecasts the call volume at time t, y_t, as the call volume at time $t-m$, y_{t-m}, where m is the length of the seasonal pattern. Our definition of the Naïve method is different than Hyndman and Koehler (2006), where the call volume at time $t-1$ is used for forecasting the call volume at time t:

$$\text{in-sample error of naïve method} = \frac{1}{\tau - m} \sum_{j=m+1}^{\tau} |y_j - y_{j-m}|,$$

where τ is the number of observations in the training set. We scale each out-of-sample error in a sequence, $e_1, e_2, \ldots, e_h$, by dividing it by the in-sample error of naïve method:

$$q_i = \frac{e_i}{\frac{1}{\tau-m}\sum_{j=m+1}^{\tau} |y_j - y_{j-m}|},$$

A scaled-error, q_i, less than one indicates that the error is less than the average in-sample error of a naïve forecast method. We calculate MASE as:

$$\text{MASE} = \frac{\sum_{i=1}^{h} |q_i|}{h}$$

4.5.2 Out-of-sample rolling forecast

We use out-of-sample forecast accuracy rather than in-sample measures like AIC, because we have a sufficinently large amount of data (Hyndman et al., 2008). The out-of-sample rolling forecast (Fan and Yao, 2008) procedure is illustrated in Figure 4.6. Our first training set consists of the first 152 weeks of data (108 weeks for Grande Prairie), with the next 2 weeks serving as the test set. We extend the training set by 2 week and shift the test forward by 2 week and repeat that process until we run out of data. This process results in 50-fold out-of-sample validation. Letting $\text{MAE}_{f,m}$ and $\text{MASE}_{f,m}$ be the error measures for Fold f and Method m. We use the average error measures across the folds, $\text{MAE}_{\cdot m}$ and $\text{MASE}_{\cdot m}$,

to compare methods.

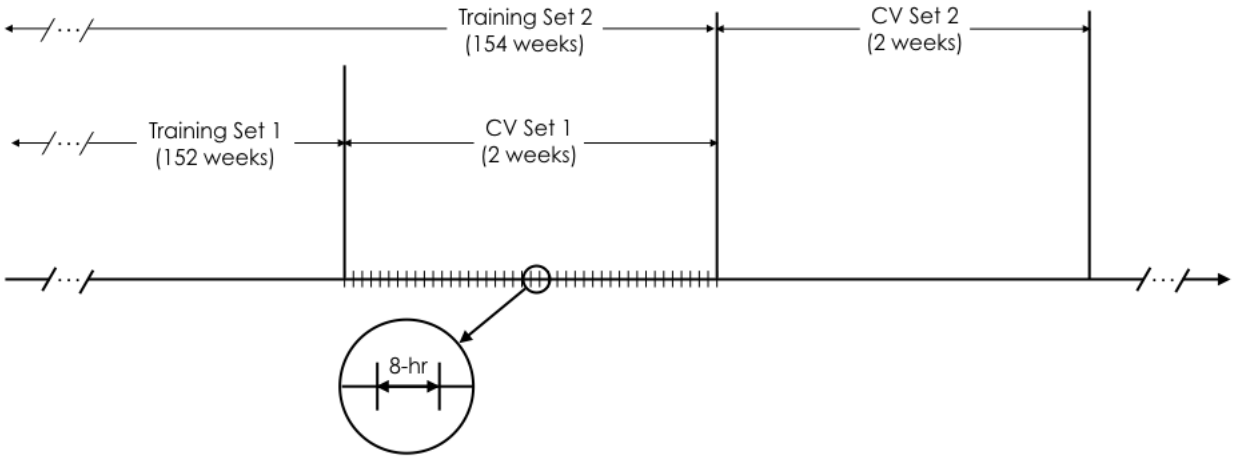

Figure 4.6: Folds for model evaluation.

4.5.3 Comparing models with different spatial and temporal resolution

We build models with 9 combinations of spatial and temporal resolutions, as shown in the first two columns of Table 4.6. We, however, measure the performance of all models on the same resolution, namely, catchment area and 8-hr resolution. We chose this level since it is well suited for operational management of EMS systems. In order to bring the forecasts of the models to the comparison level, we will either combine or divide their forecasts depending on whether the model used a higher or lower resolution, respectively. The aggregation/separation of call volumes is done independently in space and time.

4.5.4 Converting forecasts from higher to lower resolution

Bottom-up forecasting: We wish to combine forecasts for multiple regions or multiple time intervals to obtain a forecast for a larger region or longer time interval. We do this simply

Table 4.6: Spatial and temporal resolution levels.

Spatial resolution	Temporal resolution	Spatial forecast	Temporal forecast
catchment area	4-hr	same level	bottom-up
catchment area	**8-hr**	**same level**	**same level**
catchment area	24-hr	same level	top-down
intermediate region	4-hr	top-down	bottom-up
intermediate region	8-hr	top-down	same level
intermediate region	24-hr	top-down	top-down
city	4-hr	top-down	bottom-up
city	8-hr	top-down	same level
city	24-hr	top-down	top-down

by adding forecasts. For example, we add two 4-hour forecasts to obtain an 8-hour forecast:

$$\hat{y}_{t,r}^{8,\alpha} = \sum_{\tau: f(\tau,4) \in f(t,8)} \hat{y}_{\tau,r}^{4,\alpha}$$

4.5.5 Converting forecasts from lower to higher temporal resolution

Top-down forecasting: We wish to divide a forecast made at low resolution to multiple forecasts at a higher resolution. Our general approach is to estimate weights using the smallest training set (Fold 1) and use those weights to decompose forecasts. We begin by detailing how we decompose forecasts in time.

For dividing a forecast made on a 24-hour time interval to three forecasts on 8-hour intervals we use the following equation:

$$\hat{y}_{t,r}^{8,\alpha} = \omega^1(\rho)\hat{y}_{\tau,r}^{24,\alpha} \text{ for all } t : f(t,8) \in f(\tau,24),$$

where $\rho = g(t,8)$ is a number from 1 to 21, which specifies which 8-hour time interval of the week t belongs to. For example, $\omega^1(\rho)$ is the estimate of the percentage of call volumes in a

day to allocate to the 8-hour time interval ρ and is calculated as:

$$\omega^1(\rho) = \frac{N_\rho}{\sum_{j=1}^{21} N_j} \text{ for all } \rho = 1, \ldots, 21,$$

here N_ρ is the number of calls in the smallest training set (Fold 1) from the 8-hour period of the Week ρ.

4.5.6 Converting forecasts from lower to higher spatial resolution

We decompose $\hat{y}_{t,r}^{\delta,\alpha}$ into forecasts at a finer spatial resolution α', using weights $\omega(r,s)$, as follows:

$$\hat{y}_{t,s}^{\delta,\alpha'} = \omega^2(r,s)\hat{y}_{t,r}^{\delta,\alpha}, \text{ for all } s \in R(\alpha'), r \in R(\alpha),$$

where $\omega^2(r,s)$ is the estimate of the percentage of call volume from Region $r \in R(\alpha)$ to allocate to Region $s \in R(\alpha')$, using data from the smallest training set (fold 1) and is calculated as:

$$\omega^2(r,s) = \begin{cases} \frac{M_s}{M_r} \text{ if } s \subset r, \\ 0 \text{ otherwise} \end{cases},$$

where M_i is the total call volume from region i.

4.6 Model comparison

4.6.1 Comparison between methods with different components

A comparison between different univariate methods is shown in Tables 4.7–4.9. All the methods in the table have an 8-hour temporal and a catchment area spatial resolution. The performance is calculated for the one week ahead forecast of call volumes of the biggest catch-

ment area during Monday [8 am, 4 pm) interval. The best-performing method in Calgary and Grande Prairie, and the runner-up in Edmonton, based on both MAE and MASE, is the Holt-Winter additive seasonality method. The TBATS method performs slightly better in Edmonton. Removing the trend component has almost no impact on accuracy but removing the seasonality component reduces accuracy greatly. Exponential smoothing methods without seasonality perform worse than the naïve method, for Calgary and Edmonton. Exponential smoothing methods with seasonality perform up to 33% better than the naïve method. Using damped trend or multiplicative seasonality does not improve accuracy (Tables 4.21–4.23).

