IMPROVING RECOMMENDER ENGINES BY INTEGRATING SPATIAL STATISTICS

by

Aisha Sikder

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Earth Systems and GeoInformation Sciences

Committee:

_____	Dr. Andreas Züfle, Committee Chair
_____	Dr. Dieter Pfoser, Committee Member
_____	Dr. Arie Croitoru , Committee Member
_____	Dr. Andrew Crooks, Committee Member
_____	Dr. Dieter Pfoser, Department Chair
_____	Dr. Donna M. Fox, Associate Dean, Office of Student Affairs & Special Programs, College of Science
_____	Dr. Ali Andalibi, Interim Dean, The College of Science

Date: _____ Spring 2020
George Mason University
Fairfax, VA

Improving Recommender Engines by Integrating Spatial Statistics

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Aisha Sikder
Master of Science
George Mason University, 2015
Bachelor of Science
George Mason University, 2012

Director: Dr. Andreas Züfle, Professor
Department of Geography and GeoInformation Science

Spring 2020
George Mason University
Fairfax, VA

Dedication

I dedicate this dissertation to my father, Mohammad Yunus Sikder, who has shown me the power of knowledge and the value of persistence. From a poor village in Bangladesh, my father has climbed the long and arduous ladder to success purely from the desire to leave poverty, the will for a better life, and education, education, and more education. Through his actions in life, he has been my role model for imagining the unimaginable, changing the world with education, always focusing on the future, and above all - never breaking mentally, no matter how painful the obstacles.

Acknowledgments

I would like to thank my advisor, Dr. Andreas Züfle, for encouraging me to push through the continuous improvements to this dissertation and showing me how to execute quality research. I also want to thank my husband and mother for the constant emotional and mental support to overcome the stress that comes with publishing papers and writing a dissertation while working full time to produce this work.

Table of Contents

List of Tables

List of Figures

Abstract

IMPROVING RECOMMENDER ENGINES BY INTEGRATING SPATIAL STATISTICS

Aisha Sikder, PhD

George Mason University, 2020

Dissertation Director: Dr. Andreas Züfle

The web has become a significant medium for business transactions and e-commerce. With such a vast quantity of options for users to choose and buy, recommender engines have been created to analyze patterns of user interests in products. These engines help to tailor recommendations that users are likely to enjoy buying based on their previous purchases or explicit feedback on their likes and dislikes. There are numerous algorithms to build recommender engines but the one that became most popular was Simon Funk's Singular Value Decomposition (SVD) for the Netflix Challenge. He demonstrated that matrix factorization models are superior to classic nearest-neighbor collaborative filtering techniques for producing product recommendations [3].

The main scope of this dissertation is to develop effective and efficient techniques that improve upon Simon Funk's SVD by thoughtfully integrating spatial statistics in such a way that will significantly improve the prediction error. Based on the type of data (whether it may be more spatially autocorrelated or latent), the method to blend both algorithms together can be quite different. The contribution of this dissertation is focused on building unique models that integrate SVD and kriging in different ways based on the type of data given to understand when to use them and why.

Chapter 1: Introduction

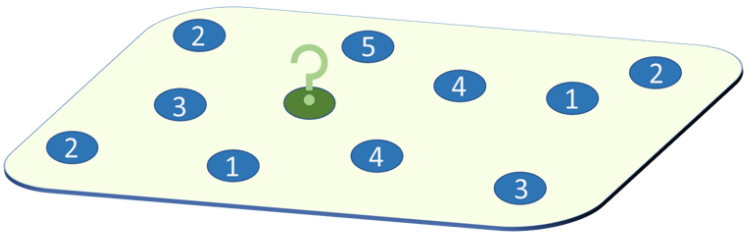
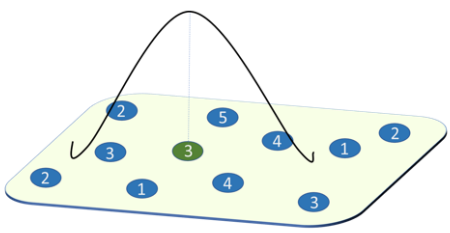


Figure 1.1: Dissertation Motivation

The contribution of this dissertation is in the ability to make a prediction at an unknown location based on the observed point values as shown in Figure 1.1. While we can use traditional approaches like kriging to interpolate a value based on the values of other nearby points (as shown in Figure 1.2a) or use matrix factorization approaches that take into consideration the hidden patterns within the data based on the available information (as shown in Figure 1.2b), the algorithms in this research improve upon these two very different paradigms. While these types of methods (kriging and matrix factorization) have their merit, the goal of this work is to combine both methods to improve the prediction power of either model.

1.1 Purpose

Singular value decomposition (SVD) is ubiquitously used in recommendation systems to estimate and predict values based on latent features obtained through the linear algebra



	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
L1	2					1			4	
L2		3								
L3								5		
L4	4			4		3				
L5			5							3
L6	2				1					
L7							4			
L8		1							5	
L9					2					
L10	4									2

(a) Kriging (b) Matrix Factorization

Figure 1.2: Paradigms to Solve for Unknown Point

approach, matrix factorization. While SVD has proven its predictive power in many applications such as user-movie recommendation [4] and face recognition [5], it does not explicitly allow to model spatial locations and spatial auto-correlation. This limitation is evident due to SVD using linear algebra only, which is too limited to describe spatial trends. Oblivious of location information, SVD has limitations in predicting variables that have strong spatial autocorrelation, such as housing prices which strongly depend on spatial properties such as the neighborhood and school districts. Therefore, this dissertation proposes a modification to recommender systems which augments the power of latent factor modeling using SVD with a kriging approach that gives spatial information appropriate special treatment.

We study two orthogonal paradigms: First, we employ the concept of kriging, also known as Gaussian process regression, which is a spatial statistics approach to interpolate metric values (such as the strength of an emotion or star ratings) on a map given locations of known value estimates. Intuitively, kriging estimates a degree of spatial autocorrelation, to estimate an unknown point given values in the spatial vicinity. We integrate the first concept with the latent feature learning capabilities of SVD. In doing so, we address the problem of modeling and predicting spatially autocorrelated data for recommender engines by integrating spatial statistics.

Unlike user location aware recommender engines that solely rely on a users explicit or implicit location which is utilized by uncovering a user spatio-temporal patterns or measurement from one location to another, kriging utilizes space in a more intelligent way. When fused together, kriging can give SVD spatial aurocorrelation information that cannot be determined using simple distance calculations that interpolate points on the theory that the closer a point is to the center of the unobserved point being predicted, the more influence it has in the averaging process.

The blending of SVD and kriging not only leverages the strengths of two completely different methods of interpolation, where SVD looks for hidden patterns within the data while kriging looks for spatial autocorrelations, but it also makes the algorithm more explainable compared to SVD alone. Kriging, unlike other methods such as Voronoi polygons or trend surface analysis, is an optimal spatial interpolation choice to integrate because it can provide unbiased predictions with estimates of the model errors, also known as the variance. Kriging utilizes a mathematical model to describe the spatial covariance, in the form of a variogram, which in its parameterized form has become the central tool of geostatistics [6]. When kriging is applied, the variance is calculated for every predicted point by determining how far the predicted value is from the covariance function. The kriging variance carries through our algorithms to explain how confident the prediction is, making machine learning more explainable. Thus, the combination of SVD and kriging allows one to determine how certain the algorithm is about any one prediction.

The variance plays a significant role in all algorithms described in Chapter 3. As described in Chapter 3.1, it is first used to aid the neural network to build a pattern that knows when to use values closer to kriging and when to use values closer to SVD. As described in Chapter 3.2, it is used as part of the algorithm output to directly determine the confidence of the prediction since SVD is fed into kriging. Lastly, as described in Chapter 3.3, the variance is used as part of the output to indirectly determine the confidence of the prediction since kriging is fed into SVD. Although the variance here is not utilized directly, our experiments show that its use in recommendations increases the accuracy significantly.

1.2 Field of Study Overview

In this chapter we describe in detail the paradigms used to improve upon each method individually.

1.2.1 Recommender Systems

With the new age of of e-businesses rapidly emerging and the explosive growth of information and consumer products available to users on the Web at the click of a button, recommender systems have been developed as an intelligent tool to alleviate the overload of options users have when selecting a movie to watch or adding an item to their cart. Inundated with choices which lead to users making poor decisions, the goal of a recommender system is to generate suggestions relating to various decision-making processes about new items or products that will actually be desirable to a user. Their purpose is to provide efficient and tailored solutions in e-commerce domains, that benefit both the customer and the retailer. When recommender systems are utilized, which are typically in an e-commerce setting, the engine offers users a list of ranked items to predict the most suitable products or services based on user preferences. This not only gives a user a personalized experience, but it also helps direct them to the items they desire most, thus leading to an increase in sales for businesses.

In today's world, recommender systems have been utilized to many different e-businesses such as Netflix, Amazon, and lastFM [7]. The reason for this is because recommender systems not only generate extra revenue from the purchase of recommended items but also generated a notable amount of additional revenue to business by introducing shoppers to new categories from which they then continued to purchase [8]. A great number of approaches for recommender systems have bee developed since the mid 1990's and are more recently categorized into the following categories [7]:

- **content-based** the user is recommended items that are content-similar to items that other users already liked.

- **collaborative** the user is recommended items that people with similar tastes and preferences have liked in the past.

- **hybrid** the user is recommended items based on a combination of both collaborative and content-based methods.

It has become such an integral part of e-commerce that in October 2006 Netflix announced that they were to hold a contest releasing a large dataset to improve the state of the current recommendation system which helps recommend movies to users. This competition attracted over 20,000 participants from 167 countries and initiated a great amount of research activity in this area, highlighting the importance of recommending items to users [9, 10]. This competition led to the birth of a matrix factorization approach, "Funk" Singular Value Decomposition (SVD) [11] which falls into the collaborative filtering category, the most successful category of recommender systems [7].

The recommender system technology has also been applied to other kinds of media such as music recommender engines due to the emergence of online music shops and music streaming services Spotify and Pandora. Recommender systems can be applied in this domain to recommend artists, albums, songs, genres, and radio stations to users [12]. Recommender systems have also been used in social media recommendations on social media websites where people create content, annotate posts with tags or "likes, make comments, and join groups to connect with other like minded people. The recommender system can be applied in two ways; 1) recommending users of social media content and 2) recommending users to other users. They play a crucial role in enhancing user engagement which leads to a successful social media website or app [13]. In addition, users can discover and explore their physical surroundings when using mobile location-based recommender system from GPS. This application for recommender systems can 1) recommend venues of particular categories top attend, 2) recommend the next place that a user may like to visit, 3) recommend new places that users have yet to visit, 4) recommend routes that users may like to take as they navigate a particular space and 5) recommend advertisements that may appeal to users by

using a users rich history of preferences *and* their current location [14].

Since the re-emergence of SVD from the Netflix Challenge, there have been many enhancements to the algorithm working to improve various aspects of the algorithm. One improvement to SVD is to use active learning for eliciting ratings for a user to get better recommendation due to the acute challenge in the lack of information and extreme sparseness of a rating matrix. [15] combined the classic matrix factorization method with a specific rating elicitation strategy to best approximate a given matrix with missing values in order to create a higher density matrix. Another approach mitigating the same problem of the ever increasing sparsity of a matrix designed a new hybrid model by generalizing a contractive auto-encoder paradigm into the matrix factorization framework with good scalability and computational efficiency, which jointly models content information as representations of effectiveness and compactness, and leverage implicit user feedback to make accurate recommendations [16]. [17] developd a recommender system algorithm based on a factorized matrix composed of user preferences associated to the movies' genres/categories. The study found that the advantage of using such a user-genre matrix factorization model is that it requires less computational resources, as the matrix will be less sparse and at lower dimension, which is a fundamental problem that may reduce the quality of the predictions generated by recommender systems. Other enhancements extend to temporal dynamics where the SVD model could trace the time changing behavior throughout the life span of the data. The solution the study adopted was to model the temporal dynamics along the whole time period, allowing the model to thoughtfully distinguish short-term factors from lasting ones. In essence, the algorithm modeled the way user and product characteristics change over time, in order to distill longer term trends from noisy patterns [18].

While there have been many improvements to SVD to help e-commerce businesses tailor suggestions to users, our approach takes the algorithm one step further by combining the collaborative filtering recommender systems, SVD, with kriging methods which incorporate spatial autocorrelations.

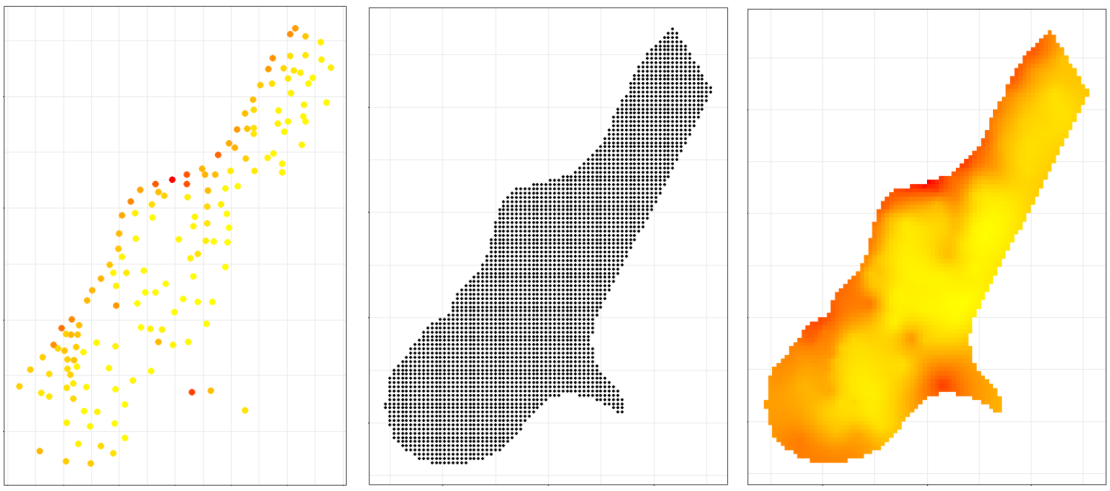
1.2.2 Spatial Statistics

No model can perfectly describe the natural or social world and any technique used for interpolation will results in some amount of error. It is here where spatial statistics, unlike other methods such as Voronoi polygons or trend surface analysis, is an optimal choice because it can provide estimates of the model errors. Even now, spatial statistics is a popular method in environmental sciences [19, 20]. Many scientists want to measure properties of interest to them (i.e. nutrients in soil, air pollution, rainfall) over a large continuous areas over which they often have only sparse observed data at limited numbers of places [6].

In this dissertation, the spatial statistics method, kriging, is widely studied to enhance SVD. This method was chosen over the spatial statistic methods geographically weighted regression (GWR) or inverse distance weighting (IDW) because while the former is a useful exploratory tool, its usefulness in making predictions is controversial and kriging can be better in capturing the spatial structure of the original data [21, 22]. The latter depends soley on distance relative to the unobserved location while kriging only takes into consideration distance between the onbserved and unobserved location, but also the spatial structure amount all observed point pairs.

Example 1. *To illustrate the purpose of kriging, we borrow an example dataset of measured levels of the heavy metal zinc in soil samples [23]. This example uses 155 point observations as depicted in Figure 1.3a, colored by intensity of the zinc concentration. The goal of kriging is to provide an estimate for all other points that are not sampled. Using a uniform grid to specify all unsampled points at a sufficiently fine resolution, Figure 1.3b displays* $31,000$ *points for which the zinc intensity is to be interpolated.*

To estimate the zinc value of one specific point kriging considers the proximity to sampled points to compute the empirical semivariogram from the sampled points. To compute the empirical semivariogram, we fit a regressive line in the space that compares the variance between different sampled data point pairs versus their distance as shown in Figure 4.1 for emotion values. Intuitively, the semivariogram is a function that estimates how distance affects the correlation between values. The weights for each sampled point are calculated



(a) Sampled Points (b) Unsampled Points (c) Kriging Results

Figure 1.3: Kriging Example

from the semivariogram to interpolate the unsampled points as shown in Figure 1.3c.

In environmental pollution, geostatistics has been applied to estimate and map potentially toxic substances in the environment. In 2006, a study by Komnitsas and Modis exemplified the strong application of geostatistics. A large study area was contaminated with arsenic and zinc south of Moscow, Russia. The study was aimed to map the contamination and assess the risk for agricultural soils in a wider disposal site containing wastes derived from coal beneficiation. Soil samples were collected from a 34 km^2 area and analysed by using geostatistics in order to produce risk assessment maps and estimate the probability of soil contamination [24]. Geostatistics has also been extended to the fisheries sub-field of environmental science. One study was conducted that mapped the abundance of shell fish in two regions of Georges Bank. The scientists quantified the average bed diameter of sea scallops using geostatistics. They analysed their count data geostatistically and mapped their estimates of species density. The authors were able to recommend strategies for sampling Georges Bank, and for managing the fishery in zones identified by their mapping so that the stocks of scallops are never severely depleted anywhere on the Bank [6].

In many of the use cases that apply geostatistics, the best that they can do is to estimate or predict in a spatial sense values between observed points. The geostatistics technique kriging is used to interpolate values of non-sampled locations from a collection of sampled areas and estimate the uncertainty surrounding the predicted value [25]. Kriging is not effective if there is little spatial autocorrelation among data points [26]. Following this intuition, for this dissertation, have utilize three different data sets that are autocorrelated, which can be exploited for predicting using a kriging estimate.

Chapter 2: Preliminaries

In this chapter, we present the existing concepts exploited by our algorithms used for our approach to combine recommender systems with spatial statistics. By selecting the appropriate hyper-parameters and models within each algorithm, we are able to utilize the strengths of each method to optimize the predicting power of recommender engines.

2.1 Spatial Statistics Algorithms

2.1.1 Ordinary Kriging

Typically, ordinary kriging (OK) interpolates values assuming stationarity and homoscedasticity, thus assuming no trend. [6]. In other words, the mean and variance across the entire spatial domain is assumed to be constant. This study incorporates OK which assumes a constant stationary function and is calculated using Definition 1:

Definition 1 (Ordinary Kriging Estimate)**.** *The kriging estimate is of the form:*

$$\hat{Z}(x_0) = \sum_{i=1}^{N} \lambda_i Z(x_i), \tag{2.1}$$

where $\hat{Z}(x_0)$ is the estimated value (specifically, housing prices in our use-case) at location x_0; $Z(x_i)$ are all other observed values at locations x_i; N is the sample size; and λ_i weights chosen for x_i.