Table 4.7: One-week-ahead forecasting performance comparison between methods with different components $\delta = 8$ and $\alpha = h$—Calgary's station 1 catchment area. The mean and standard deviation of the calls over the 50 folds are 4.8 and 2.02, respectively.

City	Method	Trend	Seasonality	MAE	IQR	MASE	IQR
Cal.	Exp.S.	none	none	2.77	2.80	137%	139%
Cal.	Exp.S.	additive	none	2.85	2.97	141%	147%
Cal.	Exp.S.	none	additive	**1.64**	**1.65**	81.4%	**81.9%**
Cal.	Exp.S.	additive	additive	**1.64**	1.71	**81.3%**	84.6%
Cal.	TBATS	additive	additive	1.65	1.79	81.8%	88.7%

Table 4.8: One-week-ahead forecasting performance comparison between methods with different components for $\delta = 8$ and $\alpha = h$—Edmonton's station 5 catchment area. The mean and standard deviation of the calls over the 50 folds are 6.82 and 2.75, respectively.

City	Method	Trend	Seasonality	MAE	IQR	MASE	IQR
Edm.	Exp.S.	none	none	5.66	3.60	185%	118%
Edm.	Exp.S.	additive	none	5.75	3.77	188%	123%
Edm.	Exp.S.	none	additive	2.24	2.60	73.3%	85.3%
Edm.	Exp.S.	additive	additive	2.24	2.62	73.4%	85.7%
Edm.	TBATS	additive	additive	**2.13**	**1.99**	**69.8%**	**65.3%**

Table 4.9: One-week-ahead forecasting performance comparison between methods with different components for $\delta = 8$. For Grande Prairie. The mean and standard deviation of the calls over the 50 folds are 2.62 and 1.71, respectively.

City	Method	Trend	Seasonality	MAE	IQR	MASE	IQR
G.P.	Exp.S.	none	none	2.02	2.19	94.0%	102%
G.P.	Exp.S.	additive	none	2.04	2.23	94.9%	104%
G.P.	Exp.S.	none	additive	**1.42**	1.41	66.1%	65.4%
G.P.	Exp.S.	additive	additive	**1.42**	**1.12**	**65.9%**	**52.0%**
G.P.	TBATS	additive	additive	1.45	1.36	67.6%	63.2%

4.6.2 Comparison between methods with different resolutions

In this subsection we fix the method to be HW-A, because that method generally performed best. Tables 4.10–4.12 compare one-week-ahead forecast of methods with different temporal and spatial resolution in Calgary, Edmonton, and Grande Prairie respectively. Forecasts are made for the the biggest catchment area (or the city, for Grande Prairie) and for Monday [8 am - 4 pm) interval. As mentioned in the Section 4.5, forecasts are brought to the same resolution level (8-hour and catchment area) when comparing methods with different resolutions. The results indicate that there is little difference between the various top-down, bottom-up, or same-level methods. This means that, surprisingly, methods with a 24-hour temporal resolution, and thus no weekly seasonal component, are performing as well as methods with a seasonal component. The reason for this is that by using historical proportions to divide the 24-hour forecasts to 8-hours we are taking the weekly seasonality of call volumes into account. The good performance of such *top-down* approach to forecasting is in accordance with past research (Huddleston et al., 2015).

4.6.3 Comparison between univariate and multivariate methods

A comparison between a univariate exponential smoothing method with multivariate methods is shown in Table 4.13. All methods have an 8-hour temporal resolution and catchment area

Table 4.10: Comparison between the performance of methods with different temporal and spatial resolution in Calgary. All methods are exponential smoothing with an additive trend and an additive seasonality component.

City	δ	α	MAE	IQR	MASE	IQR
Cal.	4	h	1.76	1.86	87.1%	92.1%
Cal.	4	m	1.79	1.73	88.6%	85.4%
Cal.	4	l	1.67	**1.40**	82.6%	**69.2%**
Cal.	8	h	**1.64**	1.71	81.3%	84.6%
Cal.	8	m	1.76	1.81	87.2%	89.7%
Cal.	8	l	1.67	1.50	82.7%	74.1%
Cal.	24	h	**1.64**	1.58	81.3%	78.1%
Cal.	24	m	**1.64**	1.94	**81.2%**	96.2%
Cal.	24	l	1.65	1.71	81.9%	84.6%

Table 4.11: Comparison between the performance of methods with different temporal and spatial resolution in Edmonton. All methods are exponential smoothing with an additive trend and an additive seasonality component.

City	δ	α	MAE	IQR	MASE	IQR
Edm.	4	h	2.22	2.63	72.6%	86.0%
Edm.	4	m	2.34	2.44	76.7%	79.9%
Edm.	4	l	2.66	2.39	87.0%	78.2%
Edm.	8	h	2.24	2.62	73.4%	85.7%
Edm.	8	m	2.28	2.44	74.8%	80.0%
Edm.	8	l	2.59	**2.31**	84.8%	**75.7%**
Edm.	24	h	**2.19**	2.53	**71.7%**	82.8%
Edm.	24	m	2.28	2.50	74.6%	81.7%
Edm.	24	l	2.54	2.40	83.2%	78.5%

spatial resolution and have additive trend and seasonal components. The results indicate that the multivariate methods do not outperform the univariate methods.

4.6.4 Performance at different forecasting horizons

The MASE of an exponential smoothing method with additive trend and additive seasonality trained on data from Calgary with an 8-hour temporal resolution and city-level spatial resolution is shown in Figure 4.7. The black line represents the median MASE (among the 50 CV folds) and the upper and lower gray lines represent the 75th and the 25th percentiles

Table 4.12: Comparison between the performance of methods with different temporal and spatial resolution in Grande Prairie. All methods are exponential smoothing with an additive trend and an additive seasonality component.

City	δ	α	MAE	IQR	MASE	IQR
G.P.	4	l	1.47	1.16	68.5%	53.8%
G.P.	8	l	1.42	**1.12**	65.9%	**52.0%**
G.P.	24	l	**1.39**	1.24	**64.8%**	57.6%

Table 4.13: Comparison between a univariate exponential smoothing with multivariate methods. All methods have an 8-hour temporal resolution and catchment area spatial resolution and have additive trend and seasonal components

City	Method	MAE	IQR	MASE	IQR
Edm.	Exp.S.	**2.24**	2.62	**73.4%**	85.7%
Edm.	VES	2.31	3.01	75.6%	98.6%
Edm.	PVES	2.46	**2.57**	80.5%	**84.0%**
Cal.	Exp.S.	**1.64**	1.71	**81.3%**	84.6%
Cal.	VES	1.76	1.97	87.3%	97.3%
Cal.	PES	1.69	**1.65**	83.8%	**81.7%**

respectively. The median MASE is less than one indicating a lower error than the in-sample-error of the Naïve method. The result indicates that the forecast accuracy does not appear to decrease with forecasting horizon.

4.7 Exploratory data analysis using models

In addition to providing forecasts of the future, models are also important tools for learning about the data (Wickham and Grolemund, 2016). In this section, we discuss what can be learned from patterns in the state variables of the PVES model, for Calgary and Edmonton. In Figure 4.8, 4.10, and 4.11 we have plotted the values of the state variables of the PVES model as they are updated along the training set. From Figure 4.8 we see that the level of EMS calls in the city of Calgary has increased over time (we also observe an increasing level in Edmonton). This long-term trend is captured in the level state variable of the model as opposed to the trend state variable, which deals with the short-term trend of EMS calls

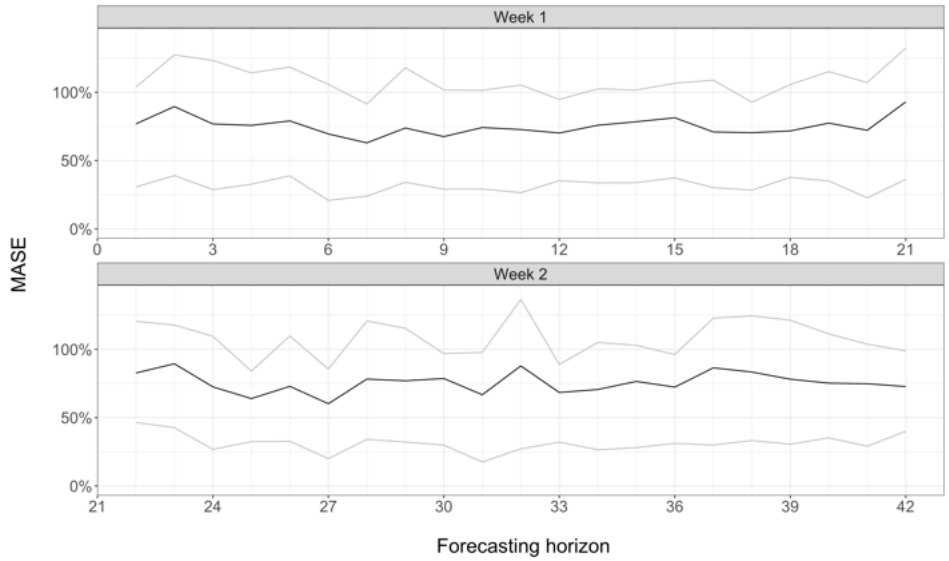

Figure 4.7: MASE at different forecasting horizons.

(changes of the level at 4-hr increments.)

Both the level and trend state variables appear to have monthly seasonality patterns. Figure 4.9 shows the seasonality pattern of the level state variable, for Edmonton and Calgary.. Each grey line is the de-trended level values of a year. The black line is a Loess curve going through all the data points. The seasonality level seems highest during January and July for Calgary and Edmonton but not Grande Prairie (not shown).

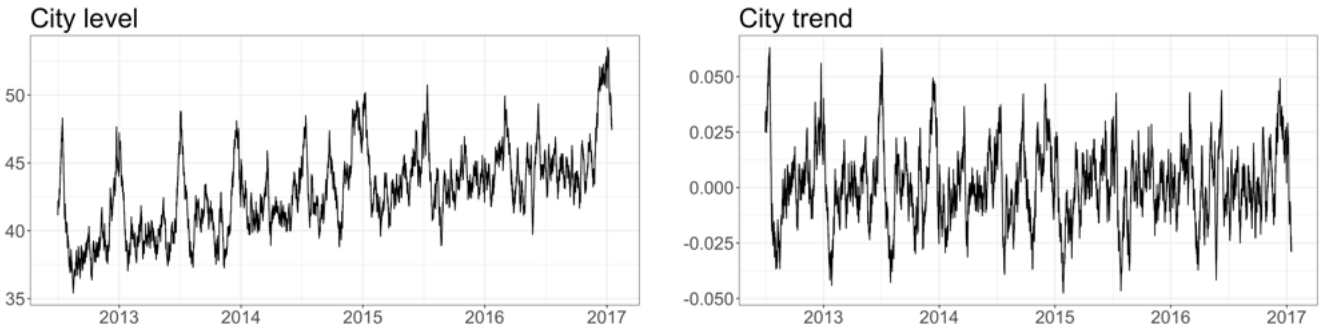

Figure 4.8: Updated values of the level (left) and trend (right) state variable, for Calgary.

Figure 4.10 shows the values of the proportion state variable of the PVES model. We see that the spatial distribution of calls also appears to have monthly seasonality patterns. This can be seen very clearly from the green curve (the central business district) of Calgary, where its proportion of the total call volume is highest during summer months and lowest

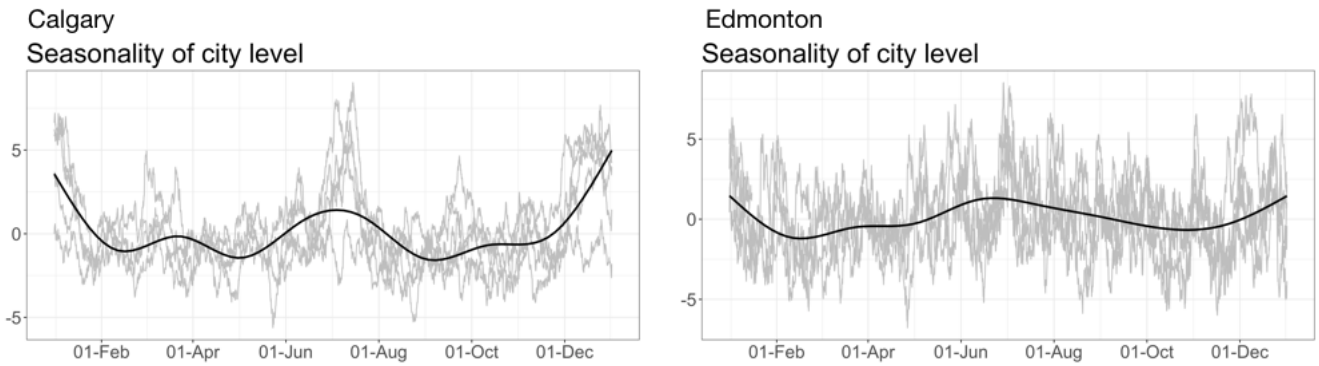

Figure 4.9: Seasonal patterns of the city level state variable.

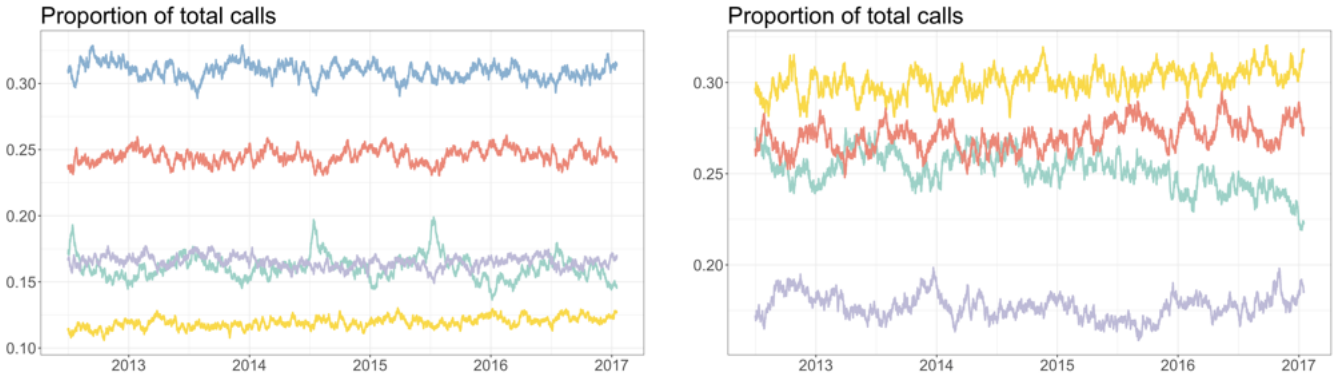

Figure 4.10: Updated values of the regions' proportion state variable for Calgary (left) and Edmonton (right).

during winter. The right-hand plot of Figure 4.10 seems to indicate that there has been a spatial redistribution of call volumes in Edmonton away from the center (green curve) and mostly towards the northern region (red) since 2015.

Figure 4.11 shows the values of the estimated seasonality index of different time intervals for different regions in Edmonton. The figure shows that the model is able to capture the temporal pattern of EMS calls shown in Figure 4.3. We also see extra information regarding how the seasonality index has changed throughout the year. For example, from the plot for [0:00, 4:00) of Region 2 (south) we see that call volumes of that region after midnight on weekends has been decreasing. Seasonality indices of regions in Calgary also follow the temporal pattern similar to Figure 4.3 but with no detectable change throughout the years.

As we have shown in this section, one advantage of using a model for doing exploratory data analysis is its ease of use. Training a model once and analysing the results decreases the

Figure 4.11: Updated values of the regions' seasonality index state variable.

amount of data work required, which in turn decreases the probability of making an error during the analysis.