The challenging part of Definition 1 is to find appropriate weights λ_i to define how much an observed value $Z(x_i)$ at location x_i affects the value $Z(x_0)$ to be estimated. For this purpose, different models (variograms) are used to choose the weight depending on the

distance $d(x_0, x_i)$ between the estimated location x_0 and the location of the predictor value x_i.

Definition 2 (Empirical Variogram)**.**

$$\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} [Z(x_i + h) - Z(x_i)]^2 \tag{2.2}$$

where $N(h)$ is the number of experimental pairs of distance vector h, and $Z(x_i + h) - Z(x_i)$ are the observed values of z at places x_i and x_i + h.

In order to build the appropriate models, kriging builds an empirical Variogram (Definition 2.2) which is calculated by substituting the distance between two observed points into Definition 2.2. The empirical variogram values provide information on the spatial autocorrelation of the dataset and can then be used to fit a model to ensure that kriging predictions have positive kriging variances [27]. The pairs are grouped by their Euclidean distance, and for each distance group, the corresponding empirical covariance (as defined in Definition 2.2) is given on the y-axis.

We can empirically observe whether there is a strong spatial autocorrelation within the data based on the empirical variograms, showing at what point the correlation drops and the covariance as distance increases. An experimental variogram that increases with increasing gradient as the lag distance increases signifies trend [6]. When trend is present within the data, the process should be modelled as a combination of a deterministic trend and spatially correlated random residuals from the trend [6]. Thus, in this dissertation, we also study and implement universal kriging where the mean will differ throughout the spatial domain while still assuming homoscedasticity, i.e., assuming that the variance is constant and is used when data has a strong trend.

To leverage the empirical variogram (Definition 2.2) for kriging, the values of λ_i in

Definition 1 are chosen by solving the following set of equations:

$$\forall 1 \leq j \leq N : \sum_{i=1}^{N} \lambda_i \hat{\gamma}(d(x_i, x_j)) + \psi(x_0) = \gamma(x_j - x_0), s.t. \sum_{i=1}^{N} \lambda_i = 1, \tag{2.3}$$

where $\hat{\gamma}(d(x_i, x_j))$ is a model of the semivariance between as defined in Definition 2.2 given their distance $d(x_i, x_j)$ and $\psi(x_0)$ is a Lagrange multiplier introduced for the minimization of the error variance. OK assumes a stationary stochastic process, i.e., having a constant (but unknown) mean.

The error variance of the predicted value, σ^2, can also be calculated which is defined as [26]:

$$\sigma^2 = \sum_{i=1}^{n} w_i \cdot \gamma_\theta(d_{0,i}) + \psi(x_0), \tag{2.4}$$

where $d_{0,i}$ is the distance between point x_0 and x_i, and $\psi(x_0)$ is the Lagrange multiplier.

2.1.2 Universal Kriging

Universal Kriging (UK) is used for non-stationarity when drift is present. It allows for non-stationarity thus allowing the mean to differ throughout the spatial domain while still assuming homoscedasticity, i.e., assuming that the variance is constant and is used when data has a strong trend. UK assumes that the prediction variable, $Z(x)$, can be written as a sum of deterministic trend component $m(x)$ and the spatially correlated random residual component $\epsilon(x)$:

Definition 3 (Universal Kriging)**.**

$$Z(x) = \sum_{i=1}^{k} a_i f_i(x) + \epsilon(x), \tag{2.5}$$

where a_i is the ith coefficient to be estimated from the data; f_i is the ith basic function of spatial coordinates which describes the drift; k is the number of functions used in modeling the drift; $\epsilon(x)$ is a spatially correlated random residual.

The mean, $m(x)$, can be a function of the coordinates in linear, quadratic or higher form. For this dissertation, the UK considering linear trend is used for depth spatial interpolation since it represents a planar model using an easting and a northing term.

$$m(x, y) = a_0 + a_1 x + a_2 y \tag{2.6}$$

where a_i is the coefficient to be estimated from the data, and x and y are the spatial coordinates in formal UK. With the appropriate equations mentioned above, we can calculate an empirical variogram to determine the weights of the observed values. Unlike OK, where the variances are calculated by the observed location values x_i and $x_i + h$, in UK they are calculated by the observed location residuals from the trend. In contrast to the mean in OK which assumes to be constant, the mean in UK is a function of the spatial coordinates and the local drift which is represented by a linear equation of the independent variables.

2.1.3 Kriging Models

To build a model $\hat{\gamma}$ of the empirical semivariogram as required in Equation 2.3 different parametric models can be used. For this study, the Spherical, Gaussian and Matern model were interchangeably chosen since they best fit the empirical semivariogram in different data sets. The Spherical model is given by the following expression:

$$\gamma(h) = \begin{cases} c_0 + c[(3h/2d) - (1/2)(h^3/d^3)], & 0 < h \leq d \\ c_0 + c, & h > d \\ 0 & h = 0 \end{cases} \tag{2.7}$$

where d is the distance parameter or range, c_0 is the nugget variance, c is the variance

of spatially correlated component, and h is the lag interval. The Matern model is given by the following expression[28]:

$$\gamma(h) = \frac{1}{2^{\theta_2-1}\Gamma(\theta_2)}(\frac{2d\sqrt{\theta_2}}{\theta_1})^2 K_{\theta_2}(\frac{2d\sqrt{\theta_2}}{\theta_1}), \theta_1 > 0, \theta_2 > 0 \tag{2.8}$$

where h is the distance between the observed point and unobserved point, $K_{\theta_2}(\cdot)$ is the modified Bessel function of order θ_2, θ_1 (the "range" parameter) controls the rate of decay of the correlation between observations as distance increases, and θ_2 controls the behavior of the autocorrelation function for observations that are separated by small distances [29].

The Gaussian model is given by the following expression:

$$\gamma(h) = \begin{cases} c_0 + c[1 - \exp(-(h/d)^2)], & 0 \leq h \leq d \\ 0, & \text{otherwise} \end{cases} \tag{2.9}$$

where h represents the lag interval, c_0 is the nugget variance, $c_0 + c$ represents the sill, d represents the range.

2.2 Collaborative Filtering Algorithms

2.2.1 Baseline Estimate

There are many sophisticated recommender engine algorithms that have been developed to help users navigate to products that they may be interested such as Neighborhood based and memory based collaborative filtering for recommender systems. Neighborhood based collaborative filtering algorithms were some of the earliest algorithms developed under collaborative filtering due to their relative simplicity and intuitiveness. The intuition behind this method is that similar users show similar patterns of rating behavior and similar products have similar ratings [30]. Typically, for use cases that utilize collaborative filtering, the data depicts large user and product effects where some users give a lower rating than

other users or some items receive a higher score than others [31]. As described by [31], it is typical to adjust the rating score for these effects by subtracting the user and item bias from the mean which is known as the baseline estimate. This estimate can extract some information that is independent of neighborhood influence. The baseline estimate for this algorithm can be described as the following:

$$b_{ui} = \mu + b_u + b_i \tag{2.10}$$

Where μ is the average ratings, bi and bu indicate the observed deviations of item i and user u respectively from the average [32], and the regularization term avoids overfitting by penalizing the magnitudes of the parameters [31].

2.2.2 Singular Value Decomposition Algorithm

Matrix factorization has become the state-of-the-art technique for recommendation systems as popularized by the "Funk" SVD method used in the 2015 Netflix Challenge [11]. The idea of matrix factorization is to represent a data matrix by multiple (much-) lower dimensionality matrices of latent factors. Then, multiplying of the factorized matrices yields an approximation of the original matrix that allows to estimate missing values within the matrix. The concept of truncated SVD is to keep only the k latent factors having the most information.

The idea behind such latent factor models is that a large portion of a $n \times m$ data matrix R may be highly correlated, and can be approximated by matrices of a lower rank. For example, a matrix that stores a rating value for each of n user combinations, and each of m items may have high redundancy: Some users may be very similar to each other, while some items may exhibit similar reactions to topics.

Definition 4 (Singular Value Decomposition (SVD))**.** *Let R be a $m \times n$ matrix. SVD factorizes R into three components:*

$$R = U\Sigma V^t, \tag{2.11}$$

where U is a $m \times m$ matrix, V is a $n \times n$ matrix, and Σ is a $m \times n$ diagonal matrix containing the singular values of R in descending magnitude. Intuitively, Σ can be seen as a scaling matrix, that scales dimensions according to their variance/information. U and V can be seen as rotation matrices, which rotate the data space in the directions of the highest variance components.

The idea of truncated SVD is to simply truncate all but the k highest variance components, discarding the rest of the matrices. This not only allows for an efficient approximation of R, but it also allows to discard detailed and potentially overfitting information.

Definition 5 (Truncated SVD)**.** *Let R be a $m \times n$ matrix and let $k \leq n, m$. Truncated SVD factorizes R into three components:*

$$R \approx U_k \Sigma_k V_k^t =: \hat{R}, \tag{2.12}$$

where U_k is a $m \times k$ matrix, describing each line of R by k latent features, V_k is a $n \times k$ matrix, describing each column of R by k latent features, and Σ_k is a $k \times k$ diagonal matrix retaining the largest singular values of Σ. Σ can be seen as a scaling matrix, that scales dimensions according to their variance/information. U and V can be seen as rotation matrices, which rotate the data space in the directions of the highest variance components.

The matrix factorization method provides a clear way to observe the correlations between rows and columns in a one-shot estimation to the entire matrix [30]. To leverage this factorization for prediction, we rebuild an approximation $\hat{R}$ of data matrix R, by multiplying the truncated matrices. In the resulting matrix, each predicted value, $\hat{r}_{ij}$, in $\hat{R}$ is calculated by the dot product of the i^{th} row factor u_i of U_k and the j^{th} column factor v_j of V_k [30]:

$$\hat{r}_{ij} \approx \sum_{k=1}^{k} u_{ik} \cdot v_{jk} \tag{2.13}$$

Once the latent vectors for rows and columns are established, it is simple to use equation (2.13) for estimating the values of R. Since SVD cannot directly compute estimates with an incomplete matrix, a simple approach is to take the average of the columns or rows of the observed values and input them into the missing ratings. However, this approach can be biased. By using the truncated SVD algorithm with Stochastic Gradient Descent (SGD) which is a convex optimization method, only observed values will be taken into consideration while avoiding overfitting through a regularized model [11]. Overfitting is common when a training dataset is small. Through regularization, the models incorporate a bias which discourages extremely large values of the U_k and V_k coefficients to promote stability. SGD is used in truncated SVD as an approach to minimize error, looping through all observed values, computing the prediction error, and then using the error from each observed value to update the k entries in row i of U and k entries in column j of V. Details of this SGD approach for approximating missing values in truncated SVD can be found in [30, 11].

While truncated SVD has proven its predictive power in many applications such as user-movie recommendation [4] and face recognition [5], it does not explicitly allow to model spatial locations and spatial auto-correlation. This limitation is evident due to truncated SVD using linear algebra only, which is too limited to describe spatial trends. Therefore, in the following, we propose a modification to kriging which augments the power of latent factor modeling using truncated SVD with a kriging approach that gives spatial information appropriate special treatment.

Chapter 3: Algorithms Developed

This chapter describes the three algorithms that were developed and utilized in this dissertation in detail. Each sub-chapter explains the method used for combining SVD with kriging, why the algorithms were integrated in such a way, and the data set that algorithms were applied to. These algorithms are explained further in Chapter 4, 5, and 6 in the form of individual papers.

3.1 Hybrid Neural Network (HNN)

The first integration of SVD with kriging in this dissertation is through the use of a neural network and a simple multi-linear regression model. These algorithms serve as a proof-of-concept to whether combinging the two paradigms can enhance the prediction power of either model individually. The hybrid approaches combine spatial statistics based on auto-regressive models on one hand and classic matrix factorization-based recommender systems on the other. The algorithm built integrates distance, spatial arrangement of nearby observations, and the variance (from ordinary kriging) and utilizes hidden patterns (from matrix factorization) that the human eye cannot detect within the data to improve the prediction of unobserved values.

Microblogs are utilized as the data set in this study. Microblogs are used by millions of users to express their emotions, such as joy, surprise and anger, on a plethora of different topics. It was observed that for the same topic, different places may exhibit different emotions for identical topics. By learning, modeling and predicting emotions on various topics and in different cities using the hybrid algorithm, the goal of this work was to create a proof-of-concept, naive algorithm that would either prove or disprove the success of integrating spatial statistics with SVD. Through experimentation, we found that there is

usefullness and validity in fusing the two very different algorithms together, and essentially receiving the best of both world.

3.2 Singular Value Decomposition with Regression Kriging (SVD-RK)

The proof-of-concept algorithm, HNN, proved to be effective in predicting emotions, thus the blending of the algorithms was further explored by integrating them in a more intelligent way. Instead of fusing the outputs of each algorithm into a neural network as a naive approach, the improved approach integrates the latent feature learning capabilities of SVD with kriging by feeding SVD as an independent variable into a regression kriging approach, which is referred to as SVD-RK. This method allows the kriging weights to be calculated using SVD outputs and residuals as the covariates. The study of this algorithm used a different data set that was more spatially correlated. Since real estate data can be spatially correlated the Zillow data set was utilized to study the proposed algorithm with.

Our study, described in detail in Chpater 5, shows that latent house price patterns learned using SVD are able to improve house price predictions of ordinary kriging in areas where the variogram of an area depicts a degree of trend. However, for the areas that are well described by a variogram, the study found that ordinary kriging is sufficient in making housing predictions. However, we wanted to attempt an improvement from ordinary kriging for these spatially correlated areas thus another approach developed was to feed the results of SVD into a geographically weighted regression (GWR) model to outperform the orginary kriging approach. Further details on GWR can be found in Chapter 5.5.3 In exploring these approaches, this study addresses the problem of modeling and predicting spatially autocorrelated data for recommender engines using real estate housing prices by integrating spatial statistics.

This algorithm also exhibits explainable machine learning. Since SVD outputs and residuals are directly fed into kriging as covariates to calculate the spatial correlation between

points, the kriging portion of the algorithm can then make direct interpolation for the data points along with the variance from the optimal curve function. By fusing the algorithms in this way, SVD-RK produces a certainty variable which can be used to understand the prediction being made. If a prediction has a high variance, than a user will know it validity is weak, while if a prediction has a low variance, than the user will know that the prediction reliable.

3.3 Kriging with Singular Value Decomposition (KARS)

The third integration technique of SVD and kriging is similar to Chapter 3.2 except, instead of feeding SVD outputs into kriging, ordinary kriging is fed into SVD. While SVD-RK, described in 3.2, enhances areas where the variogram depicts trend while leaving it up to ordinary kriging to make predictions when the variogram fits the data well, the algorithm developed in this chapter focuses on enhances the prediction power of points when the variogram fits the data well. In addition, this method utilizes the variance from kriging indirectly and through experimentation, it was found that using variance as a threshold can significantly influence the reduction in error.

The data used for this chapter is derived from the Yelp Data Challenge. This chapter successfully shows the potential OK-SVD has on a data-driven approach to urban planning and for businesses corporations alike, instead of traditional market analysis techniques. While recommender systems tend to be utilized to personalize user experience in e-commerce, when SVD is leveraged with spatial statistics, it can be applied to two different applications; 1) urban planning to determine the optimal stores for a vacant lot and 2) business market analysis in order to find optimal locations for a new store. This method allows users for both applications not just to have a prediction to use in decision making, but also an accuracy measure (the variance) which allows the process of recommender systems, in an urban planning sense, to be more explainable in comparison to just SVD alone.

Chapter 4: Emotion Predictions in Geo-Textual Data usingSpatial Statistics and Recommendation Systems

Abstract

Microblogs are used by millions of users to express their emotions, such as joy, surprise and anger, on a plethora of different topics. We observe that for the same topic, different places may exhibit different emotions for identical topics. The goal of this work is to learn, model and predict emotions on various topics and in different cities. For this purpose, we propose a hybrid approach which combines spatial statistics based on auto-regressive models and Kriging on one hand and classic matrix factorization-based recommender systems on the other. The algorithm built integrates distance and spatial arrangement of nearby observations (from Kriging) and utilizes hidden patterns (from matrix factorization) that the human eye cannot detect within the data to improve the prediction of unobserved values. Our experimental evaluations, using millions of tweets across the United States, show that our hybrid approach outperforms individual approaches based on matrix factorization and Kriging alone. This case study shows the potential of combining spatial statistics methods such as Kriging with machine learning solutions to support knowledge discovery on spatial data.

4.1 Introduction

Emotions have the power to influence and determine the outcome of many major decisions our world makes today. From our personal life, to work, to even politics, emotions play a huge role in how we live our life. According to Nobel Laureate Herbert Simon, "In order to have anything like a complete theory of human rationality, we have to understand what role emotion plays in it" [33].

A study at the Stanford School of Business examined if personal emotions played a strong role in political decision making by comparing voting trends to the success and failures of local college football teams. The results showed that if the local team won, their opinion of the incumbent party was more positive and if they lost, their opinion tended to be more negative. The conclusion was that a voter's decision is dependent on their emotional state of mind, even when those events are entirely unrelated to government activities [34].

With the explosion of Web 2.0 and social media, a vast amount of content has become available. For example, Yelp users comment on their experiences with businesses; Instagram users post images from their daily life; and Twitter users express themselves in short 280 character messages. Users are communicating what they feel about a restaurant or an event that has occurred in their life. Users are freely creating peta-bytes of data that is just waiting to be harnessed. By collecting all of this data and analyzing the emotions of the user, feelings on different topics can be derived. With enough of this kind of data, we can collectively discover the emotions of our society.

However, data collection can be difficult when analyzing thousands of specific locations. As the data is divided into cities, we are faced with a lack of data in some cities. Our goal is to utilize machine learning and geostatistics to interpolate the emotions of these undocumented cities. In this study we have developed a new algorithm leveraging Singular Value Decomposition (SVD) which uses matrix factorization to fill data gaps and predict emotions for cities where only a limited amount of data is available. The algorithm also utilizes Kriging to exploit spatial auto-correlation and interpolate emotions by using nearby data points. Our algorithm combines these two methods to estimate how cities emotionally react to current events in the years 2018 and 2019 even if data is missing from specific cities of interest.