4.8 Two-week forecast of jumps in call volume

As an example of how the forecasting models can be used for assisting decision makers of an EMS system, we conduct the following analysis: At the end of each week we make forecasts for 8-hr call volumes of the next two weeks for each catchment area. If the forecast of the call volume for a certain time in the week is α times more than the historical call volume of that time (based on average call volumes over the past 10 weeks), we label that forecast as a *jump*. We compare the forecasts against the actual call volumes and we calculate the true positives (TP), the number of times that we correctly forecast a jump, the false positives (FP), the number of times we incorrectly forecast a jump, and false negatives (FN), the number of

times we failed to forecast a jump. We then calculate the Precision, Recall, and the F_1 score as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}; \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \quad \text{F}_1 \text{ score} = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

The results of this analysis for the city of Calgary are shown in Figure 4.12. The left confusion matrix is for an α value of 10% and the right is for a value of 20%. We see that both Precision and Recall are lower for the higher α value, suggesting that larger jumps are identified with lower accuracy.

A limitation of this analysis is that the accuracy measures are aggregated over forecast horizons. Short forecasts horizons, say 4 to 24 hours, could be particularly relevant for short-term staffing adjustments.

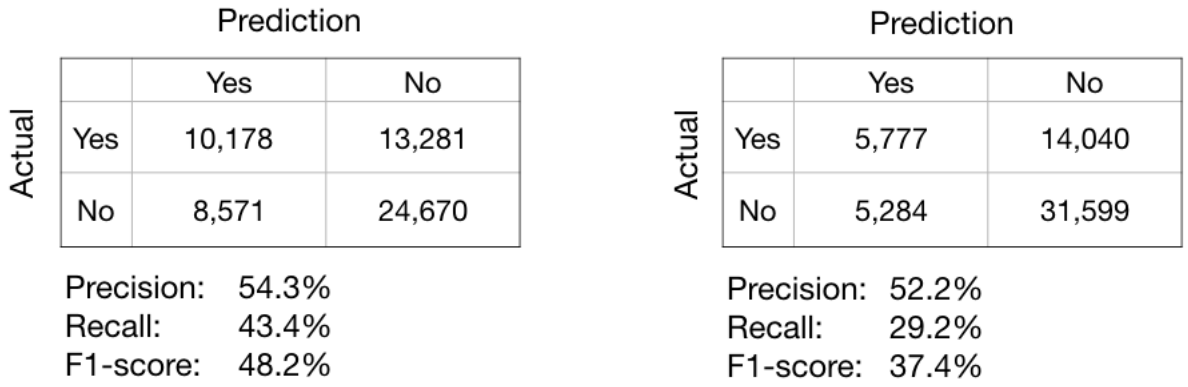

Figure 4.12: Confusion matrices for predicting a jump in call volumes in Calgary. The left matrix is for an α value of 10% and the right is for a value of 20%.

4.9 Discussion

In this paper we have explored the forecasting performance of different univariate and multivariate innovations state-space models on a diverse data set for three cities in Alberta,

Canada. We used three levels of temporal resolution, where forecasts were made for 4-hour, 8-hour, and 24-hour intervals, and three levels of spatial resolution, with forecasts for station catchment areas, intermediate regions, and the entire city. The results emphasis the importance of accounting for weekly seasonality of call volumes. Other factors, namely: the level of spatial and temporal resolution, univariate vs. multivariate forecasting, including a trend component, initial transformation of call volumes, accounting for autocorrelations among errors, do not seem to improve the performance of the models. In fact, a simple exponential smoothing model, having only a level component, with a 24-hour temporal resolution and a city-level spatial resolution can perform as well as other complicated models when comparing its performance on 8-hour intervals of a week and station catchment areas, provided that we appropriately divide its forecasts using historical values of call volumes. All models were, furthermore, show to have a lower error on out-of-sample data than the in-sample error made by a naïve model that forecasts each week's call volume as the previous week's value. We also proposed a new multivariate exponential smoothing model suitable for forecasting time series of multiple streams, where each stream's *market share* could stochastically change with time. We demonstrated the benefits of using forecasting models for learning about the data and for assisting with the operational decisions of EMS systems.

4.10 Apendix

4.10.1 Perturbing the data

The geographical location of each call is anonymized by randomly perturbing the latitude and longitude (ϕ, λ) to $(\phi + \epsilon_1, \lambda + \epsilon_2)$. The perturbation (ϵ_1, ϵ_2) follows a bivariate normal distribution, where the two components have mean zero and zero correlation. The variances are chosen so that the average perturbation area is roughly 400 square metres. The square perturbation distance is approximately:

$$\left(\frac{\pi}{180} R\epsilon_1\right)^2 + \left(\frac{\pi}{180} R\cos\phi\epsilon_2\right)^2$$

where R is the Earth's radius and ϕ is the latitude of the city. The approximation is because we are assuming the traversed distance is a straight line. In order for the average area of the perturbed location from the original location to equal 400 square metres, we need that:

$$\mathrm{E}\left[\left(\frac{\pi}{180} R\epsilon_1\right)^2 + \left(\frac{\pi}{180} R\cos(\phi)\epsilon_2\right)^2\right]\pi = 0.4$$

Substituting the latitude of Edmonton, $\phi = 53.54$ degrees and $R = 6{,}371$ km we obtain:

$$\sigma_{\epsilon_1}^2 + 0.3531\sigma_{\epsilon_2}^2 = 1.030e^{-05}$$

A one degree increase in latitude translates to a larger displacement than a one degree increase in longitude by a factor of $\frac{1}{\cos\phi}$. We, therefore, rescale the standard deviations of the perturbation as:

$$\sigma_{\epsilon_1} = \cos\phi\sigma_{\epsilon_2}$$

Solving for σ_{ϵ_1} and σ_{ϵ_2} we get $\sigma_{\epsilon_1} = 2.269e^{-03}$ and $\sigma_{\epsilon_2} = 3.819e^{-03}$

4.10.2 Test for inter-stream lag dependence

Results of the test for inter-stream lag dependence for call volumes of different station catchment area for Calgary and Edmonton are in Tables 4.14–4.17 and 4.18–4.20, respectively.

	Station 1	Station 11	Station 12	Station 14	Station 15	Station 17	Station 18
Station 1	0.111 *	0.074	0.036	0.025	0.007	0.035	0.008
Station 11	0.008	0.093	0.002	0.016	0.034	0.044	0.028
Station 12	0.019	0.007	0.045	0.031	0.016	0.001	0.019
Station 14	-0.002	0.010	0.040	0.082	0.011	-0.002	0.054
Station 15	-0.007	0.045	0.016	0.033	0.050	-0.001	-0.019
Station 17	0.027	-0.001	0.030	-0.006	0.014	0.065	0.049
Station 18	-0.059	0.057	0.023	0.016	0.019	0.069	0.066
Station 2	0.050	0.026	0.048	-0.018	0.044	0.046	0.060
Station 20	0.012	0.050	-0.004	-0.008	0.012	0.013	0.029
Station 21	0.007	0.027	0.011	0.027	0.025	0.008	0.027
Station 22	0.033	0.018	0.039	0.017	0.088	0.015	0.018
Station 23	0.014	-0.009	0.010	0.040	0.022	0.051	0.036
Station 24	-0.009	0.048	-0.004	0.022	0.046	0.005	-0.002
Station 28	-0.019	0.060	-0.013	0.000	0.058	0.051	0.028
Station 29	-0.012	-0.001	0.001	0.003	-0.011	-0.005	0.002
Station 3	0.090	-0.015	-0.030	0.005	0.057	-0.001	0.000
Station 30	0.029	0.040	0.026	0.022	0.037	0.041	0.004
Station 31	0.037	0.036	0.038	0.001	0.013	0.064	0.039
Station 32	0.047	0.052	-0.023	0.008	0.009	0.041	0.004
Station 33	-0.011	0.025	0.012	0.020	-0.051	0.037	0.077
Station 34	0.002	-0.028	-0.016	0.027	-0.017	0.022	-0.029
Station 4	0.068	0.033	0.027	0.011	0.070	0.023	0.013
Station 41	0.031	0.032	0.017	0.043	-0.017	0.044	0.032
Station 5	-0.014	0.071	0.033	-0.015	0.002	-0.002	0.024
Station 6	0.017	0.032	0.008	0.025	0.024	-0.001	0.001
Station 8	0.031	-0.006	0.007	-0.027	0.009	0.012	0.008
Station 9	0.009	0.028	0.014	0.008	0.044	-0.005	-0.012

* $p < 0.000066$ (Bonferroni adjusted significance level)

Table 4.14: Estimated d_{ij} coefficients for Calgary, station catchment area spatial resolution—Part 1 of 4.

	Station 2	Station 20	Station 21	Station 22	Station 23	Station 24	Station 28
Station 1	0.077	-0.007	0.029	0.027	0.028	0.044	-0.026
Station 11	-0.004	0.035	-0.004	-0.009	0.003	-0.014	-0.002
Station 12	0.020	0.039	0.053	0.033	0.025	0.038	0.070
Station 14	-0.010	0.046	0.013	0.009	0.003	0.030	0.012
Station 15	0.011	0.001	-0.019	-0.013	0.004	0.034	-0.029
Station 17	0.014	0.020	0.026	-0.008	-0.016	0.012	-0.012
Station 18	0.047	0.041	-0.017	0.001	-0.047	0.002	0.029
Station 2	0.124 *	-0.052	-0.021	0.078	0.025	0.012	0.007
Station 20	0.031	0.019	-0.009	-0.009	-0.006	0.042	0.042
Station 21	-0.020	0.009	0.054	0.061	0.015	-0.035	0.084
Station 22	0.020	0.037	0.040	0.023	0.018	0.050	0.010
Station 23	0.005	0.004	-0.015	-0.039	0.046	0.013	0.047
Station 24	-0.012	0.011	0.023	0.002	0.015	0.033	0.049
Station 28	-0.016	-0.011	0.025	0.010	-0.017	0.002	0.017
Station 29	0.030	-0.020	0.015	0.009	-0.009	0.008	-0.027
Station 3	0.031	-0.009	0.043	0.047	0.002	0.044	-0.006
Station 30	-0.017	0.040	0.030	0.016	0.021	0.004	0.061
Station 31	0.035	0.045	0.050	0.016	-0.004	0.022	0.080 *
Station 32	0.008	0.005	0.018	-0.017	-0.018	0.021	0.005
Station 33	0.002	0.064	-0.016	-0.011	-0.034	0.012	0.068
Station 34	-0.004	0.011	-0.020	0.027	-0.015	0.006	-0.006
Station 4	-0.010	0.067	-0.011	0.016	0.042	0.033	0.027
Station 41	0.034	0.060	0.030	0.054	0.032	-0.002	0.054
Station 5	0.020	-0.005	0.030	-0.008	0.078	0.028	0.029
Station 6	0.025	0.048	0.046	0.044	0.040	-0.008	-0.004
Station 8	-0.008	0.043	0.010	0.011	0.010	-0.004	-0.002
Station 9	-0.048	0.006	-0.006	0.012	0.033	-0.001	0.065

* $p < 0.000066$ (Bonferroni adjusted significance level)

Table 4.15: Estimated d_{ij} coefficients for Calgary, station catchment area spatial resolution—Part 2 of 4.

	Station 29	Station 3	Station 30	Station 31	Station 32	Station 33	Station 34
Station 1	-0.013	0.052	0.010	0.050	0.067	0.008	0.011
Station 11	-0.026	0.005	-0.014	0.034	0.060	0.058	-0.016
Station 12	0.000	0.018	0.043	-0.015	0.006	0.038	-0.027
Station 14	0.054	0.034	0.025	0.030	0.008	-0.005	-0.054
Station 15	0.011	0.039	0.050	0.011	0.012	0.054	-0.020
Station 17	0.031	0.009	0.015	0.046	0.038	0.044	0.042
Station 18	0.001	0.022	0.046	0.008	0.001	0.094 *	-0.002
Station 2	0.017	0.057	-0.011	0.031	0.057	0.020	0.008
Station 20	-0.003	-0.037	0.042	0.002	0.062	0.010	-0.026
Station 21	0.017	0.022	0.019	0.103 *	0.017	0.030	-0.012
Station 22	-0.002	0.059	0.031	0.051	0.041	0.021	-0.063
Station 23	0.002	0.033	-0.032	-0.006	0.061	0.010	-0.038
Station 24	0.027	0.007	-0.005	0.027	0.061	0.036	-0.024
Station 28	0.023	-0.015	0.027	0.037	0.063	0.033	-0.037
Station 29	0.452 *	-0.033	-0.039	-0.029	-0.028	0.012	0.274 *
Station 3	0.019	0.198 *	0.001	-0.026	0.007	-0.001	-0.036
Station 30	-0.035	0.011	0.108 *	0.123 *	0.069	-0.029	-0.072
Station 31	-0.017	0.017	0.049	0.052	0.087 *	0.030	-0.069
Station 32	-0.092 *	0.004	0.049	0.071	0.100 *	0.061	0.014
Station 33	-0.030	-0.007	0.060	0.016	0.073	0.103 *	0.044
Station 34	0.330 *	-0.016	-0.031	-0.015	-0.022	-0.004	0.380 *
Station 4	0.006	0.011	0.038	-0.037	0.037	-0.010	-0.014
Station 41	0.013	0.022	0.001	0.038	0.032	0.012	-0.063
Station 5	0.003	-0.035	0.000	0.039	0.037	0.031	-0.017
Station 6	-0.007	-0.002	-0.011	0.033	-0.010	0.015	-0.039
Station 8	-0.046	0.032	0.029	0.028	0.042	-0.001	-0.041
Station 9	0.013	0.032	0.033	-0.007	0.032	0.022	-0.008

* $p < 0.000066$ (Bonferroni adjusted significance level)

Table 4.16: Estimated d_{ij} coefficients for Calgary, station catchment area spatial resolution—Part 3 of 4.

	Station 4	Station 41	Station 5	Station 6	Station 8	Station 9
Station 1	0.024	0.032	-0.012	0.054	0.059	-0.001
Station 11	-0.020	0.048	0.045	-0.005	0.033	0.002
Station 12	0.029	0.000	0.008	0.027	0.026	-0.029
Station 14	0.010	0.001	0.034	0.014	-0.005	0.025
Station 15	-0.014	-0.017	0.039	0.015	-0.012	0.052
Station 17	-0.022	-0.005	-0.006	-0.033	-0.013	0.018
Station 18	-0.008	0.014	0.028	-0.058	0.022	0.027
Station 2	-0.009	0.003	-0.013	0.055	-0.006	0.004
Station 20	-0.006	0.006	0.005	-0.017	-0.002	0.020
Station 21	-0.020	-0.015	-0.009	-0.035	0.004	0.048
Station 22	0.015	0.080	-0.011	0.021	0.019	0.043
Station 23	0.040	0.076	0.095 *	-0.011	-0.024	-0.036
Station 24	0.050	0.030	0.006	0.038	-0.021	-0.004
Station 28	0.024	0.092	0.053	-0.006	-0.007	0.004
Station 29	-0.008	-0.009	-0.015	-0.005	-0.018	-0.001
Station 3	0.030	0.017	-0.001	-0.008	0.047	0.024
Station 30	-0.008	0.029	0.049	0.017	-0.010	0.015
Station 31	-0.043	0.036	0.018	0.031	-0.003	0.055
Station 32	-0.012	0.080	0.038	0.022	0.019	0.004
Station 33	0.004	0.027	-0.018	-0.063	0.003	0.015
Station 34	-0.017	-0.005	-0.022	0.004	-0.039	-0.023
Station 4	0.009	0.012	0.066	0.005	0.006	-0.028
Station 41	-0.048	0.069	0.026	0.025	0.029	0.006
Station 5	0.018	-0.008	0.053	0.000	-0.002	0.011
Station 6	0.025	0.016	0.031	0.014	0.036	0.041
Station 8	-0.008	0.034	0.004	0.005	0.057	0.026
Station 9	0.047	0.041	-0.013	0.013	0.008	0.031

* $p < 0.000066$ (Bonferroni adjusted significance level)

Table 4.17:]
Estimated d_{ij} coefficients for Calgary, station catchment area spatial resolution—Part 4 of 4.