The purpose of this study, is not to find another technique to detect emotions within texts but rather to utilize one emotion detection method to begin filling in data gaps. The purpose of this work is to learn, model, predict, and map emotions of cities for different topics using an algorithm that combines geostatistics and matrix factorization.

We address the problem of modeling and predicting emotions using large sets of geo-textual data and exemplify the approach using microblog data. In more detail, given a topic t (specified by a "hashtag" keyword), given a location l (specified by a place name), and given an emotion e (specified as one of eight emotions), the problem approached in this work is to predict the strength of emotion e for topic t in location l. For this purpose, we study two orthogonal paradigms: First, we employ the concept of kriging, also known as Gaussian process regression, which is a spatial statistics approach to interpolate metric values (such as the strength of an emotion) on a map given locations of known value estimates.

Intuitively, kriging estimates a degree of spatial autocorrelation, to estimate an unknown point given values in the spatial vicinity. By taking into account Tobler's first law of geography [35], we assume that the emotional value of current events can be related in certain locations and can have a moderate spatial autocorrelation as shown in Figure 4.1. This figure shows the average variation for each pair of points for the topic Starbucks and the emotion anger. At 75 miles, the autocorrelation trend line becomes independent which means there is no longer any spatial relationship between the closeness of the data points.Kriging can adapt to different degrees of spatial autocorrelation for different topics. For example, the emotions about services, such as food services, where everyone in town is negative about a burger restaurant, may yield a semivariogram where only tens of kilometers are required to quickly lose spatial autocorrelation as shown in Figure 4.1. In contrast, for political issues, it may be common to have very diverse emotions within a city, but with a trend among larger areas such as states and countries. In this case, the variogram would indicate less autocorrelation at low distances, while having more autocorrelation remain at larger distances.

The second concept we employ is matrix factorization using singular-value decomposition (SVD), a state-of-the-art approach in recommendation systems. Intuitively, SVD takes as input a matrix M of emotion values for (location, topic) pairs. It estimates an emotion e at location l for topic t by extracting latent features corresponding to the set of k largest eigenvalues of M. By combining both approaches, we can augment the latent features (of

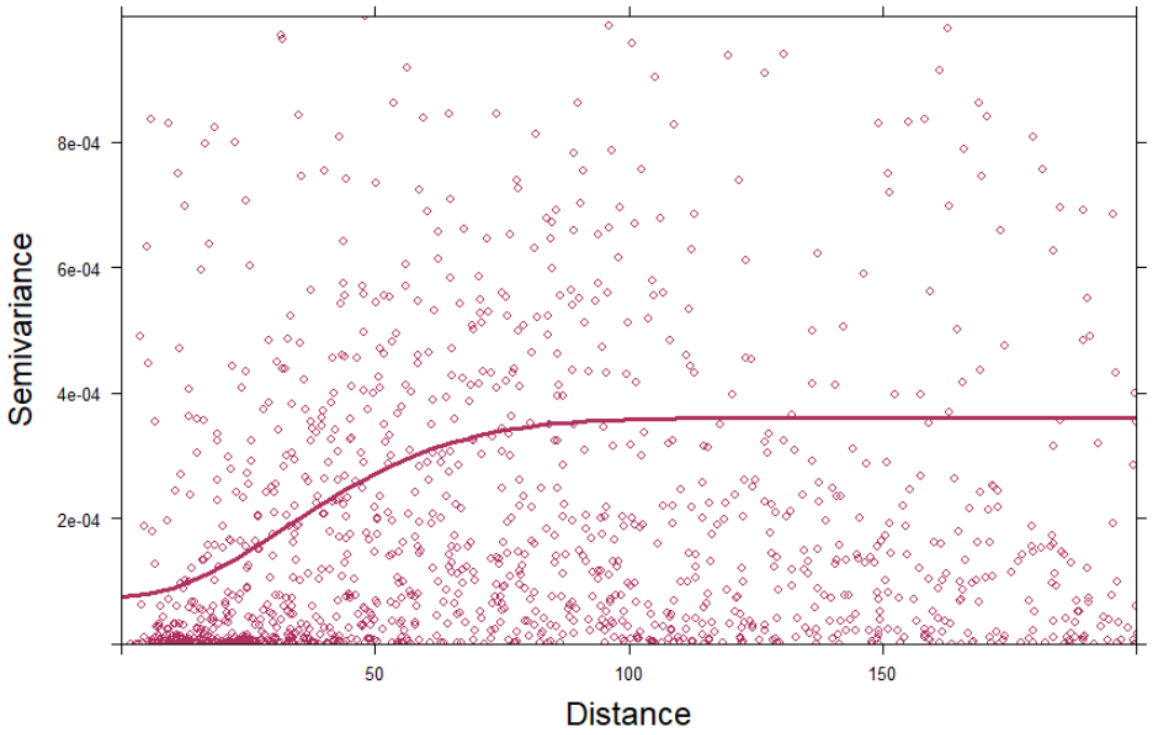


Figure 4.1: Autocorrelation (in miles) for Anger Sentiment of Starbucks

both topics and locations) learned by SVD, with spatial auto-correlation, thus allowing to predict that places close to each other are likely to have similar emotion towards the same topics.

The main contribution of this work is not only to experimentally let kriging compete with SVD, but rather, to have kriging complement SVD. By combining predictions made by kriging and SVD, and by utilizing the variance information of kriging, which indicates the level of confidence that kriging has for a specific spatial location, we are able to propose a hybrid approach which outperforms each individual approach.

This hybrid approach can be applied not just to emotional values but other domains such as real estate pricing which heavily relies on spatial arrangement. There have been a number of studies conducted that utilize spatial statistics to predict the pricing of houses [36, 37, 38]. These studies have researched kriging in detail within the real estate domain. One author

explained that the actual variations in prices and property values can considerably diverge from the values predicted by kriging, since data can be scarce and unevenly distributed. In fact, it is noted in this study that geostatistical methods may facilitate the interpretation and analysis of the causes of price variations rather than serve as a tool for exploring the nature of the analyzed phenomenon [36]. However, with the integration of SVD, hidden patterns within the prices are revealed aiding the algorithm to make stronger and well-rounded predictions. For the scope of this paper, we have decided to use emotions from tweets as the value for our algorithm to predict since Twitter data is widely available. With a larger data set to train and test, we were able to ensure that the algorithm could handle a considerable stream of data.

The remainder of this work is organized as follows. Section 4.2 surveys related work on emotion prediction, recommender systems and kriging. Our approach to combine the best of both these worlds for emotion prediction is presented in Section 4.3. Our experimental evaluation to quantitatively evaluate kriging, SVD, and our proposed hybrid approach is presented in Section 4.4. Finally, we conclude in Section 4.5.

4.2 Related Work

The goal of this work is to fusion two elements: 1) recommender system that incorporate an emotional dimension and fill gaps of missing data 2)geospatial statistics to integrate the physical space. The concept of integrating recommender systems (RS) with other elements has been well researched and there is a significant amount of related work in this field which will be discussed in this section.

4.2.1 Emotion-Aware Recommender Systems

Most recommender systems use collaborative or content-based filtering techniques allowing the customer to sift through items that may not be of interest and to help them navigate to personalized and relevant items faster such as truncated SVD (as described in section 2.12 or k-nearest neighbors [30]. There has been extensive research on integrating emotions

into recommender systems for a wide range of applications. Wakil et al. [39] improved a movie recommender system by integrating emotional values. They combined the nearest neighbor content-based method and a rule-based collaborative method into their algorithm. The emotional values were collected from explicit feedback from users. The researchers concluded that their results provided better movie recommendations to users because it allowed them to create a relationship between their emotional state and the recommended movies. Deng et al. [40] conducted a study on improving music recommendation systems through emotion. The study collected their own emotion lexicon and applied it to microblogs. These emotional values were then applied to their algorithm derived from rule-based collaborative filtering [40]. This study also showed an improvement in accuracy when compared to recommender engines that did not incorporate emotion. Recommender systems and emotions can also be applied to other fields as well. Recently, researchers applying this idea to health care managed to improve the matrix factorization technique by revising all user ratings through positive and negative sentimental offset [41]. The researchers concluded that their recommender engine, iDoctor, showed higher accuracy in recommending medical services to its users better than the baseline algorithms, item-based collaborative filtering and user-based collaborative filtering.

4.2.2 Spatially-Aware Recommender Systems

Recommender systems have also been improved by utilizing spatial elements. One approach by [42] used Gaussian spatial processes, also known as Kriging, to predict which exhibits in a museum users would enjoy seeing next. They compared their process with a rule-based collaborative filtering recommender system, since matrix factorization was not yet used as the state-of-the-art method for recommendation systems. The study did not combine the two approaches but rather compared one to the other and found that the kriging-based approach had a higher accuracy to the collaborative filtering method [42]. More recently, a study was conducted to develop a recommendation engine that inferred interest in activities for a given time, around the current location of a user [43]. Using matrix factorization to

infer temporal preferences and a personal function region to infer spatial activity preferences, the researchers improved Point-of-Interest (POIs) recommendation engines. In 2016 a group of researchers made additional improvements to recommender engines [44]. Unlike the method from [43], this study utilized spatial information as direct features of the matrix factorization model [44]. Another approach by Yang et al. [45] used derived spatial information such as the distance from a users home location as explanatory variables to improve the recommendation of sites to users in a location-based social network. All of these approaches have in common, that they enrich the recommendation system with simple explicit and task-specific spatial features. Our approach goes an additional step ahead by combining recommender systems with Kriging methods.

4.2.3 Kriging for Emotion Prediciton

Klettner et al. [46] propose a framework, the "EmoMap", for crowdsourcing emotion data from individuals. The main focus of this work was the crowd-sourced collection of emotion information such as comfort, safety, and attractiveness in different locations of an urban environment. Therefore, the focus on this work is not on the emotion of textual topics, but on the subjective feeling of different locations in a city. While kriging is used for visualization, there is no evaluation on whether the resulting predictions are accurate.

4.3 A Hybrid Approach for Emotion Prediction

In this section, we describe our approach to predicting emotions of a specified location for a specified topic as shown in Figure 4.2. For this purpose, we first present our approach to extract topics from geo-textual data in Section 4.3.1, proposing both a traditional approach based on explicit topics (keywords, hashtags), and a latent topic modeling approach.

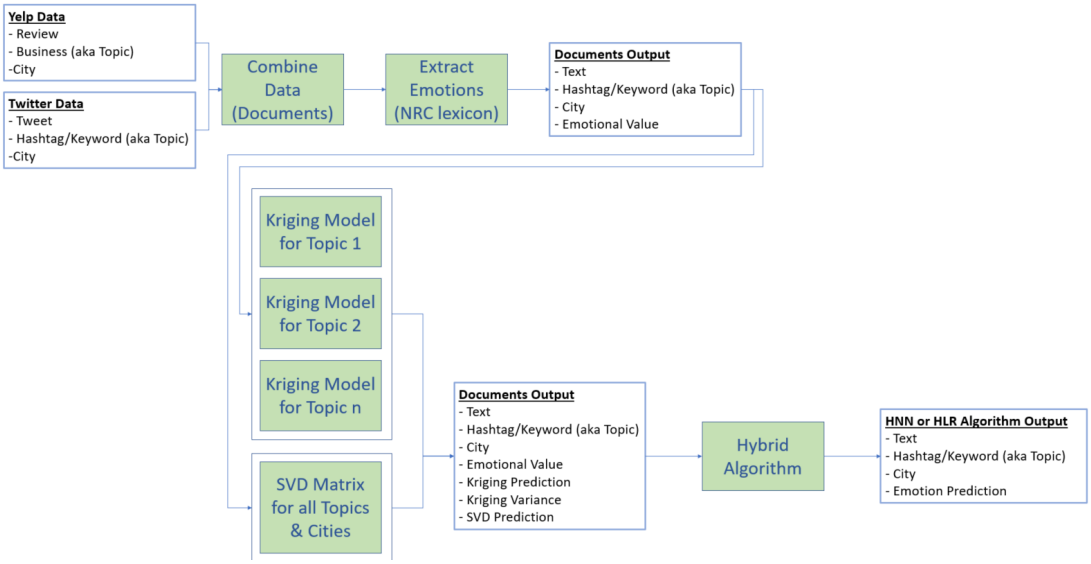


Figure 4.2: Flowchart Diagram for Emotional Predictions

4.3.1 Geo-Textual Topic Extraction

To analyze the emotions on different topics in different places, our approach utilizes Twitter microblogs and Yelp user reviews within the United States. The tweets were collected through web scraping over a period of five months from September 8, 2018 to February 15, 2019. The user reviews were derived from Yelp's 2018 Dataset Challenge [47]. We collected data in 257 cities with 35 over-arching topics. These documents were all tokenized by word and pre-processed to remove punctuation, numbers, and stop-words (extremely common words such as "the" or "a").

Explicit Topic Modeling

To extract topics from geo-textual data, we manually specified common over-arching topics that were derived from both common business names from the Yelp data set (such as "Starbucks" and "Wholefoods") and trending hashtags or specific word combinations on twitter (such as #christmas and "governmentshutdown") which were manually selected

based on current events at the time of collection.

For each topic, both Twitter and Yelp datasets were used to obtain emotions from both datasets using the NRC lexicon which is further explained in Chapter 4.3.2. Thus, the set of documents related to a single topic (from which emotions will be extracted in the next section) is the combination of scraped tweets and the Yelp user reviews.

There were 2,205,476 unique documents (tweets and Yelp reviews) that were used for the study once both data sets were combined and pre-processed. The Twitter data used in this study was relatively recent, as they reflect current events that have occurred in years 2018 and 2019. Over-arching topics of the tweets are shown in Figure 4.3 and include, Brett Kavanaugh, Hurrican Florence, #metoo , Jamal Khashoggi, Google Home, and the Bezos Scandal.

Latent Topic Modeling

To verify that our model was not over-fitting, we created an artificial data set. First, we created arbitrary topic id's and did so by employing a latent topic modeling approach for which we used a collection of 5,328,940 geo-tagged tweets collected between February 7, 2014 and July 31, 2016. To avoid spatial inconsistencies in the model, this study only considered tweets from within the United states. Second, we needed to create arbitrary location id's. In order to allow for larger geo-location groupings, we broadened the location boundaries of each city, since many cities contained only a few tweets. We created our own partition of space by expanding the spatial cells of the given latitudes and longitudes by simply rounding the latitude and longitude values to the nearest tenth (i.e. 60.3587 to 60.4). This allowed us to group more data points in each spatial grouping.

For this dataset, we did not select topics explicitly. Instead, we applied topic modeling using Latent Dirichlet Allocation (LDA) [48] – a generative probabilistic model which assumes that each tweet is a mixture of underlying (latent) topics, and each topic has a (latent) distribution of more and less likely keywords. This technique allows us to organize and summarize documents (such as tweets) at a scale that would be impossible by human

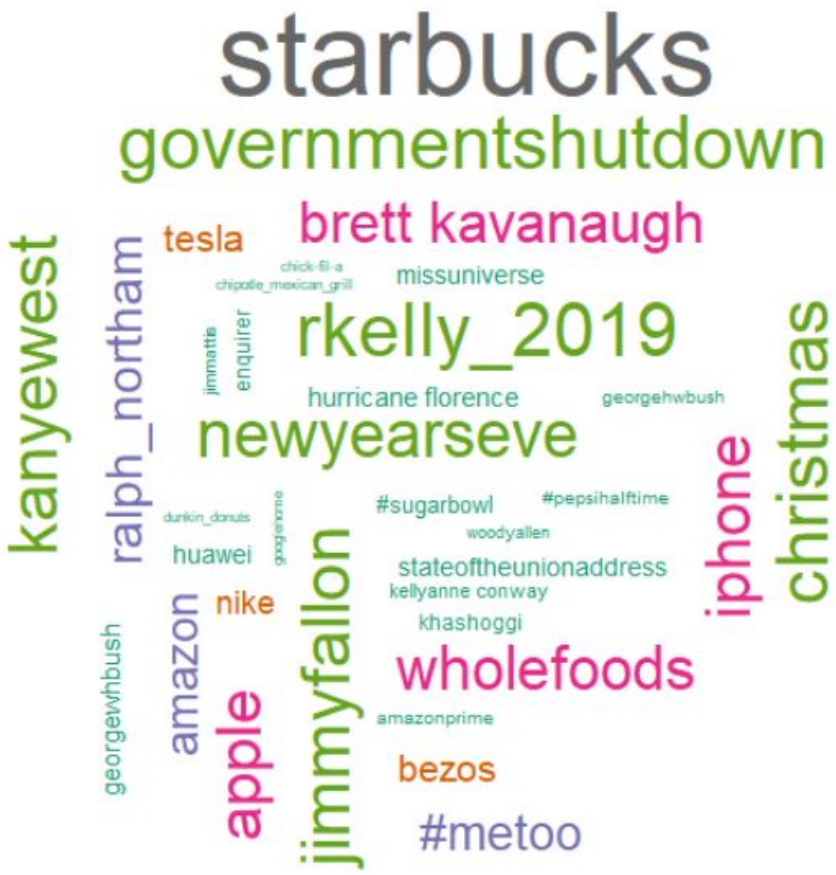


Figure 4.3: Topics Word Cloud

annotation. A graphical representation of our LDA model is shown in Figure 4.4[1]. We used a uniformly distributed vector α of length K to parameterize the apriori distribution of topics. The parameter K corresponds to the number of latent topics we want to find. When a tweet is created, we assume that its topics are chosen following a *Dirichlet distribution* having distribution parameter α which we use to obtain a topic distribution θ for each of our M tweets. Thus, the large plate in Fig. 4.4 corresponds to a set of all M tweets, each having a topic distribution θ drawn randomly (and Dirichlet distributed) from α. To infer the topics of tweets, LDA uses a generative process similar to Monte-Carlo sampling with

[1]Source: https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

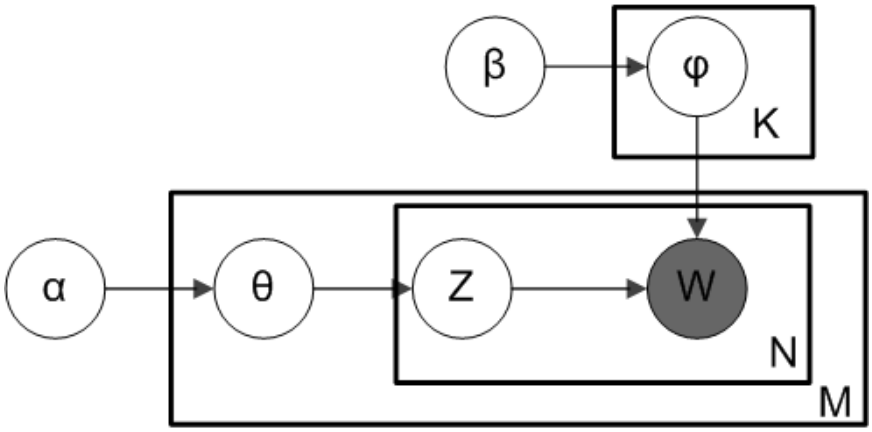


Figure 4.4: Graphical Model in *plate notation* of LDA-based topic modeling. Boxes represent entities (M Tweets, N keywords within a tweet, K latent topics). Nodes correspond to random variables, shaded nodes are observable random variables, and arrows indicate stochastic dependencies.

iterative refinement of the distributions α and θ. More details on LDA can be found in [48].