	Station 24	Station 25	Station 26	Station 27	Station 28	Station 29
Station 24	0.084	0.003	0.059	0.020	0.011	0.041
Station 25	0.005	0.060	0.083	0.032	-0.035	0.022
Station 26	0.069	0.043	0.123	0.073	0.008	0.012
Station 27	0.058	0.014	0.051	0.019	0.013	0.053
Station 28	0.066	0.016	0.036	-0.001	0.100 *	-0.017
Station 29	0.014	0.025	0.038	0.041	-0.006	0.049
Station 30	0.075	0.017	0.082	-0.010	0.028	0.035
Station 31	0.002	0.025	0.042	0.012	-0.007	0.002
Station 32	0.057	0.036	0.032	0.094 *	0.015	-0.006
Station 33	0.074	0.022	0.064	0.031	0.037	0.021
Station 34	-0.002	0.037	0.021	-0.020	0.015	0.014
Station 36	0.004	-0.023	-0.005	-0.050	-0.025	0.052
Station 37	0.003	0.013	0.032	0.013	0.020	0.021
Station 39	0.008	-0.004	0.032	0.056	0.016	0.027
Station 41	0.058	0.054	-0.007	0.049	0.010	0.081
Station 5	0.004	0.029	0.021	0.021	0.073	0.042

* $p < 0.000195$ (Bonferroni adjusted significance level)

Table 4.18: Estimated d_{ij} coefficients for Edmonton, station catchment area spatial resolution—Part 1 of 3.

	Station 30	Station 31	Station 32	Station 33	Station 34	Station 36
Station 24	0.105	0.022	0.035	0.043	0.018	-0.023
Station 25	0.048	0.018	0.022	0.042	-0.022	0.008
Station 26	0.097 *	-0.014	0.051	0.058	0.023	0.029
Station 27	0.011	0.051	0.006	-0.001	0.025	-0.013
Station 28	0.042	0.011	0.024	0.022	0.023	-0.021
Station 29	0.103 *	0.024	-0.010	0.037	0.017	0.034
Station 30	0.096 *	0.024	0.030	0.090 *	0.004	0.017
Station 31	0.010	0.081	0.036	0.053	-0.019	-0.006
Station 32	-0.018	-0.005	0.025	0.021	0.013	0.004
Station 33	0.074	0.032	0.053	0.086	0.014	0.008
Station 34	0.050	-0.022	0.009	0.043	0.025	0.037
Station 36	-0.001	0.041	0.008	-0.011	0.044	0.183 *
Station 37	0.038	0.004	0.050	0.022	0.018	0.002
Station 39	-0.002	0.045	0.018	0.041	0.028	0.003
Station 41	0.022	-0.027	-0.007	0.050	0.037	-0.025
Station 5	0.015	0.062	0.033	0.013	0.038	0.027

* $p < 0.000195$ (Bonferroni adjusted significance level)

Table 4.19: Estimated d_{ij} coefficients for Edmonton, station catchment area spatial resolution—Part 2 of 3.

	Station 37	Station 39	Station 41	Station 5
Station 24	0.019	0.044	0.018	-0.037
Station 25	0.011	0.021	0.018	0.051
Station 26	0.033	0.008	0.075	0.022
Station 27	-0.022	0.062	-0.007	0.044
Station 28	0.076	0.046	0.009	0.070
Station 29	-0.014	-0.011	0.042	0.041
Station 30	-0.028	0.063	0.008	0.038
Station 31	0.072	0.040	0.001	0.048
Station 32	0.019	0.050	0.046	0.014
Station 33	0.000	0.017	0.024	0.008
Station 34	0.036	0.010	0.001	0.023
Station 36	0.019	0.022	-0.019	0.002
Station 37	-0.021	0.021	0.025	0.052
Station 39	0.018	0.086	0.056	0.068
Station 41	0.001	0.008	-0.030	0.012
Station 5	0.078	0.012	-0.016	0.235 *

* $p < 0.000195$ (Bonferroni adjusted significance level)

Table 4.20: Estimated d_{ij} coefficients for Edmonton, station catchment area spatial resolution—Part 3 of 3.

4.10.3 Results of models with multiplicative and damped trend components

Results of comparison between models with different components, including multiplicative trends or seasonality and damped trend, are presented in Tables 4.21–4.23. Since we cannot train models with multiplicative trend or seasonality when zero values are present, we use a city level spatial resolution. The results, however, are for the forecast for a single catchment area in the city center district. This means that we are taking a top-down approach where we first forecast for the entire city and then divide the forecasts at smaller, station catchment area, spatial regions. The temporal resolution is 4-hours in all cases. Furthermore, we do not have multiplicative state values for Grande Prairie since even at the city level, it contains zero values.

City	Method	Trend	Seasonality	MAE	IQR	MASE	IQR
Cal.	Exp.S.	none	none	2.64	2.97	131.0%	147.0%
Cal.	Exp.S.	additive	none	2.66	2.97	132.0%	147.0%
Cal.	Exp.S.	additive-damped	none	2.64	2.97	131.0%	147.0%
Cal.	Exp.S.	multiplicative	none	2.66	2.97	132.0%	147.0%
Cal.	Exp.S.	multiplicative-damped	none	2.66	2.99	132.0%	148.0%
Cal.	Exp.S.	none	additive	1.67	1.40	82.5%	69.5%
Cal.	Exp.S.	none	multiplicative	1.65	1.76	81.7%	86.9%
Cal.	Exp.S.	additive	additive	1.67	1.40	82.6%	69.2%
Cal.	Exp.S.	additive-damped	additive	1.67	1.45	82.8%	71.7%
Cal.	Exp.S.	additive	multiplicative	**1.65**	1.76	**81.5%**	86.9%
Cal.	Exp.S.	additive-damped	multiplicative	**1.65**	1.79	81.9%	88.5%
Cal.	Exp.S.	multiplicative	multiplicative	**1.65**	1.76	81.8%	87.1%
Cal.	Exp.S.	multiplicative	additive	1.67	1.40	82.8%	**69.3%**
Cal.	Exp.S.	multiplicative-damped	multiplicative	**1.65**	1.77	81.7%	87.8%
Cal.	Exp.S.	multiplicative-damped	additive	1.67	1.40	82.5%	69.5%
Cal.	TBATS	additive	additive	1.68	**1.35**	83.1%	66.8%

Table 4.21: One-week-ahead forecasting performance comparison between methods with different components for $\alpha = l$—Calgary.

City	Method	Trend	Seasonality	MAE	IQR	MASE	IQR
Edm.	Exp.S.	none	none	6.00	3.25	197.0%	106.0%
Edm.	Exp.S.	additive	none	6.07	3.26	199.0%	107.0%
Edm.	Exp.S.	additive-damped	none	6.02	3.24	197.0%	106.0%
Edm.	Exp.S.	multiplicative	none	6.08	3.26	199.0%	107.0%
Edm.	Exp.S.	multiplicative-damped	none	6.01	3.33	197.0%	109.0%
Edm.	Exp.S.	none	additive	2.64	2.39	86.3%	78.3%
Edm.	Exp.S.	none	multiplicative	2.52	2.36	**82.7%**	77.4%
Edm.	Exp.S.	additive	additive	2.66	2.39	87.0%	78.2%
Edm.	Exp.S.	additive-damped	additive	2.63	2.34	86.2%	76.8%
Edm.	Exp.S.	additive	multiplicative	2.54	2.32	83.3%	76.1%
Edm.	Exp.S.	additive-damped	multiplicative	**2.53**	2.36	**82.7%**	77.4%
Edm.	Exp.S.	multiplicative	multiplicative	**2.53**	**2.26**	82.9%	74.0%
Edm.	Exp.S.	multiplicative	additive	2.64	2.29	86.5%	74.9%
Edm.	Exp.S.	multiplicative-damped	multiplicative	**2.53**	2.34	82.8%	76.6%
Edm.	Exp.S.	multiplicative-damped	additive	2.64	**2.26**	86.6%	**73.9%**
Edm.	TBATS	additive	additive	2.75	3.09	90.1%	101%

Table 4.22: One-week-ahead forecasting performance comparison between methods with different components for $\alpha = l$—Edmonton.