After running LDA, we used the topic distribution for each tweet and assigned each tweet to the topic with the highest probability. Once this collection of tweets was processed and cleaned, LDA was applied to the remaining 3,767,411 tweets. Since this dataset is so large, in order to successfully apply LDA on the training set, we transitioned from our local Windows environment to the virtual Windows environment mentioned in the beginning of Chapter 4.4.

Regardless of using the explicit or latent topic modeling approach, for each topic and location pair, we obtain a set of tweets and Yelp reviews relevant to a particular topic and location. In the next Chapter, the task is to associate each of these microblogs with emotion information.

4.3.2 Emotion Detection

Sentiment analysis has become a well-known technique in mining opinions from texts. It has been used extensively in the commercial world to better understand consumer products

and services. Emotions are viewed as nuances of sentiment analysis, moving from a two dimensional view to an eight dimensional view (anger, anticipation, disgust, fear, joy, sadness, surprise, trust). There have been many complex methods to detect emotion including lexicons, wearable physiological sensors - a computing device integrated with various sensors [49] and the EQ-Radio - a device that can detect a person's emotions using wireless signals [50].

Emotions can be expressed through facial expressions, heart rate, and even breathing patterns. However, for this study, we implemented the NRC Emotion Association Lexicon created by the experts of the National Research Council of Canada (NRC) which focuses on how emotions manifest in the written word [51]. The NRC Emotion Association Lexicon lists English words and their associations to eight basic emotions; anger, fear, anticipation, trust, surprise, sadness, joy, and disgust [2]. The lexicon is based on Robert Pultchik's theory of eight basic emotions where the radius indicates intensity — the closer to the center, the higher the intensity as shown in Figure 4.5.

To annotate a substantial list of words, Mohammad et al. used a crowd-sourcing platform to compile the large English lexicon [2]. Through manual annotation using Amazon's Mechanical Turk service, 14,182 words derived from the Macquarie Thesaurus were associated to any of the eight emotions [2]. While arduous, this approach proved to be effective in building the lexicon. The European University of Lefke compared a selection of lexicons, including the NRC, with respect to their size, common words, differences and effects on the success of classifying documents into emotions [52]. They found that the NRC and one other emotion lexicon provide better classification results compared to the others. Interestingly, the NRC lexicon is small in comparison to the others and yet it still proved to be more effective in associating emotions.

The NRC lexicon was applied to all tokenized words in each document (tweets and Yelp reviews). This yields, for each document d a count $EmoCount(d, e)$ of emotion words for any emotion e, thus allowing us to calculate the proportion of the emotions present in each document.

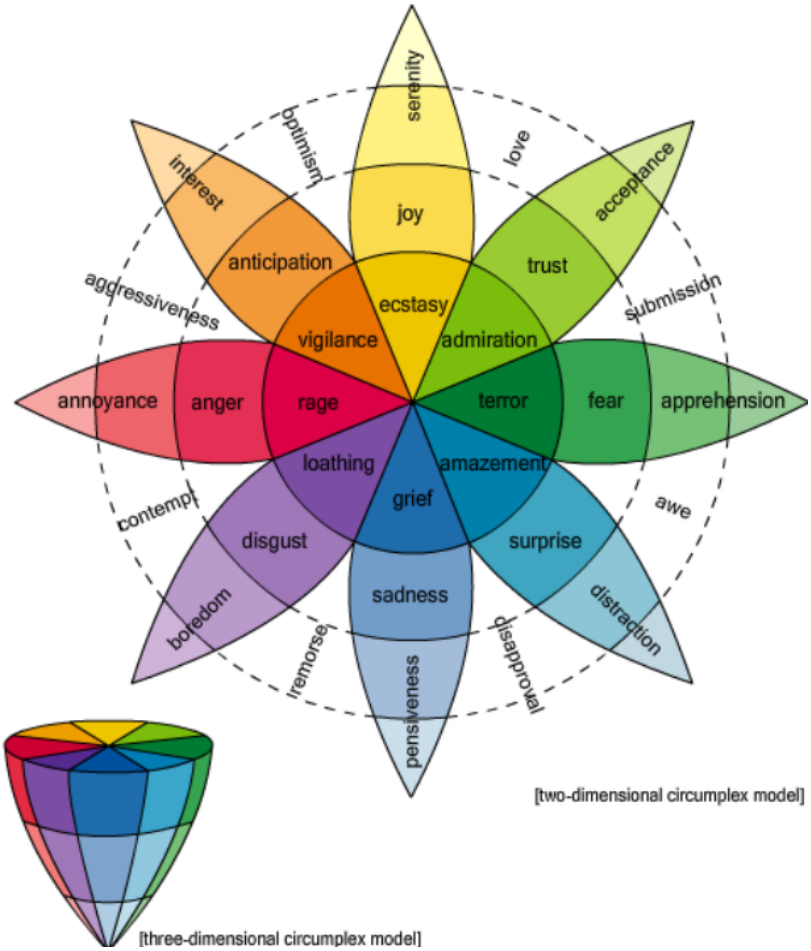


Figure 4.5: Pultchik's Wheel of Emotions [2]

Example 2. *For example, consider the following Tweet: "We visited #JoshuaTree in 2017 & it was incredible. It makes me sad to think of all the damage in the park during the #GovernmentShutdown. It's truly a magical place worth preserving & respecting!"*

The NRC lexicon identifies all underlined words as an emotion, yielding an $EmoCount(d, joy) = 2$, $EmoCount(d, anticipation) = 2$ *and* $EmoCount(d, sadness) = 1$ *for this document* d. *In this case, the word "sad" was classified as sadness, and all other words were classified as anticipation and joy.*

For a place, topic, and emotion, we simply define the intensity of emotion as the average

number of emotion-words at that place having that topic.

Definition 6 (Intensity of Emotion)**.** *Let t be a topic, let l be a location, and let $D_{t,l}$ be a set of documents of topic t at location l. For a given emotion e we define the Intensity of Emotion as*

$$IoE(D_{t,l}, e) = \frac{\sum_{d \in D_{t,l}} EmoCount(d, e)}{|D_{t,l}|} \tag{4.1}$$

The emotions of all documents were then grouped by city and topic resulting in two data structures: 1) one three column data frame per emotion containing latitude, longitude, and intensity of emotion which can be fed into the kriging algorithm and 2) a three mode location, topic, emotion tensor which we linearize into a matrix where each line corresponds to a (location, topic) pair to feed to SVD. This transformation is performed to directly apply SVD to the full tensor, rather than building a model separately for each emotion. This allows SVD to learn correlations between different emotions. We note that a tensor factorization approach, such as [53, 54] which directly factorizes the (topic, location, emotion) tensor into latent topic features, latent location features, and latent emotion features, may be a potential direction to improve the prediction results, but is out of the scope of this work.

Now that we have emotion data extracted from documents and arranged in data structures for kriging and SVD, the next Chapter describes our approach of combining kriging and SVD into a hybrid algorithm that exploits both the latent features modeled by SVD and the spatial autocorrelation estimated by kriging.

4.3.3 A Hybrid Approach to Combine Recommender Systems and Spatial Statistics

Once the two data structures are fed into their respective algorithms, the following variables are calculated: the error variance of the kriging predicted value - σ^2 in Equation 2.4, the kriging prediction - $\hat{z}(x_0)$ in Equation 2.1, and the SVD prediction - r_{ij} in Equation 2.13.

The error variance, σ^2, is of particular interest, as it indicates how confident the kriging estimation is about the accuracy of its results. These variables, σ^2, $\hat{z}(x_0)$ and r_{ij} are strategically combined to produce a stronger prediction of the emotional value of cities across the United States.

To combine these predictors into a single model to predict the intensity of an emotion, we employ two different approaches: One using a multilinear regression model, and one using a neural network. These two approaches are detailed in the following.

Multilinear Regression

Our first approach simply combines the kriging-based emotion prediction at a location (including the predicted value and the error varriance) with the SVD-based prediction using a linear model which we have named Hybrid Linear Regression (HLR).

Definition 7 (Hybrid Linear Model)**.** *Let t be a topic, let x_0 be a location and let e be an emotion. Further, let $\hat{z}(x_0)$ be the kriging-based emotion prediction (Equation 2.1) and let $\sigma^2(x_0)$ be the kriging error variance (Equation 2.4) both using using the Intensity of Emotion values of Definition 6. Also, let r_{ij} be the SVD prediction obtained from Equation 2.13 using the Intensity of Emotion matrix of Definition 6, where line i corresponds to the (location, topic)-pair (x_0, l), and column j corresponds to emotion e. We predict the intensity of e for topic t at location x_0 using the following linear model:*

$$\textit{Hybrid-LM}(x_0) = \beta_0 + \beta_1 \cdot \hat{z}(x_0) + \beta_2 \cdot \sigma^2(x_0) + \beta_3 \cdot r_{ij} + \beta_4 \cdot \hat{z}(x_0) \cdot \sigma^2(x_0) + \epsilon \qquad (4.2)$$

An interesting part of this regression model is the interaction term $\beta_4 \cdot \hat{z}(x_0) \cdot \sigma^2(x_0)$, which allows the model to learn to give less weight to the kriging estimating $\hat{z}(x_0)$ if the kriging variance $\sigma^2(x_0)$ is high, i.e., if kriging is not confident about it's estimation.

Neural Network Model

We additionally used a machine learning model that we implemented in a two layer neural network, with 300 neurons in the first layer and 100 neurons in the second layer as shown in Figure 4.6 which we have named Hybrid Neural Network (HNN). Since neural networks can be considered a black box where hyperparameters are fine tuned through grid search or evaluating every combination, we tested various combinations of layers and neurons for our model ranging from one to three layers, and 50 to 500 neurons. After running different networks, we found that a two layer network with 300 neurons in the first layer and 100 neurons in the second layer performed the best.

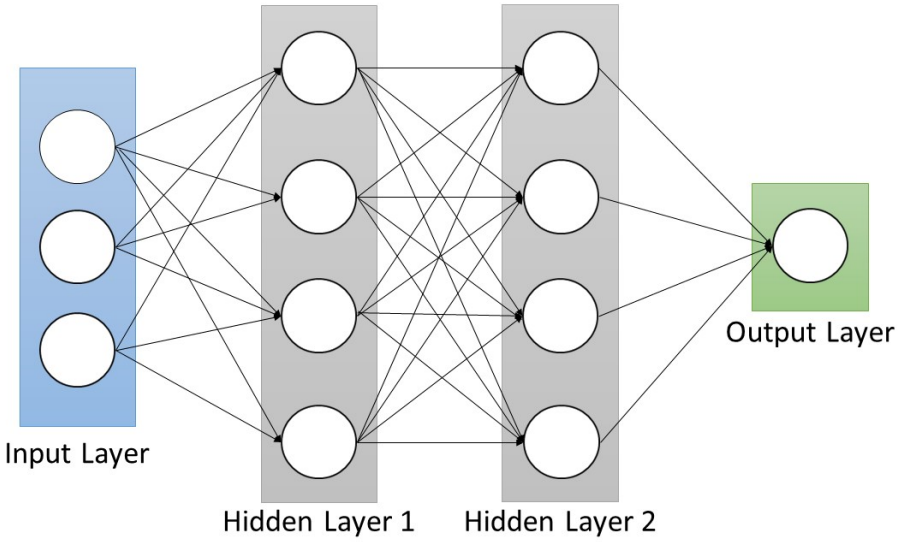


Figure 4.6: Neural Network

The input layer of the neural network takes the variables, $\sigma^2(x_0)$, $\hat{z}(x_0)$, and r_{ij}. The hidden layers will then calculate the weights of the variables and determine the value of the output variable. The neural network uses stochastic gradient descent using back-propagation to alter the weights between neurons to reduce the error of the model. Equation 4.3 is used to ingest the input variables into the neural network:

$$\hat{z} = \sum_{i=1}^{n} w_i x_i + b \tag{4.3}$$

where $\hat{z}$ is the output value for a specific layer, x is the input layer, w is the weight between the neurons, and b is the bias between the neurons. $\hat{z}$ is fed into the Sigmoid activation function which performs a transformation on the input received in order to keep values within a manageable range [55]. Therefore $h_1^1 = f(\hat{z})$ and is fed into the next layer, using the following equation:

$$\hat{z} = \sum_{i=1}^{n} w_i^{h_1} h_i^1 + b \tag{4.4}$$

where n is the number of hidden layer inputs and the output of the first layer, h_i^1 , is now the input layer. The final prediction value, $\hat{y}$, is calculated once $\hat{z}$ is fed through another Sigmoid activation function where $\hat{y} = f(z)$ [56]. More details on using neural networks for prediction can be found in [57].

In the following chapter, we empirically evaluate which approach yields the most promising results for topic-based emotion prediction: kriging, SVD, or one of the hybrid approaches.

4.4 Experimental Evaluations

We conducted part one of our experiments in a Windows environment with 4 CPU's at 2.7 GHz and 16 GB of RAM. The second part of our experiments was conducted in a virtual

Windows environment with 4 CPU's and 32 GB of RAM. The purpose of the experiments was to determine the optimal hyper-parameters for the algorithms mentioned in chapter 2 of this paper and to determine the Root Mean Squared Error (RMSE) of SVD, Kriging, and our hybrid approaches (HLR and HNN).

4.4.1 Kriging Hyper-parameters

In order to perform Kriging, we created a variogram sample which would then be used to fit a variogram model. A variogram quantifies spatial correlation by depicting the variances within groups of sampled points, plotted as a function of distance between them. The variogram sample is then fitted to the model of choice. There are several model options to choose from as a hyper-parameter which include Exponential, Spherical, Gaussian, and Matern. In order to determine the best model for our study, we split the data into train and test sets. We applied all four models to the training set and calculated the RMSE of each model. Since the Matern model had the lowest RMSE as shown in Figure 4.7, we selected it as the variogram model of choice.

From Equation 2.8, we can further tweak other hyper-parameters such as the possibility of measurement error known as the "nugget [29]. However, to reduce the complexity of the current discussion, we have chosen not to include a nugget effect in our process and analysis.

4.4.2 SVD Hyper-parameters

SVD has become a popular collaborative filtering option when it comes to recommender systems. Through matrix factorization we are able to discover the underlying hidden features that the interactions have between two entities. For recommender systems it is typically a user to item interaction, but for this study we are interested in the city to topic interaction.

The first hyper-parameter to optimize, and most obvious, is the number of hidden features, or the k value which affects the dimensions of the U and V matrix in Equation 2.12. If the algorithm is given fewer hidden features than what it actually has, then the RMSE increases. If the algorithm is given too many hidden features, the RMSE does not

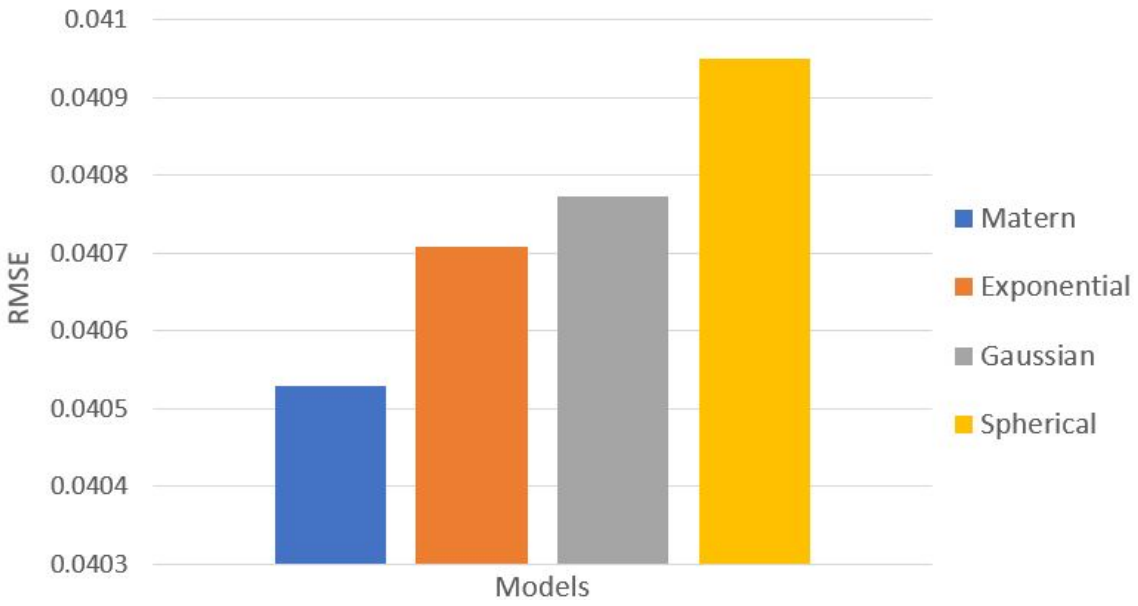


Figure 4.7: Variogram Models

improve and takes a longer time to execute without any added benefit. Figure 4.8 shows the RMSE ranging from k values 10 to 100. The RMSE score begins to level off at 40 hidden features and plateaus at 50 hidden features therefore we used $k = 50$ for this study.