City	Method	Trend	Seasonality	MAE	IQR	MASE	IQR
G.P.	Exp.S.	none	none	2.02	2.16	93.8%	100.0%
G.P.	Exp.S.	additive	none	2.03	2.18	94.5%	101.0%
G.P.	Exp.S.	additive-damped	none	2.01	2.15	93.6%	100.0%
G.P.	Exp.S.	none	additive	**1.42**	1.39	**66.0%**	64.6%
G.P.	Exp.S.	additive	additive	1.47	1.16	68.5%	53.8%
G.P.	Exp.S.	additive-damped	additive	1.45	**1.12**	67.3%	**52.3%**
G.P.	TBATS	additive	additive	1.48	1.53	68.8%	71.1%

Table 4.23: One-week-ahead forecasting performance comparison between methods with different components for $\alpha = l$—Grande Prairie.

Chapter 5

Conclusion

In this dissertation we presented three studies related to business analytics. As we mentioned in Chapter 1, analytics can be descriptive, predictive, or prescriptive.

In Chapter 2, our focus was mostly on descriptive analytics. This was because, due to the complexity of the data, simply discovering pattern is a challenging task. We proposed a new methodology that can be used as an exploratory data analysis tool to gain a better understanding of the dynamics among editors within Wikipedia. We mentioned several possible avenues for future study at the end of Chapter 2. In addition, as a follow-up study, using this analytical tool, we can investigate how editors in Wikipedia balance seemingly paradoxical goals, like acquiring new knowledge while maintaining existing knowledge. This trade-off between exploration and exploitation (O'Reilly III and Tushman, 2013; Posen and Levinthal, 2012) is especially interesting in Wikipedia once we realize that unlike traditional organizations, the vast majority of editors in Wikipedia engage in exploratory activities while only a small minority deal with exploitatory activities (Aaltonen and Kallinikos, 2012). Another interesting follow-up study is to identify success factors of open, online, and collaborative communities like Wikipedia and to study why some communities succeed in producing high-quality products (articles) while others fail.

In Chapter 3, our focus was mostly on predictive analytics. We provided a case study of

how our methodology of finding informative variables can be used as a prescriptive analytics tool in a real-world setting. As a follow-up study we can empirically test the performance of such a tool. For example, we can conduct an experimental study to measure the difference in outcome (final sales) of applying a tool that first estimates customers' probability of purchase return and then provides incentives to decrease the probability. In a healthcare setting, we might be interested in identifying patients with a high probability of not showing up to their scheduled appointment and use that information to improve scheduling.

In Chapter 4, our focus was on finding the best forecasting model suitable for prescriptive analytics like improving staffing of EMS stations. In our analysis we only considered the time and location of call volumes. We can extend this study by looking at other factors that might affect call volumes such as weather condition, holidays, or special events that might be happening within the city. We can also further explore the binary classification methods for predicting when staffing adjustments should be made. For example, since the operational costs of prediction errors (false-positive vs. false-negative) are different, we can investigate ways of accounting for such asymmetry. Finally, with respect to multivariate forecasting, we can study whether a VES model can be formulated that incorporates the type of lag dependence that Ye et al. (2019) found to be beneficial to include, has a limited number of parameters, and performs well out of sample.

Bibliography

A. Aaltonen and J. Kallinikos. Coordination and learning in Wikipedia: Revisiting the dynamics of exploitation and exploration. In *Managing Human Resources by Exploiting and Exploring Peoples Potentials*, pages 161–192. Emerald Group Publishing Limited, 2012.

E. T. Anderson, K. Hansen, and D. Simester. The option value of returns: Theory and empirical evidence. *Marketing Science*, 28(3):405–423, 2009.

O. Arazy, H. Brausen, D. Turner, A. Balila, E. Stroulia, and Lanir J. coDNA: Visualizing peer production processes. *Computer Supported Cooperative Work (CSCW)*, 2015.

R. Aringhieri, M. E. Bruni, S. Khodaparasti, and J.T. Van Essen. Emergency medical services and beyond: Addressing new challenges through a wide literature review. *Computers & Operations Research*, 78(1):349–368, 2017.

G. Athanasopoulos and A. de Silva. Multivariate exponential smoothing for forecasting tourist arrivals. *Journal of Travel Research*, 51(5):640–652, 2012.

C. Batini, S. Ceri, and S. Navathe. *Entity Relationship Approach.* Elsevier Science Publishers BV (North Holland), 1989.

J. D. Bermúdez, A. Corberán-Vallet, and E. Vercher. Multivariate exponential smoothing: A Bayesian forecast approach based on simulation. *Mathematics and Computers in Simulation*, 79(5):1761–1769, 2009.

R. P. Biuk-Aghai and R. C. K. Chan. Feeling the pulse of a wiki: Visualization of recent changes in Wikipedia. In *Proceedings of the 5th International Symposium on Visual Information Communication and Interaction*, pages 77–86. ACM, 2012.

H. D. Bondell and B. J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics*, 64(1):115–123, 2008.

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

P. Buhlmann, P. Rutimann, S. van de Geer, and C. H. Zhang. Correlated variables in regression: Clustering and sparse estimation. *J. Statist. Plann. Inference*, 143(11):1835–1858, 2013.

N. Channouf, P. LEcuyer, A. Ingolfsson, and A. N. Avramidis. The application of forecasting techniques to modeling emergency medical system calls in Calgary, Alberta. *Health Care Management Science*, 10(1):25–45, 2007.

A. Corberán-Vallet, José D. Bermúdez, and E. Vercher. Forecasting correlated time series with exponential smoothing models. *International Journal of Forecasting*, 27(2):252–265, 2011.

T. Davenport and J. Harris. *Competing on Analytics: Updated, with a New Introduction: The New Science of Winning.* Harvard Business Press, 2017.

A. M. De Livera, R. J. Hyndman, and R. D. Snyder. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496):1513–1527, 2011.

A. De Silva, R. J. Hyndman, and R. Snyder. The vector innovations structural time series framework: A simple approach to multivariate forecasting. *Statistical Modelling*, 10(4): 353–374, 2010.

M. Dettling and P. Bühlmann. Finding predictive gene groups from microarray data. *Journal of Multivariate Analysis*, 90(1):106–131, 2004.

J. Du Preez and S. F. Witt. Univariate versus multivariate time series forecasting: An application to international tourism demand. *International Journal of Forecasting*, 19(3): 435–451, 2003.

D. Eddelbuettel and R. François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011. doi: 10.18637/jss.v040.i08. URL `http://www.jstatsoft.org/v40/i08/`.

J. Fan and J. LV. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101, 2010.

J. Fan and Q. Yao. *Nonlinear Time Series: Nonparametric and Parametric Methods.* Springer Science & Business Media, 2008.

R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

L. Feng and Y. Shi. Forecasting mortality rates: Multivariate or univariate models? *Journal of Population Research*, 35(3):289–318, 2018.

R. L. Francis, T. J. Lowe, M. B. Rayco, and A. Tamir. Aggregation error for location models: Survey and analysis. *Annals of Operations Research*, 167(1):171–208, 2009.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 2010.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques.* Elsevier, 2011.

T. Hastie, R. Tibshirani, D. Botstein, and P. Brown. Supervised harvesting of expression trees. *Genome Biology*, 2(1):1–0003, 2001.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Prediction, Inference and Data Mining.* Springer-Verlag, 2nd edition, 2009. New York.

J. D. Hess and G. E. Mayhew. Modeling merchandise returns in direct marketing. *Journal of Interactive Marketing*, 11(2):20–35, 1997.

J. D. Hess, W. Chu, and E. Gerstner. Controlling product returns in direct marketing. *Marketing Letters*, 7(4):307–317, 1996.

C.C. Holt. Forecasting trends and seasonals by exponentially weighted averages. Carnegie Institute of Technology. Technical report, Pittsburgh ONR memorandum, 1957.