We incorporated the SVD algorithm with stochastic gradient descent (SGD) in order to solve for the optimization problem in minimizing RMSE. SGD randomly initializes all vectors u_i and v_j, making predictions of what the decomposition should be and then calculates the error of the rating prediction using this initialization. The error and the derivative of the error is then used to improve the error and this process is continued for a number of iterations until it reaches a local minimum. The number of iterations is a hyper-parameter that was optimized for this study. Figure 4.9 shows that we tested for 30, 60, 100, 130, and 150 iterations. There was a 9.69% RMSE improvement from 30 to 60 iterations, a 6.46% improvement from 60 to 100 iterations, a 1.02% improvement from 100 to 130 iterations, and a 2.60% improvement from 130 to 150 iterations. Since the greatest RMSE improvements occurred within the 30 to 100 iteration range, we incorporated 100 iterations.

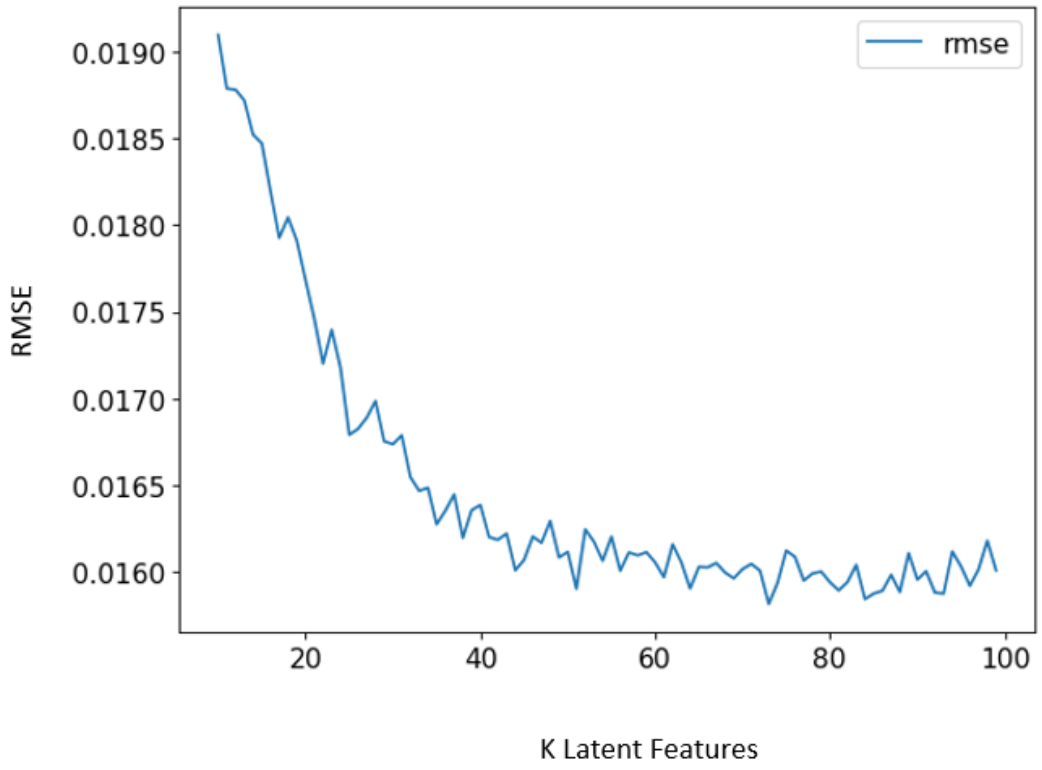


Figure 4.8: Latent Features vs RMSE

Other hyper-parameters for SVD with SGD include the learning rate and the regularization parameter. The learning rate parameter controls the rate at which the model learns by determining how fast or slow to move towards the optimal weights. The regularization parameter is used to avoid overfitting when predicting the values in the R matrix. As these hyper-parameters did not have significant effect on the prediction quality, we omit these experiments for brevity, using the default values of Python scikit-learn for truncated SVD, using a regularization term of 0.02 and a learning rate of 0.005.

4.4.3 Emotion Prediction Quality

Once the hyper-parameters were fine tuned for our case study, we began to test the RMSE of each algorithm: stand-alone SVD, standalone-kriging, the hybrid approach using linear

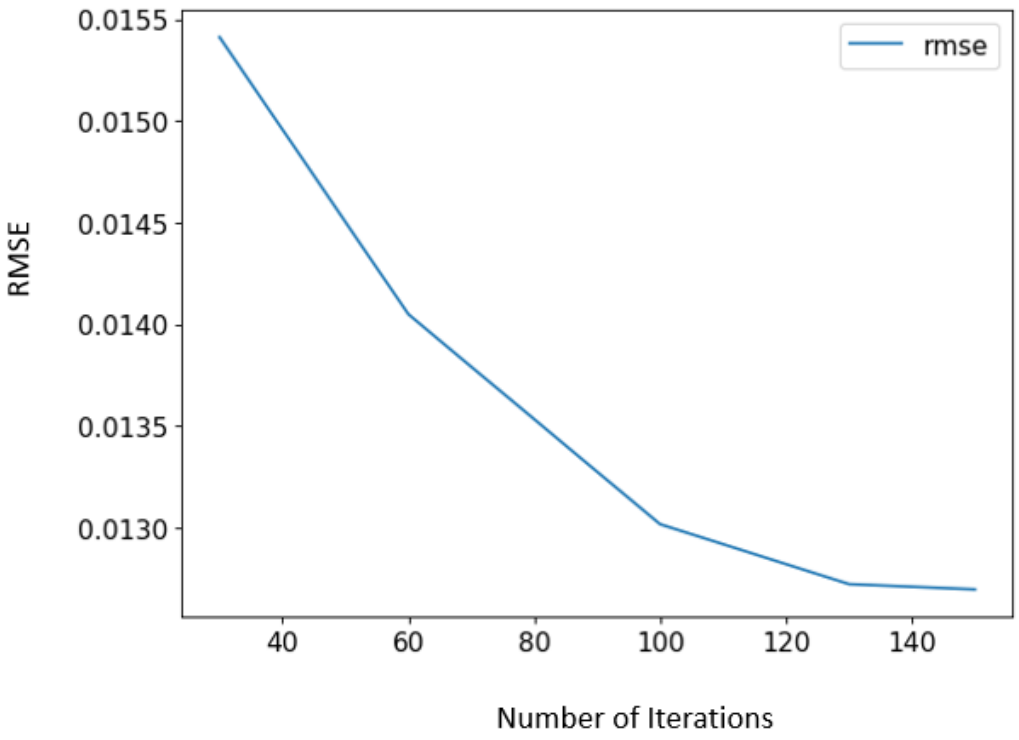


Figure 4.9: Iterations vs RMSE

regression (HLR), and the hybrid approach using a neural network (HNN). We tested the RMSE and execution time for each approach based on two data sets: 1) the combined data collected from Twitter and Yelp using explcit topics, and 2) the latent topic data set as described in Section 4.3.1. For the first data set, we used randomly sampled subsets of size 500,000 (Data Set A), 1.5 million (Data Set B), and the full 2.5 million (Data Set C) to evaluate efficiency. We also observed the RMSE and execution time for the latent feature data set (Data Set D), having 3.5 million documents, to verify that our approach was not over-fitting to the first data set.

Figure 4.10 shows the RMSE results for document sizes 500,000, 1.5 million, 2.5 million, and 3.5 million labeled A, B, C, and D, respectively. First, we observe that all algorithms improve as the dataset size increases in Datasets A, B, and C. This is intuitive, as a larger

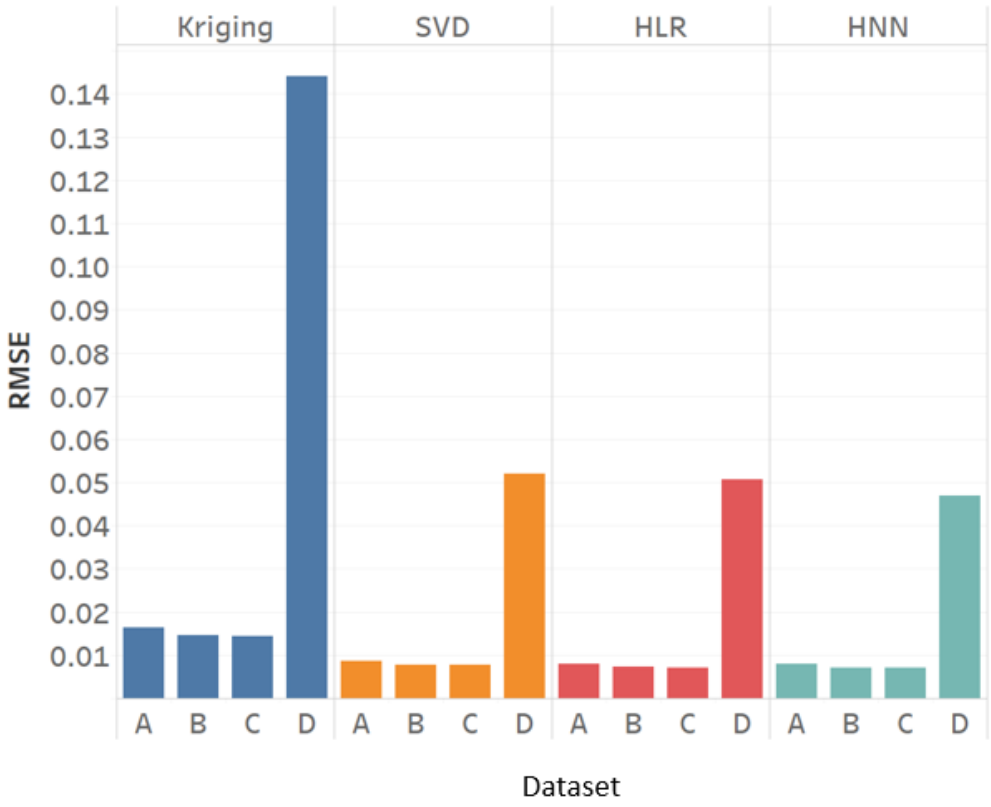


Figure 4.10: Algorithms vs. RMSE

data set to learn from allows all algorithms to exploit more information for more accuracte predictions. For Dataset D we see a huge spike in RMSE. This can be explained, as Dataset D is generated differently (using latent topics rather than explicit topics), and it appears that the supervising task of manually identifying topics does help the algorithm significantly. In contrast, Dataset D may have very noise topics, which are difficult to be exploited for accurate emotion predictions.

More importantly, Figure 4.10 shows our main result: Standalone kriging yields the largest prediction error. Standalone SVD improves this result significantly, thus proving to be a better model for predicting emotions. But, surprisingly, combining kriging and

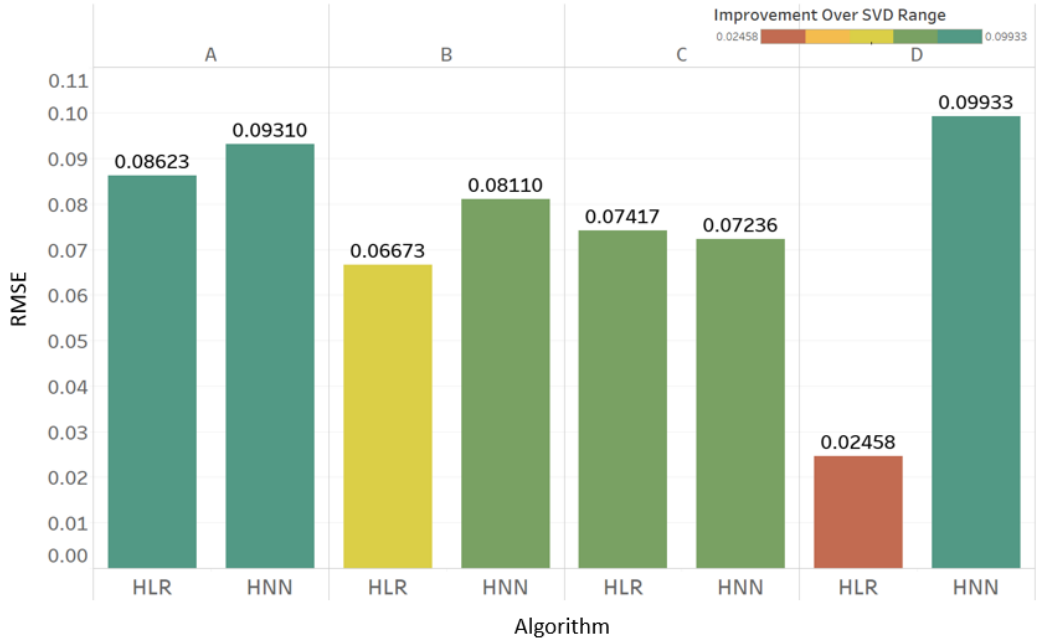


Figure 4.11: Relative RMSE improvement vs SVD

SVD outperforms standalone-SVD, showing the potential of combining the two orthogonal concepts. While the difference in RMSE between SVD, HLR and HNN is difficul to see, we show the relative difference in RMSE between them in Figure 4.11. Using SVD as a baseline, Figure 4.11 shows the relative improvement (in % of RMSE) of HLR and HNN compared to SVD for the four datasets. We see that, especially for the smaller Dataset A, the improvement exceeds 9.3% for HNN. This result indicates that in the case where we do not have sufficient data to train good latent features, spatial auto-correlation can still be used to replace the missing information. Even for the large Dataset C, the improvement of HNN still exceeds 7.2%. In the case of the large Dataset D using latent topics, we observe that HNN achieves a reduction of RMSE of nearly 10%, while HLR drops to an improvement of less than 3%, indicating that the neural network approach is more capable of making sense of the latent features for prediction.

To summarize these experiments, we see that SVD outperforms kriging, but our hybrid approaches significantly outperform SVD by up to nearly 10% of RMSE. This result is *not* insignificant, as an improvement of recommendation systems of 10% can be tremendous, shown by the example of the Netflix Challenge, where the $1M challenge was to decrease the RMSE by 10%. Further, we show that the neural network based HNN approach yields better results than the linear regression approach. Our result shows that the combination of recommender systems and spatial statistics is a powerful combination that is more potent than the sum of its parts, especially when applied to spatial data.

Finally, Figure 4.12 shows the execution time for all four data sets and all four algorithms. In all cases, the run-time increases in the size of the data, peaking at the largest Dataset D having 3.5 million tweets. We also see that SVD takes significantly longer than kriging. For our hybrid approaches HLR and HNN, we measure the time required to combine the kriging and SVD results. Thus, the corresponding times of kriging and SVD are also part of the run-times of HLR and HNN. We see that the neural network appraoch takes significantly longer than the linear regression approach. Thus, we see a trade-off between HLR and HNN in terms of efficiency and effectiveness.

4.5 Conclusions & Future Work

In this work, we propose an approach to predict the emotion of an area for a given topic derived from Twitter and Yelp to learn from. We approach this problem from two different angles. First, we tackle the problem from a spatial statistical perspective, using kriging to learn the spatial autocorrelation of different emotions of nearby places to interpolate the emotion at an unknown location. This allows the algorithm to utilize distance and spatial arrangement of nearby points. Second, we tackle the problem for a machine learning perspective, using a matrix factorization approach to learn latent emotion-features of topics and locations that the human eye cannot detect. This allows the algorithm to learn hidden patterns within the data that is not related to spatial information. These two different

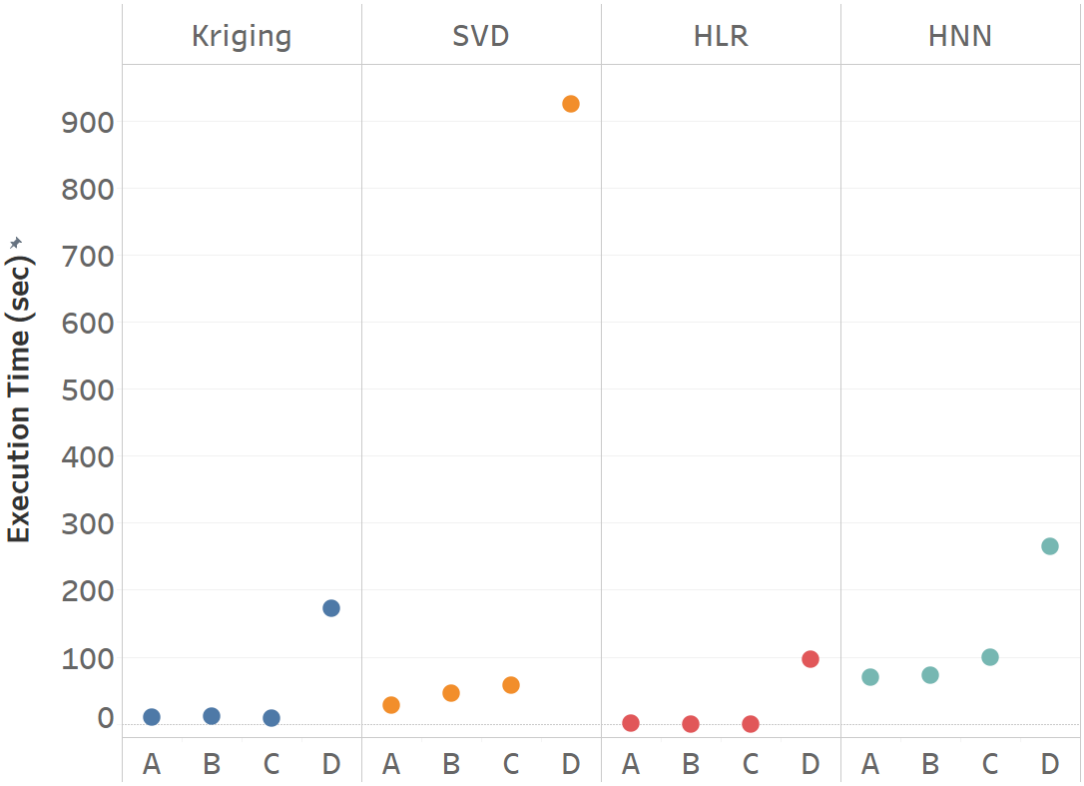


Figure 4.12: Algorithms vs. Execution Time

angles give the prediction algorithm more context to work with, thus resulting in better accuracy. While we show that the matrix factorization approach significantly outperforms the kriging approach, the main merit of our approach is the combination of both perspectives: We combine kriging and SVD into two models: 1) a single regression model using linear regression models (HLR) and 2) a neural network (HNN). Both models outperform the matrix factorization based approach, thus showing the promising result that geostastistical models can be used to leverage machine learning.

Chapter 5: Augmenting Spatial Statistics with Singular Value Decomposition:A Case Study for House Price Estimation

Abstract

Singular value decomposition (SVD) is ubiquitously used in recommendation systems to estimate and predict values based on latent features obtained through matrix factorization. But, oblivious of location information, SVD has limitations in predicting variables that have strong spatial autocorrelation, such as housing prices which strongly depend on spatial properties such as the neighborhood and school districts. In this work, we build an algorithm that integrates the latent feature learning capabilities of truncated SVD with kriging, which is called SVD-Regression Kriging (SVD-RK). In doing so, we address the problem of modeling and predicting spatially autocorrelated data for recommender engines using real estate housing prices by integrating spatial statistics. We also show that SVD-RK outperforms purely latent features based solutions as well as purely spatial approaches like Geographically Weighted Regression (GWR). Our proposed algorithm, SVD-RK, integrates the results of truncated SVD as an independent variable into a regression kriging approach. We show experimentally, that latent house price patterns learned using SVD are able to improve house price predictions of ordinary kriging in areas where house prices fluctuate locally. For areas where house prices are strongly spatially autocorrelated, evident by a house pricing variogram showing that the data can be mostly explained by spatial information only, we propose to feed the results of SVD into a geographically weighted regression model to outperform the orginary kriging approach.

5.1 Introduction

The price of a house or apartment is challenging to estimate. The price not only depends on various variables of the house itself, such as the number of rooms, but also on the location [58]. Different school districts [59], proximity to public transport [60], and safety of the neighborhood [61] affect the price of a house. Fluctuations in house prices can have a strong impact on real economic activity. For instance, the rapid rise and subsequent collapse in US residential housing prices is widely considered as one of the major causalities of the financial crisis of 2007-2009 [62], which has in turn led to a deep recession and a protracted decline in employment. In this work, our goal is to leverage powerful, but spatially oblivious, recommendation systems to improve geostatistics for house price estimation.

Recommender engines have become an integral part of our e-commerce. With the progression of the recommender systems, there have been many improvements to it's implementation. Recommender system engines allow to predict the price of a house by leveraging knowledge about similar houses and their prices. One enhancement has been in integrating spatial elements [63, 42] based on Tobler's first law of geography [35]. The idea of these existing works is to build individual recommender systems for each areal unit.

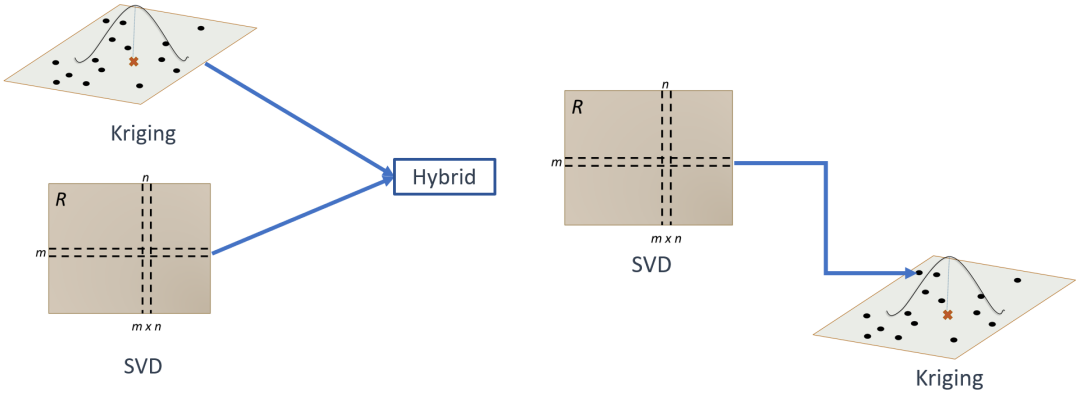
In this work, in addition to building a recommender system for each region, we feed the results of recommender systems, namely using matrix factorization, directly into a geostatistical interpolation, namely regression kriging. Combined, our approach leverages the strong predictive power of matrix factorization by building successive latent variables explaining most of the inter-correlation of variables. Used in a regression kriging, it allows to explicitly model locations and distances in order to predict well locally (to a certain horizon) through interpolation. We also show that our approach outperforms the purely latent features based solutions as well as purely spatial approaches such as ordinary kriging, universal kriging and geographically weighted regression (GWR).

Intuitively, kriging calculates a degree of spatial autocorrelation, to estimate an unknown point given values in the spatial vicinity. To augment kriging, we apply matrix factorization

using truncated singular-value decomposition (SVD), a state-of-the-art approach in recommendation systems. Truncated SVD takes an input matrix M of estimate housing values by location and house-type. Truncated SVD not only smooths out noise in the data, but also fills gaps in the data. This is particularly important, if, for example, we want to predict the price of a two bedroom/three bath house in an area, where we have not yet observed a house with these features. It estimates a house price for each location and for selected features by extracting latent factors corresponding to the set of k largest eigenvalues of M. By combining both approaches, we can augment the latent features (of both house features and locations) learned by truncated SVD, with spatial auto-correlation, thus obtaining the best of two worlds: latent feature learning and spatial statistical models.

There have been a number of studies conducted that utilize spatial statistics to predict the pricing of houses(e.g. [36, 37, 38]). These studies have researched kriging in detail within the real estate domain. The work of Cellmer [36] shows that the actual variations in prices and property values can considerably diverge from the values predicted by kriging, since data can be scarce and unevenly distributed and may depend on non-spatial variables. This work further notes that geostatistical methods may facilitate the interpretation and analysis of the causes of price variations rather than serve as a tool for predicting house prices. However, with the integration of truncated SVD, hidden patterns within the prices are revealed aiding the algorithm to make stronger and well-rounded predictions. To summarize, the main contribution of this work is not only to experimentally let kriging compete with truncated SVD, but rather, to have kriging complement truncated SVD. By combining predictions made by kriging and truncated SVD we are able to propose a hybrid approach which outperforms each individual approach. While our combination of truncated SVD and regression kriging is quite straight-forward, it yields promising prediction results enabling future research towards combining geostatistical models and recommender systems.

The remainder of this work is organized as follows. Section 5.2 surveys related work on spatially aware recommender systems and combining machine learning with variations of spatial statistics.Our approach to combine the best of both these worlds for housing price



(a) Combination of Individual Results [1]

(b) Feeding SVD Results into Kriging

Figure 5.1: Combination of SVD and Kriging

predictions is presented in Section 5.4. Our experimental evaluation to quantitatively evaluate our proposed SVD-Regression Kriging (SVD-RK) and compare it to baseline solutions is presented in Section 5.5. Finally, we conclude in Section 5.6.

5.2 Related Work

The goal of this work is to incorporate spatial information into recommender engines and improve upon what researchers in this field have already studied in regression kriging (RK). First, Spatially-Aware recommender systems are related as surveyed in Chapter 4.2.2. In this section we further look at how RK has been used in its dominant field, environmental science, and how it is being used as part of machine learning algorithms.

5.2.1 Machine Learning with Spatial Statistic

Ordinary kriging is a widely used spatial interpolation method and is based on a stochastic model of continuous spatial variation which can be depicted though a variogram or covariance function [64, 6]. It is by far the most popular form of kriging which was been

combined with SVD using traditional classification algorithms in [1] as illustrated in Figure 5.1a. However, rather than combining the results independently returned by kriging and SVD, the goal of this work is to integrate SVD directly in the kriging process, thus allowing kriging to leverage learned patterns and estimated house prices provided by SVD as sketched in Figure 5.1b.

The technique of using auxiliary attributes of data as independent variables in universal kriging is also referred to as regression kriging (RK). This method has been used extensively in the environmental sciences. Hengl et al. [65] describes different case studies where RK is traditionally applied in the field of environmental science. The study explores mapping soil organic matter using RK, mapping presence/absence of the yew plant, and mapping land surface temperatures. More recently, RK has also been extended to the machine learning field of study.Li et al. [64] used residuals from machine learning outputs into ordinary kriging (OK). For their combined methods, they applied LM, GLM, GLS, Rpart, RF, SVM and KSVM to the data, then OK or Inverse Distance Squared (IDS) was applied to the residuals of these models. They predicted values of each model and the corresponding interpolated residual values were added together to produce the final predictions of each combined method. The combination of RF and SVM with OK and IDS were novel at the time and had not been applied in previous studies in environmental sciences. Tadic et al. [66] used a similar approach to [64] and applied the output of their neural network as covariates in the universal kriging (UK) algorithm and also applied the neural network residuals as covariates to the ordinary kriging algorithm to interpolate atmospheric temperatures . However, the study found that their approaches were only successful 50% of the time. Another study used the residuals from the machine learning algorithms Random Forest and Boosted Regression Trees to predict topsoil organic carbon [67]. However, the researchers did not find any significant improvements using the residuals as covariates to OK.

Our approach spatially improves recommender engines using similar approaches in [66] and [67] but stabilizing and improving the results. For our algorithm, truncated SVD is segmented by county and calculated individually. Then, both the outputs and residuals

from the collaborative filtering method for each county are used as independent variables in UK. The observed values from truncated SVD are weighted based on the spatial structures calculated from UK, resulting in statistically significant improvements to truncated SVD.

5.3 Zillow Study Area and Dataset

The data used for this study was derived from the Zillow Price Competition on Kaggle in 2016 and can still be found on the Kaggle website[1]. The competition was to develop an algorithm that makes predictions about the future sale prices of houses and in turn help increase the accuracy of the current Zillow algorithm at the time. "Zestimates" are estimated house values that were based on 7.5 million statistical and machine learning models that analyze hundreds of data points on each property. While the competition has now been closed, we utilized this data set on our algorithm because the data points are densely populated, as shown in Figure 5.2, which makes it an excellent candidate for testing our algorithm that requires at least a moderate level of spatial autocorrelation.

The data set contains points from three different counties in California: Ventura, Los Angeles, and Orange county with County ID's 2061, 3101, and 1286 respectively. Each county will be evaluated to examine what methods are best to predict housing prices based on the characteristics of each county. The data contains dozens of attributes of each house but for this study we selected the attributes that had the least "NA" values which include, latitude, longitude, bedroom count, and bathroom count. Figure 5.3 provides a boxplot to show the range of housing prices in each of the three counties. The boxes show the 25% to 75% quantiles of housing prices for each county, whereas their whiskers (the lines reaching out from the boxes) show the 2.5% to 97.5% quantiles. We observe that County 3101 has the largest range of house prices, including outliers costing less than USD 1,000, and houses priced at as much as ten million USD.

[1]Kaggle Data Set available at: https://www.kaggle.com/c/zillow-prize-1/data

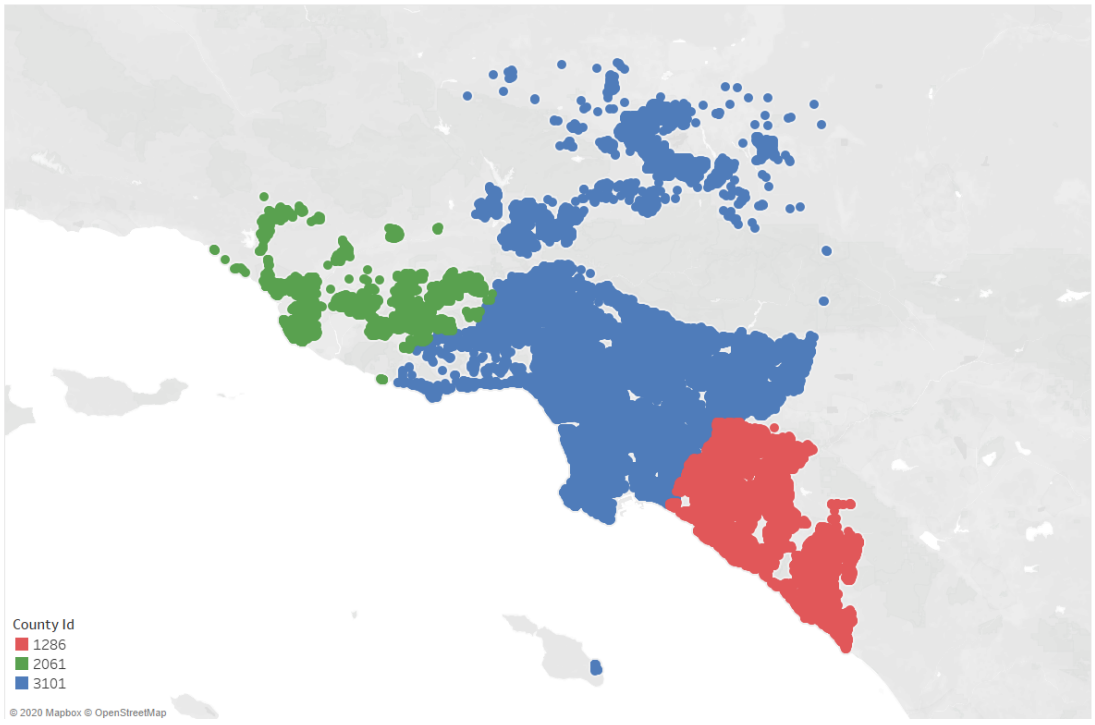


Figure 5.2: Zillow Price Competition Data

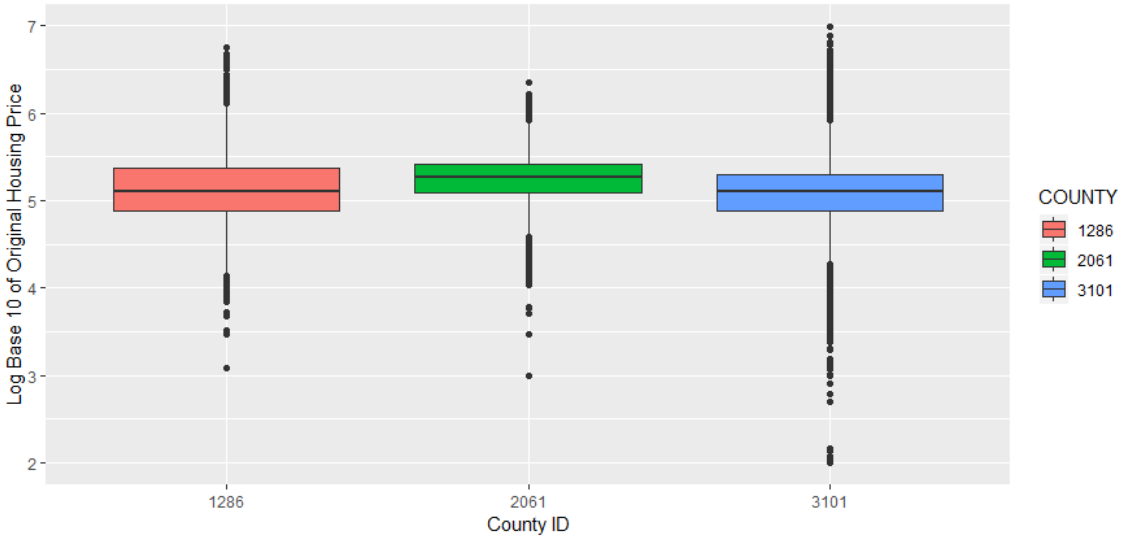


Figure 5.3: Boxplot of Housing Prices

5.4 Methodology

In this study, with the Zillow data discussed in Section 5.3, we first show that ordinary kriging can provide better house price prediction better than machine learning approaches based on SVD. Second we show how kriging combined with SVD can make strong predictions better than kriging alone.

Simply stated, the idea of our approach is to first assess how well the deviation of prices can be explained by distance. Cases where the covariance function shows a strong fit to the empirical variogram indicate that ordinary kriging may be sufficient to yield good predictions, without the need to further explain the deviation using SVD. However, in cases where some places cannot be explained by their local neighborhoods, such as an expensive area adjacent to cheaper housing, or areas in close proximity having varying house prices due to different school districts, we may be able to leverage SVD. The idea is that SVD may find that a location is similar to other areas (regardless of distance), and thus, estimate house prices similar to these locations having similar (latent) features. The following subsection describes our approach, as shown in Figure 5.4, to leverage universal kriging with information derived from latent feature analysis using truncated SVD.

5.4.1 Step 1 : Segmenting

Our proposed algorithm integrates hidden patterns that the human eye cannot detect using distance and spatial arrangement of nearby observations to improve the prediction of unobserved values. In a previous study [1], we combined the outputs of SVD and OK into a hybrid model using a multi-linear regression model and a neural network as shown in Figure 5.1a. The shortcoming of this approach is the assumption of a stationary data set, where mean and standard deviation of the variable of interest are identical in all locations. This assumption held in the previous study [1], of predicting emotion values rather than house prices. However, different neighborhoods can have significantly different mean house prices, incurring a large error if stationary means and homoscedasticity are assumed, and leading to major confusion by the neural network approach as shown in Table 5.1.