J. Huang, S. Ma, H. Li, and C. H. Zhang. The sparse Laplacian shrinkage estimator for high-dimensional regression. *Annals of statistics*, 39(4):2021, 2011.

S. H. Huddleston, J. H. Porter, and D. E. Brown. Improving forecasts for noisy geographic time series. *Journal of Business Research*, 68(8):1810–1818, 2015.

K. Hwang, D. Kim, K. Lee, C. Lee, and S. Parki. Variable selection method for signomial classification. *Annals of Operations Research*, 2016.

R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder. *Forecasting with Exponential Smoothing: The State Space Approach.* Springer Science & Business Media, 2008.

R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 26(3):1–22, 2008. URL `http://www.jstatsoft.org/article/view/v027i03`.

R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

A. Ingolfsson. EMS planning and management. In *Operations Research and Health Care Policy*, pages 105–128. Springer, 2013.

N. Janakiraman and L. Ordóñez. Effect of effort and deadlines on consumer product returns. *Journal of Consumer Psychology*, 22(2):260–271, 2012.

M. Kendall. *A Course in Multivariate Analysis.* Griffin: London, 1957.

A. Kittur and R. E. Kraut. Harnessing the wisdom of crowds in Wikipedia: Quality through coordination. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 37–46. ACM, 2008.

A. J. Knobbe, M. De Haas, and A. Siebes. Propositionalisation and aggregates. In *Principles of Data Mining and Knowledge Discovery*, pages 277–288. Springer, 2001.

M. Kuhn and K. Johnson. *Applied Predictive Modeling.* Springer, 2013.

J. A. Lowthian, P. A. Cameron, J. U. Stoelwinder, A. Curtis, A. Currell, M. W. Cooke, and J. J. McNeil. Increasing utilisation of emergency ambulances. *Australian Health Review*, 35(1):63–69, 2011.

P. Massa, M. Napolitano, F. Scrinzi, and M. Ferron. WikiTrip: Animated visualization over time of geo-location and gender of Wikipedians who edited a page. In *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration*, page 40. ACM, 2012.

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

A. Micheletti, D. Morale, D. Rapati, and P. Nolli. A stochastic model for simulation and forecasting of emergencies in the area of Milano. In *2010 IEEE Workshop on Health Care Management (WHCM)*, pages 1–6. IEEE, 2010.

J. H. Miller and S. E. Page. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life.*, volume 17. Princeton university press, 2009.

D. A. Mollenkopf, R. Frankel, and I. Russo. Creating value through returns management: Exploring the marketing–operations interface. *Journal of Operations Management*, 29(5): 391–403, 2011.

K. G. Munjal, R. A. Silverman, J. Freese, J. D. Braun, B. J. Kaufman, D. Isaacs, A. Werner, M. Webber, C. B. Hall, and D. J. Prezant. Utilization of emergency medical services in a large urban area: Description of call types and temporal trends. *Prehospital Emergency Care*, 15(3):371–380, 2011.

R. Nason. *It's Not Complicated: The Art and Science of Complexity in Business.* University of Toronto Press, 2017.

J. Ni, S. A. Neslin, and B. Sun. Database submission-the ISMS durable goods data sets. *Marketing Science*, 31(6):1008–1013, 2012.

S. Nunes, C. Ribeiro, and G. David. WikiChanges: Exposing Wikipedia revision activity. In *Proceedings of the 4th International Symposium on Wikis*, page 25. ACM, 2008.

C. A. O'Reilly III and M. L. Tushman. Organizational ambidexterity: Past, present, and future. *Academy of management Perspectives*, 27(4):324–338, 2013.

B. Otjacques, Maël Cornil, and F. Feltz. Visualizing cooperative activities with ellimaps: The case of Wikipedia. In *International Conference on Cooperative Design, Visualization and Engineering*, pages 44–51. Springer, 2009.

C. Perlich and F. Provost. Distribution-based aggregation for relational learning with identifier attributes. *Machine Learning*, 62(1-2):65–105, 2006.

J. A. Petersen and V. Kumar. Are product returns a necessary evil? Antecedents and consequences. *Journal of Marketing*, 73(3):35–51, 2009.

J. A. Petersen and V. Kumar. Perceived risk, product returns, and optimal resource allocation: Evidence from a field experiment. *Journal of Marketing Research*, 52(2):268–285, 2015.

B. Pfaff. VAR, SVAR and SVEC Models: Implementation within R package vars. *Journal of Statistical Software*, 27(4), 2008. URL `http://www.jstatsoft.org/v27/i04/`.

D. Pfeffermann and J. Allon. Multivariate exponential smoothing: Method and practice. *International Journal of Forecasting*, 5(1):83–98, 1989.

A. Popescul and L. H. Ungar. Statistical relational learning for link prediction. In *IJCAI workshop on learning statistical models from relational data*, 2003.

H. E. Posen and D. A. Levinthal. Chasing a moving target: Exploitation and exploration in dynamic environments. *Management Science*, 58(3):587–601, 2012.

R Core Team. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria, 2013. URL `http://www.R-project.org/`.

M. Reuter-Oppermann, P. L. van den Berg, and J. L. Vile. Logistics for emergency medical service systems. *Health Systems*, 6(3):187–208, 2017.

A. P. Reynolds, G. Richards, B. de la Iglesia, and V. J. Rayward-Smith. Clustering rules: A comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms*, 5(4):475–504, 2006.

M. Samorani. Automatically generate a flat mining table with Dataconda. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1644–1647. IEEE, 2015.

M. Samorani, M. Laguna, R. K. DeLisle, and D. C. Weaver. A randomized exhaustive propositionalization approach for molecule classification. *INFORMS Journal on Computing*, 23 (3):331–345, 2011.

H. Setzler, C. Saydam, and S. Park. EMS call volume predictions: A comparative study. *Computers & Operations Research*, 36(6):1843–1851, 2009.

Y. She. *Sparse Regression with Exact Clustering.* ProQuest, 2008.

D. T. Shih, S. B. Kim, V. C. P. Chen, J. M. Rosenberger, and V. L. Pilla. Efficient computer experiment-based optimization through variable selection. *Annals of Operations Research*, 216(1):287–305, 2014.

H. A. Simon. Rational decision making in business organizations. *The American Economic Review*, pages 493–513, 1979.

B. Suh, E. H. Chi, A. Kittur, and B. A. Pendleton. Lifting the veil: Improving accountability and social transparency in Wikipedia with wikidashboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1037–1040. ACM, 2008.

I. Svetunkov. *smooth: Forecasting using state space models.*, 2018. URL `https://CRAN.R-project.org/package=smooth`. R package version 2.4.7.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

R. S. Tsay. *Multivariate Time Series Analysis: With R and Financial Applications.* John Wiley & Sons, 2013.

F. B. Viégas, M. Wattenberg, and K. Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 575–582. ACM, 2004.

F. B. Viegas, M. Wattenberg, J. Kriss, and F. Van Ham. Talk before you type: Coordination in Wikipedia. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pages 78–78. IEEE, 2007.

J. L. Vile, J. W. Gillard, P. R. Harper, and V. A. Knight. Predicting ambulance demand using singular spectrum analysis. *Journal of the Operational Research Society*, 63(11): 1556–1565, 2012.

M. Wattenberg, F. B. Viegas, and K. Hollenbach. Visualizing activity on Wikipedia with chromograms. In *Human-Computer InteractionINTERACT*, pages 272–287. Springer Berlin Heidelberg, 2007.

H. Wickham and G. Grolemund. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. " O'Reilly Media, Inc.", 2016.

P. R. Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

H. Ye, J. Luedtke, and H. Shen. Call center arrivals: When to jointly forecast multiple streams? *Production and Operations Management*, 28(1):27–42, 2019.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society*, 68(1):49–67, 2007.

Z. Zhou, D. S. Matteson, D. B. Woodard, S. G. Henderson, and A. C. Micheas. A spatio-temporal point process model for ambulance demand. *Journal of the American Statistical Association*, 110(509):6–15, 2015.

Z. Zhou, D. S. Matteson, et al. Predicting Melbourne ambulance demand using kernel warping. *The Annals of Applied Statistics*, 10(4):1977–1996, 2016.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.