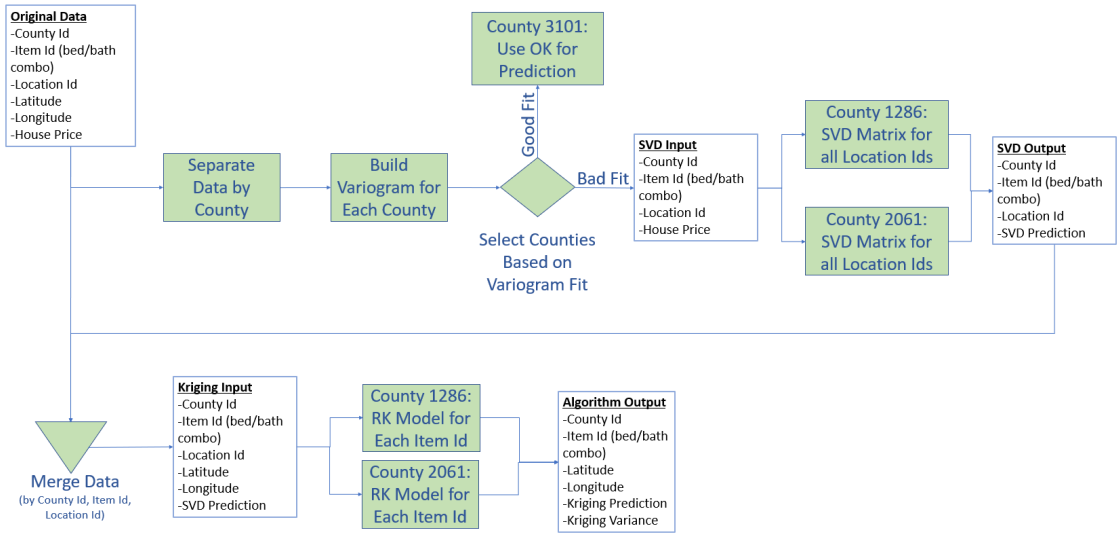


Figure 5.4: Methodology Flowchart

Table 5.1 also shows us that when the whole data set is ingested into SVD, UK, and OK, then SVD outperforms both types of kriging significantly. However, as shown in the second row of Table 5.1, when the data is segmented by county, OK performs better than when the whole data set is utilized at once. This is because, when segmented, each county has its own covariance function that describes the unique spatial auto-correlation. We leverage this segmentation for both SVD matrices and kriging models, and build an algorithm that can outperform both SVD and kriging. Thus, as shown as in Table 5.4, the first step from the Original Data is to separate the data by county.

5.4.2 Step 2: Variogram Construction

In the second step of Figure 5.4, we take the separated data and construct house pricing variograms for each study area to understand their spatial structure. The purpose of the variogram for a study area is to decide whether kriging alone is expected to yield good predictions, or if we should leverage SVD to explain the unexplained residuals which is the decision point in Figure 5.4. Our experimental evaluation in Section 5.5 will depict

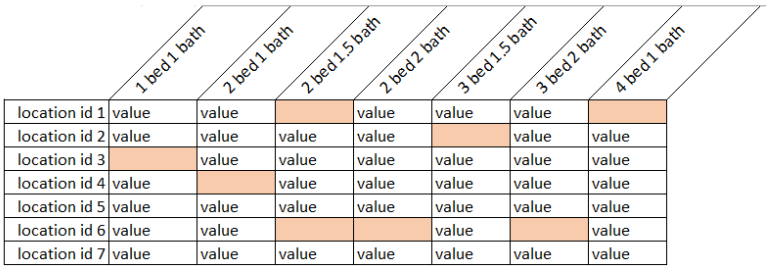
Table 5.1: Mean Absolute Error (MAE) in USD for baseline approaches, including Ordinary Kriging (OK), Universal Kriging (UK), Singular Value Decomposition (SVD), and the Hybrid Algorithm presented in [1]

County Id	OK	UK	SVD	Hybrid
All Counties	808,000	791,000	130000	> 1,000,000
Average of Segmented Counties	74,400	103,400	99,800	> 1,000,000

example cases in which the model variogram strongly fits the empirical variogram such that SVD cannot improve the housing prediction (county id 3101), and show cases in which the model variogram does not fit the empirical variogram well such that the strength of SVD can improve the housing price prediction (county id 1286 and 2061).

5.4.3 Step 3: Applying SVD to House Price Matrix

In order to utilize truncated SVD as part of our algorithm, we used location ids that were derived from unique latitude and longitude combinations as the rows for the matrix, and the non-spatial attributes (bedroom and bathroom combinations) for the columns of the matrix as shown in Figure 5.5. This data structure allows the truncated SVD algorithm to make housing price predictions of different locations for all the possible bedroom-bathroom possibilities that were unavailable, depicted in orange cells in Figure 5.5. Truncated SVD allows our algorithm to leverage the benefits of traditional recommender systems by identifying similar locations (i.e.,, locations having similar prices for the similar types of houses), and by identifying similar types of houses (i.e., types of houses having similar prices for similar locations). Truncated SVD only takes into consideration hidden patterns within the data and does not look at spatial elements in any way. Our approach is designed to use the SVD outputs and residuals and weigh the truncated SVD predictions based on their spatial structure through RK, which we call SVD-RK.



	1 bed 1 bath	2 bed 1 bath	2 bed 1.5 bath	2 bed 2 bath	3 bed 1.5 bath	3 bed 2 bath	4 bed 1 bath
location id 1	value	value		value	value	value	
location id 2	value	value	value	value		value	value
location id 3		value	value	value	value	value	value
location id 4	value		value	value	value	value	value
location id 5	value	value	value	value	value	value	value
location id 6	value	value			value		value
location id 7	value	value	value	value	value	value	value

Figure 5.5: Housing Price Matrix Representation

5.4.4 Step 4: Feeding Truncated SVD into Regression Kriging

As described in Figure 5.4, kriging and SVD are combined by feeding the SVD output into a regression kriging as covariates as sketched in Figure 5.1b and architected in Figure 5.4. The reason for feeding SVD outputs into kriging instead of kriging outputs into SVD is because in real estate housing price estimation, the SVD outputs are stronger than UK when compared individually as shown in Table 5.1. Thus, feeding more accurate estimations from SVD into kriging, our algorithm will further improve the predictions by taking into account spatial autocorrelation.

The idea of our approach is that truncated SVD leverage latent features of locations for prediction. For example, SVD may find that one location exhibits similar housing prices than a set of other locations, thus leveraging these other locations for prediction. Truncated SVD treats each location as a nominal variable, and thus, is oblivious of distances between locations and unable to leverage spatial autocorrelation. By leveraging the variogram constructed as described in Section 5.4.2 (and shown in Section 5.5 for the Zillow dataset described in Section 5.3), we can explore if a given dataset is strongly spatially autocorrelated. In that case, we use the unobserved values estimated by trunacated SVD as independent variables for universal kriging (UK). Traditionally, the mean of UK is a function of the site coordinates, such that the latitude and longitude are independent variables

of a linear model [6]. However, UK can also use a regression from other independent variables which is known as regression kriging (RK). RK is a spatial interpolation technique that combines a regression of the dependent variable on auxiliary variables (such as soil nutrients, or temperature) with simple kriging of the regression residuals. It is mathematically equivalent to the UK, where auxiliary predictors are used directly to solve the kriging weights [65]. Instead of using latitude and longitude as traditional auxiliary variables, we use other attributes, the SVD outputs and residuals, as independent variables which lead to better results. This allows UK to use new auxiliary variables to calculate the weight matrix for all observed points.

Variations of this method have been used in the traditional field of environmental science for interpolating points such as radioactive soil contamination, CO2 anisotropic atmospheric environments, and soil organic carbon, but not yet on the truncated SVD recommender engine [66, 67, 64]. The study [66] explored regression kriging using the outputs of an artificial neural network as covariates to UK. The approach showed some promising properties, in that it did better than UK alone in four out of eight examined model cases, however, the evidence of benefits was inconclusive since it showed varying degrees of success. In the study [67], Random Forest and Boosted Regression Tree residuals were used as covariates with ordinary kriging but RT residuals with OK did not show any improvements as covariants. The study found that OK and UK were the most accurate mapping methods in comparison to RK. We experimented using the truncated SVD outputs and the truncated SVD residuals as covariates separately for OK and UK. However, we found that that these methods, as explained in [66] and [67] showed no benefit in predicting points. Thus we applied a segmentation enhancement to the approach that has shown strong improvements to the final predictions.

The strength of kriging is that a variance can be calculated to determine the certainty of the kriging interpolation. SVD-OK feeds SVD into kriging which not only outputs a stronger prediction than either SVD or kriging individually, but it also outputs a variance. This can be utilized to make intelligible predictions because it quantigies the confidence

of the prediction that the algorithm makes. Thus the variance causes a more explainable approach to machine learning. Another important aspect to UK is that the technique is able to calculate the weight vector for unobserved points using independent variables that may only be present in training the model but not present or available in the unobserved points. With this in mind, we have utilized both the truncated SVD output *and* the truncated SVD residual as independent variables together in creating our covariance function. Thus, the kriging approach not only knows the SVD prediction based on latent factors, but it also knows the confidence of these prediction given its difference to the ground truth of training data. Our results show that our combined approach SVD-RK, which feeds the truncated SVD results in regression kriging outperforms individual approaches based on matrix factorization and kriging alone as well as the approaches in previous studies [66, 67].

5.4.5 Coding Language and Packages

This study was conducted using both the R and Python programming languages. The python package "surprise" was used to apply truncated SVD to the data [68]. Specifically the SVD function, under the matrix factorization algorithms, within the "surprise" package was used to to perform truncated SVD. More information on this technique can be found in [69] and [70]. Table 5.2 describes the hyperparameters used for this study. The latent factor was selected by running experiments that evaluated the error for each k latent feature, which will be described in the later sections (as shown in Figure 5.7). The R package “gstat” was used in applying kriging to the data [71]. The “variogram”, “fit.variogram”, and “krige” functions were utilized in making the predictions using the truncated SVD outputs and residuals. All code for this study can be found on on GitHub [72].

5.5 Experimental Evaluations

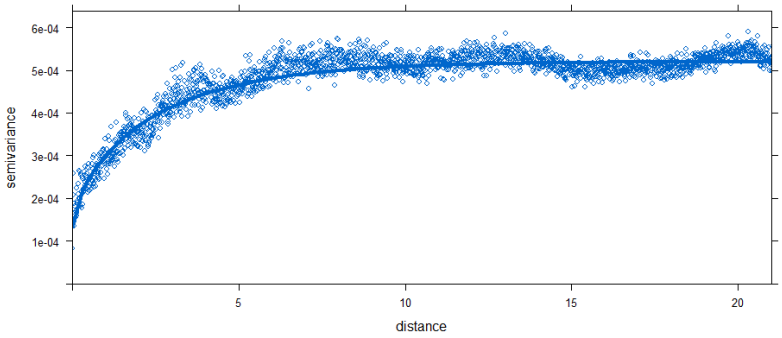
Our aim is to explore and combine the techniques of kriging and Recommender Engines using truncated SVD such that the results will stabilize and reduce the mean absolute error (MAE) of housing price predictions. In order to leverage the strengths of both methods, we

Table 5.2: SVD Hyperparamter Values

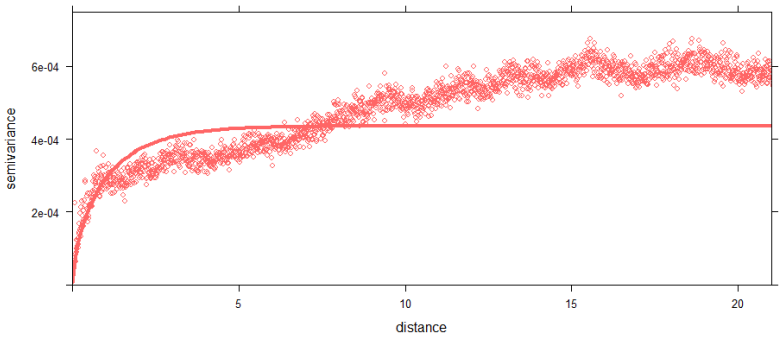
Parameter	Value
Latent Factors	5
Epochs	70
Learning Rate for b_u	5e-8

first built variograms for each county. Each county describes a different spatial structure as shown in Figure 5.6. As shown with subfigure 5.6a, the covariance function fits the empirical variogram very well for county id 3101 which gives us insight into what algorithm to use for this county. Since the covariance function can describe the spatial structure well for county id 3101, OK will be the most suitable approach to predicting housing prices. As shown with subfigure 5.6b and 5.6c, the best fit curve function still cannot fully describe the spatial structure for county id 1286 and 2061. However, with our proposed algorithm, we found that SVD-RK can utilize hidden patterns within the data and kriging can weigh the SVD outputs and residuals based on spatial correlation to produce stronger results than OK alone.

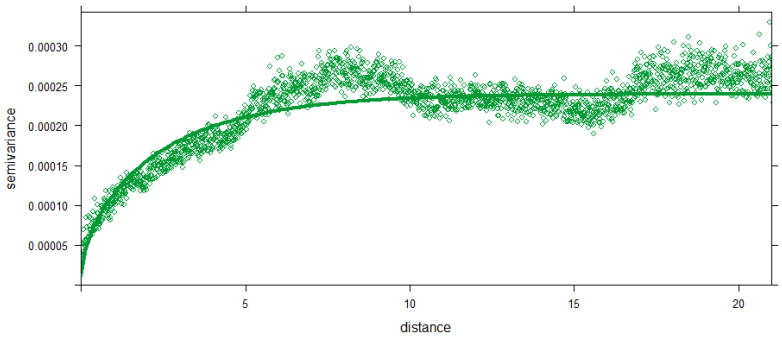
For the experiments conducted, we split the data into training and testing sets. Since the data was derived from the Zillow Kaggle Challenge as explained in Section 5.3, the challenge specified which data points were to be used as part of the training and testing. This study followed the exact same train/test split procedure for the purpose of producing comparable results. Following the exact same procedure allows all the experiments in this study to be evaluated fairly. In addition, there were no issues with minimum point density as a limitation for kriging. Since the data has already been pre-processed for the Kaggle challenge, there was a sufficient amount of points to build a variogram for every county. Furthermore, for data points that had NA values for some attributes, these lines of data were dropped. Only values within the data set that had a value for latitude, longitude, bedroom count, bathroom count, and price were utilized in this study.



(a) Variogram for County ID 3101



(b) Variogram for County ID 1286



(c) Variogram for County ID 2061

Figure 5.6: Variograms by County (Distance in Miles)

5.5.1 Evaluation of SVD-RK

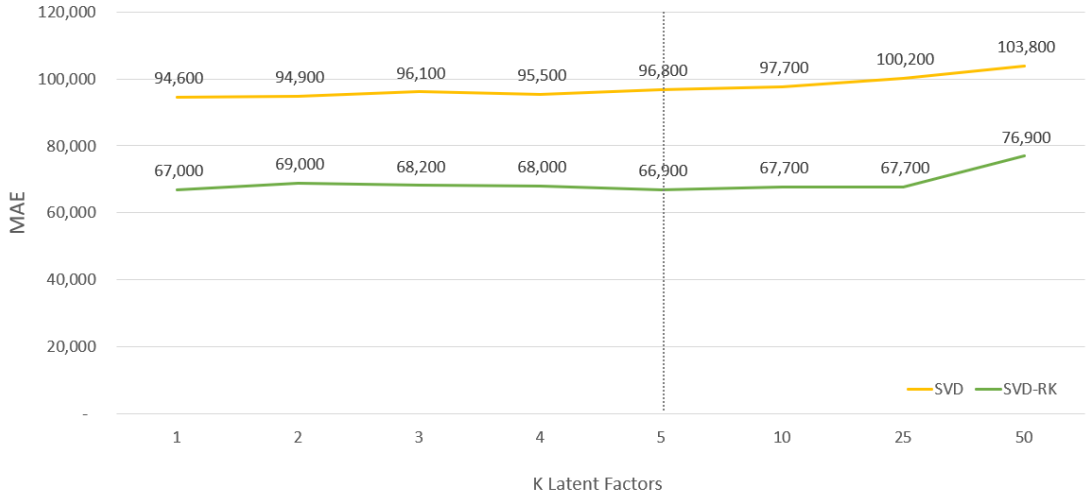


Figure 5.7: Latent Factor Experiment

Integrating truncated SVD into regression kriging, we first created separate matrices for each county to find the latent patterns. An important SVD parameter is the choice of K, the number of latent features, shown in Table 5.2. From running experiments shown in Figure 5.7, we found that any number less than 10 features yields similarly good prediction results. In particular, we get similar (only slightly worse results) using k=1 latent features instead of k=5. This shows that a single feature (which may latently describe the "priceliness" of a neighborhood) yields good price prediction results.

Table 5.2 also shows the appropriate amount of epochs, which is the number of iterations of the SGD procedure. The learning rate in SVD is the average rating given by a user minus the average ratings of all items. This term is used to reduce the error between the predicted and actual value. For this study, the learning rate set to 5e-8 was found to produce optimal results.

The SVD results are then fed into separate kriging models by county to weigh the observed values based on the spatial description of that county. Since every county may have different characteristics, segmenting the models helps to weigh the observations appropriately.

Table 5.3: Segmenting SVD Matrices and Kriging

County	OK MAE	UK MAE	SVD MAE	NMF-RK	SVD-RK MAE	Improvement
3101	83,500	158,000	128,300	> 1,000,000	85,100	-1.9%
1286	80,600	83,700	96,800	69,600	66,900	17.0%
2061	59,100	68,500	74,400	56,000	55,800	5.6%

As shown in Table 5.3, our method, SVD-RK, which applies truncated SVD matrices and kriging models by region presents improvements to counties 1286 and 2061, which could not be fully described by the covariance function in kriging alone as shown in Figure 5.6. We found that when appling SVD-RK to each county separately, truncated SVD matrices are able to better learn the hidden features of the data for a particular area thus producing results with less outliers which leads to better kriging results. For truncated SVD, every county may have different hidden characteristics which can be extracted more clearly when segmented. For UK, segmenting each model by county, in addition to segmenting matrices in truncated SVD, allows the algorithm to better describe the spatial structure of each county better than apply one generic covariance function to describe all three counties. Thus, applying SVD-RK by county, allows the algorithm to tailor the spatial structure differently for a given area. One area may be best described with a Spherical covariance function while another may be better described with a Matern covariance function. Table 5.4 lists the spatial structure for each county to highlight their differences and why segmenting would stabilize the results and make better predictions for certain counties.

For comparison purposes we also tested how Non-Negative Matrix Factorization (NMF), an implementation for SVD that follows [70], would react if fed into RK. NMF is similar to SVD, except fo having an additional constraint of imposing non-negative weights in matrices U and V. Table 5.3 shows the results of NMF-RK and show that SVD-RK yields better results. Thus this study focuses on SVD-RK as the method of choice.

Our approach, as shown in Table 5.3, does not work well for county 3101, which covers regions of different densities with large distances between these regions (see Figure 5.2), which also has outliers with extremely low/high house prices, as shown in Figure 5.3. In this case, the large variance of house prices combined with the large distance to nearest points in some locations yields to vast mis-estimations. We also see that ordinary kriging yields extremely good results for this county, due to the variogram fitting the empirical covariances very closely as shown in Figure 5.6a. Thus, in this case, the spatial location (captured by orginary kriging) is already able to tightly fit the data and explain most of the error. Yet, we will propose two approaches to improve our approach to work in cases such as County 3101. First, we will propose an approach for outlier remove in Section 5.5.2. While this appraoch stabilizes our SVD-RK approach, it still yields worse results than ordinary kriging. Thus, we propose to substitute regression kriging with GWR, in our approach, leading to better results as will be shown in Section 5.5.3.

Table 5.4: Spatial Structure by County

County ID	Curve Function
3101	Spherical
1286	Matern
2061	Spherical

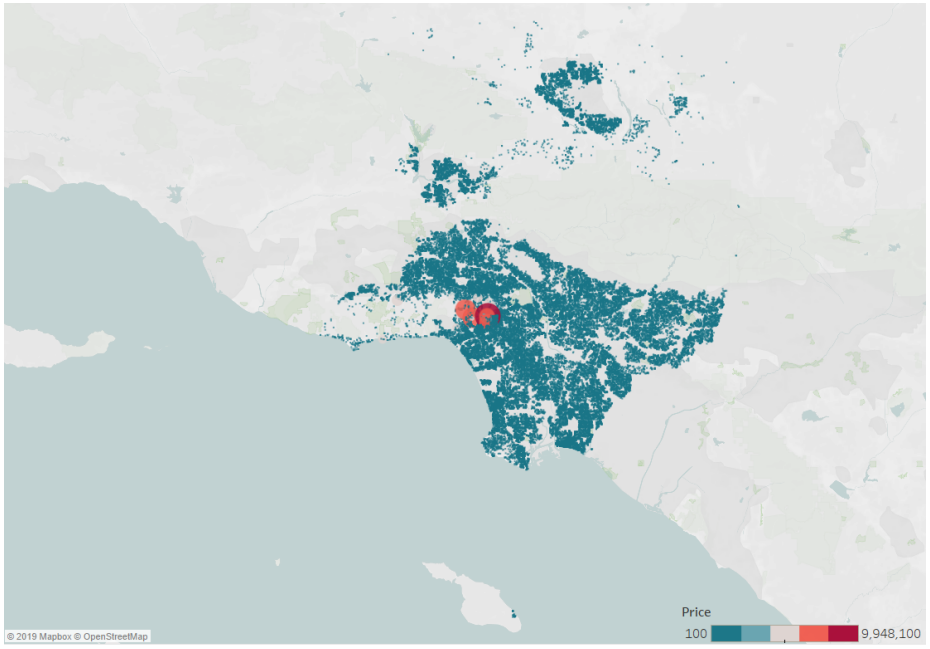
5.5.2 Outlier Removal for SVD-RK

While SVD-RK has shown significant improvements to the truncated SVD predictions, county id 3101 had no improvement. To better understand why the improvement for county id 3101 was not as strong as the others, we created a boxplot to visually see the outliers in Figure 5.3 and mapped the housing prices shown in Figure 5.8. Subfigure 5.8a shows county id 3101 with a relatively even distribution of housing price values except near the bend where there are red housing prices showing more extreme values, while subfigure 5.8b for county id 2061 and 1286 shows almost no extreme values. We then calculated the minimum, maximum, average, and standard deviation of the housing prices for each county id shown in Table 5.5. From Table 5.5, it is evident that county id 3101 has the largest difference between the minimum and maximum house price points and has visible outliers. Since kriging is highly sensitive to skewed distributions and extreme positive values having a large impact on semi-variance calculations, this caused the improvement in error to be relatively smaller for county id 3101 [6].

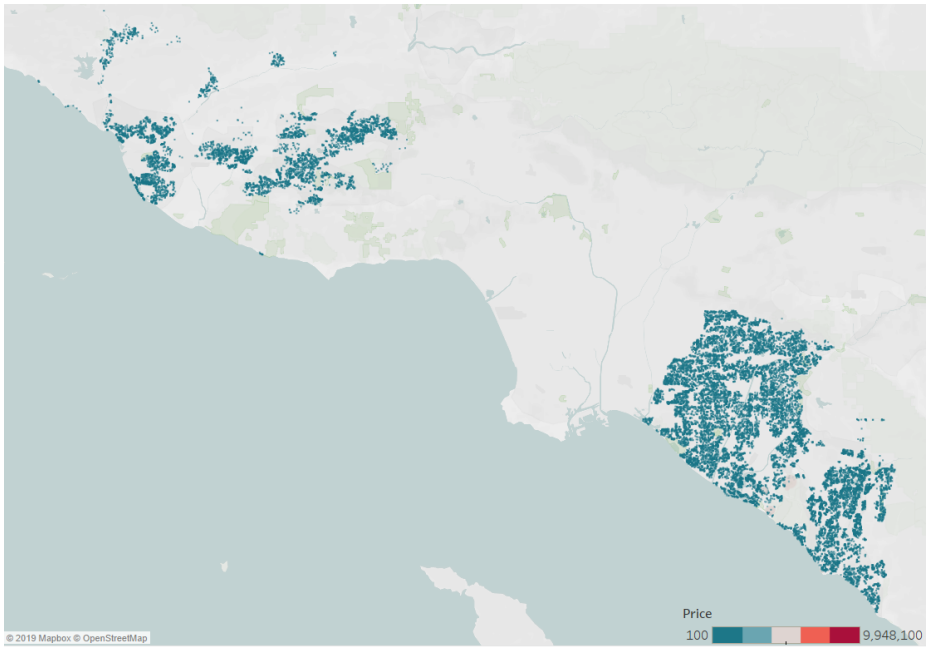
Table 5.5: Price Distribution by County in One Thousand USD

County ID	Max	Min	Mean	Standard Deviation
3101	9948.1	0.1	174.3	218.9
1286	5680.5	1.2	198.3	237.7
2061	2265.0	1.0	215.1	151.6

To improve the mean absolute error for county id 3101, we removed the outliers when kriging to decrease standard deviation and range between the maximum and minimum values. We removed all outliers deviating by more than two standard deviations from the mean. This resulted in a smaller standard deviation and an increase in accuracy by 25.9% from the original segmentation OK method (in Table 5.3). For those outliers that were left



(a) Distribution for County ID 3101



(b) Distribution for County ID 2061 and 1286

Figure 5.8: Data Distribution

from being ingested into the algorithm, the results from using OK without treating outliers can still be used with an MAE of 83,000.

Table 5.6: Removing Outliers for County Id 3101

County ID	OK MAE	UK MAE	SVD MAE	NMF-RK	SVD-RK MAE
3101	61,900	126,900	104,700	66,800	63,300

5.5.3 Feeding Truncated SVD into Geographically Weighted Regression (SVD-GWR)

In addition to non-stationary kriging, this study also explored improving truncated SVD with GWR since it is one of the recent developments of spatial analytical methods [73]. GWR is a local spatial statistical technique for exploring non-stationary data and, although controversial, it can also be used to make predictions. Similar to kriging, this algorithm is closely related to Tobler's first law of geography assuming that closer observations are more related than distant observations [35]. However, kriging uses a variogram which determines the variation of the data's spatial structure to determine the spatial autocorrelaton while GWR uses a kernel which determines the geographically weighting schema to determine the spatial autocorrelation. GWR calibrates a separate ordinary least squares regression at each location in the data set within a certain bandwidth and allows the relationships between the independent and dependent variables to vary locally. The GWR model can be described as the following:

$$y_i = \beta_{i0} + \sum_{k=1}^{p} \beta_{ik} x_{ik} + \epsilon_i \tag{5.1}$$

where y_i is the dependent variable at location i, β_{i0} is the intercept coefficient at location i, x_{ik} is the k-th explanatory variable at location i, β_{ik} is the k-th local regression coefficient for the kth explanatory variable at location i, ϵ_i is the random error term associated with location i, and i is indexed by two-dimensional geographic coordinates (u_i, v_i) [74]. The equation for GWR is similar to traditional regression however, for GWR, the regression coefficients are estimated at each data location at a local level rather than being at a global level [73]. GWR uses a geographical weighting scheme, known as a Kernel, which dictates the nearest neighbors that will be used to calculate local coefficients [75]. Further details for the GWR method can be found in [73], [74], [76], and [77].

While some studies have shown the potential of GWR as a predictor [78] there is still some controversy whether to use this method to make predictions [21]. Thus we compared GWR to OK for county id 3101 and our algorithm SVD-RK for county id's 1286 and 2061 to determine if this method would increase the prediction accuracy better. For this experiment we used the truncated SVD output and residual as independent variables for Equation 5.1. Table 5.7 shows the results of applying integrating truncated SVD to GWR, which we called SVD-GWR, compared to the OK algorithm (since this method performed the best for county id 3101 as seen in Table 5.3). We found that SVD-GWR improves the accuracy by 6.2%. However, Table 5.8 shows that SVD-GWR performed worse for county id 1286 and 2061 in comparison to SVD-RK (the optimal method for the two counties as shown in Table 5.3).

Table 5.7: GWR over OK in USD

County ID	SVD-GWR MAE	OK MAE	GWR Improvement
3101	78,300	83,500	6.2%

Table 5.8: GWR over SVD-RK in USD

County ID	SVD-GWR MAE	SVD-RK MAE	GWR Improvement
1286	76,800	69,600	-10.3%
2061	64,200	55,700	-15.3%

Since removing outliers for county id 3101 improved the OK method (shown in Table 5.6), we also wanted to determine if removing outliers would improve the SVD-GWR method. Table 5.9 shows that there was an improvement to the error in comparison to SVD-GWR without being treated for outliers, however the improvement of OK treated with outliers in Table 5.6 is still the method that produced the lowest error for county id 3101.

Table 5.9: GWR over OK without Outliers in USD

County ID	SVD-GWR MAE	OK MAE	SVD MAE
3101	64,000	54,000	104,000

5.6 Conclusion and Future Work

Housing price estimation is a challenging task that depends not only on properties of a house, but also on its location. We propose a system that leverages the predictive power

of a latent-feature based recommender engine using truncated singular value decomposition (SVD). As such systems are spatially oblivious, we integrate the result into regression kriging. This way, kriging fits a model to a data set in which gaps are filled by SVD, and where outliers, which cannot be explained by the main latent features of the data, are removed by truncation of non-principal components of SVD. Our experimental evaluation shows that our proposed integrated approach SVD-RK outperforms both standalone SVD, standalone ordinary kriging, and universal kriging in two out of three study areas where the variogram fit shows a degree of trend. However, for the study area that has a good variogram fit a straightforward application of our proposed approach may fail. Therefore, we propose to augment our approach using an outlier detection approach, as well as combining it with GWR to better account for outliers in the data.

Our work shows that, given only basic information about houses such as the number of rooms and bathrooms, the combination of SVD and kriging yields the best of both worlds of recommender systems and spatial interpolation. A next step of this work is consider further how to improve ordinary kriging when the variogram fits the data well which is explored in the next chapter.

Chapter 6: Data Driven Urban Planning with Spatial Statistics and Recommendation Systems

Abstract

Kriging is a geostatistical method to predict a variable in a specified location using the weighted average of known values in the spatial neighborhood of the location. Using Yelp business review data for training, we leverage kriging to augment a singular value decomposition (SVD) based recommender engine to predict the rating of a specified business in a specified location. Kriging not only returns an estimated rating based on ratings for similar businesses, but it also specifies its level of confidence. This allows the recommender engine to locally weight the result of the kriging estimator and the SVD estimator. Our experimental evaluation shows that our kriging-augmented recommender system (KARS) significantly outperforms the predictive power of classic kriging and classic recommender systems. We leverage KARS for an urban planning system that can recommend the best business to build in specified location, and that can specify the best location to build a specified business.

6.1 Introduction

With the rapid emergence of location-based social networks, such as Foursquare Swarm[1] and Yelp[2] which allow users to rate locations such as stores and restaurants, recommender systems have become a powerful tool to recommend places to users [79, 80, 81, 82]. While the existing work focuses on recommending locations to users, this work approaches the problem of recommending potential locations to build a new business such that predicted business ratings are maximized.

[1] www.swarmapp.com
[2] www.yelp.com

In this study, business-based location recommendations have applications in two use-cases: The first use-case selects the best location for a specified business by predicting where the highest rating of a business would be. This use-case is paramount for a business to select locations that maximize future user-reviews and thus, the success of the business. The second use-case assumes a given location for which a business (or type of business) is to be recommended for construction. This use-case is taken from the perspective of urban planning, thus choosing business types that best fit the specified location. Suburbs in the United States and around the world have become as diverse as cities in regard to race, culture, income, age, sexuality, and lifestyle [83], mimicking cities through mixed-use development town centers [84, 85, 86]. Traditional market analysis in urban development would include but not limited to 1) determining the spending pattern of the surrounding population (i.e. where they shop or how much they spend), 2) documenting the type size, and location of existing and planned competitive retail facilities that are nearby and in the region, and 3) conducting a site and traffic analysis to ensure that the project development can be accommodated [83]. Our approach for business location recommendation does not replace, but, *enhances* the traditional market analysis for urban planning with a data-driven approach.

Specifically, in this work we leverage recommender engines to predict the user-rating of a specified business in a specified location. To predict user ratings, we leverage that this or similar businesses may have been observed elsewhere, and that other businesses may have been observed nearby. Given the name of a business (such as "Starbucks" or "Andy's Donuts") and a geospatial location, our system returns an estimated average user ratings. For this purpose, we go beyond classic matrix factorization based recommendation to leverage latent similarity between different businesses. We argue that location matters and that user ratings are location sensitive for different business. Thus, we leverage the geostastical method of kriging to leverage the spatial autocorrelation between business ratings.

This way, we can not only leverage similar businesses for prediction, such as "people who liked that coffee chain may also like this coffee chain", but may also leverage location

by predicting that "people in this location seem to particularly like coffee shops".

Our work is structured as follows. After formally defining the problem of location-based business rating prediction in Section 6.2 we survey existing solutions for spatially aware recommender systems in Section 6.3. Then, Section 6.4 introduces our proposed approach. Our experimental evaluation in Section6.5.3 shows that the combination of both spatial statistics and matrix factorization improves recommendation accuracy compared to traditional solutions.

6.2 Problem Definition

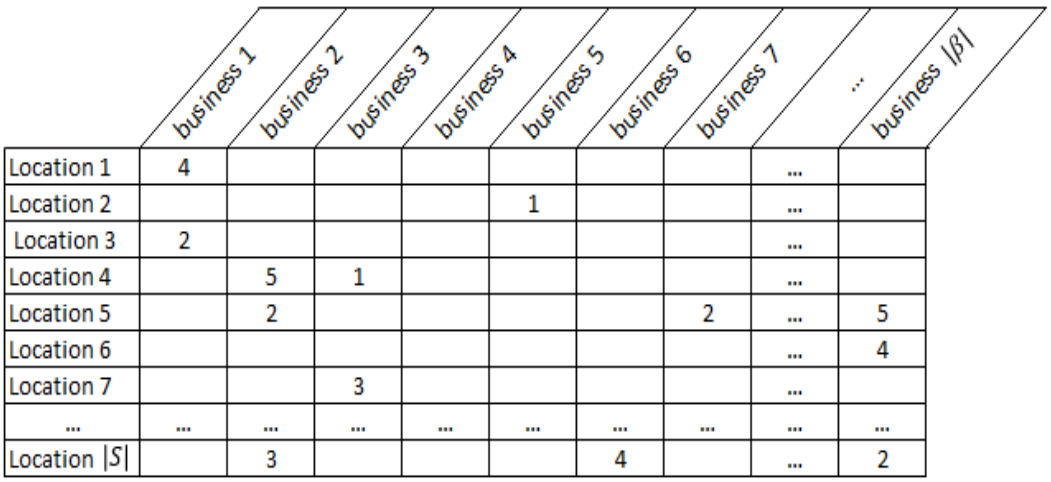


Figure 6.1: SVD Matrix Structure

In this section, we formally define the problem of location-based business rating prediction. In a nutshell, our goal is to predict the average rating that a new business b will have in location l. TO make such prediction, we first require a database of existing businesses, their locations and their ratings.

Definition 8 (Rating Database)**.** *A rating database $\mathcal{D}$ is a collection of triples $(l, b, r) \in \mathcal{S} \times \mathcal{B} \times \mathcal{R}$, where $\mathcal{S}$ is a spatial domain of locations, $\mathcal{B}$ is a nominal set of nominal business names, and $\mathcal{R}$ is a range of ratings that we assume to be in the interval $[1, 5] \subset \mathbb{R}$.*

Figure 6.1 shows an example of a rating database in matrix notation. Note that each business (column of the matrix) may have multiple values, as the same business (e.g., "Starbucks") may have branches in multiple locations, an that each location (line of the matrix) may have multiple values, as many businesses may co-locate in the same building or areal unit. Given a rating database, the task of location-based business rating prediction is formally defined as follows.

Definition 9 (Location-Based Business Rating Prediction)**.** *Let $b \in \mathcal{B}$ be a business in location $l \in \mathcal{L}$ having rating $r \in \mathcal{R}$ and let $\mathcal{D}$ be a rating database. The task of location-based business rating prediction is to infer r given l, b and $\mathcal{D} \setminus (l, b, r)$.*

As an example, Figure 6.2 shows locations and ratings of business "Walgreens" in the greater Phoenix region. We see that different locations yield vastly different ratings. While such differences may be partially explained by non-spatial factors, such as the quality of service at different locations, we observe spatial auto-correlation in the data, indicating that location matters.

6.3 Related Work

The goal of this work is to incorporate spatial statistical elements into recommender engines and improve upon what researchers in this field have already studied. In this section we first look at how traditional urban planning systems have been implemented with the integration of spatial information. Next we review recommender engines that have been developed for urban planning. Spatially-Aware recommender systems are related as surveyed in Chapter 4.2.2. Machine learning with spatial statistics are also related as surveyed in Chapter 5.2.1.

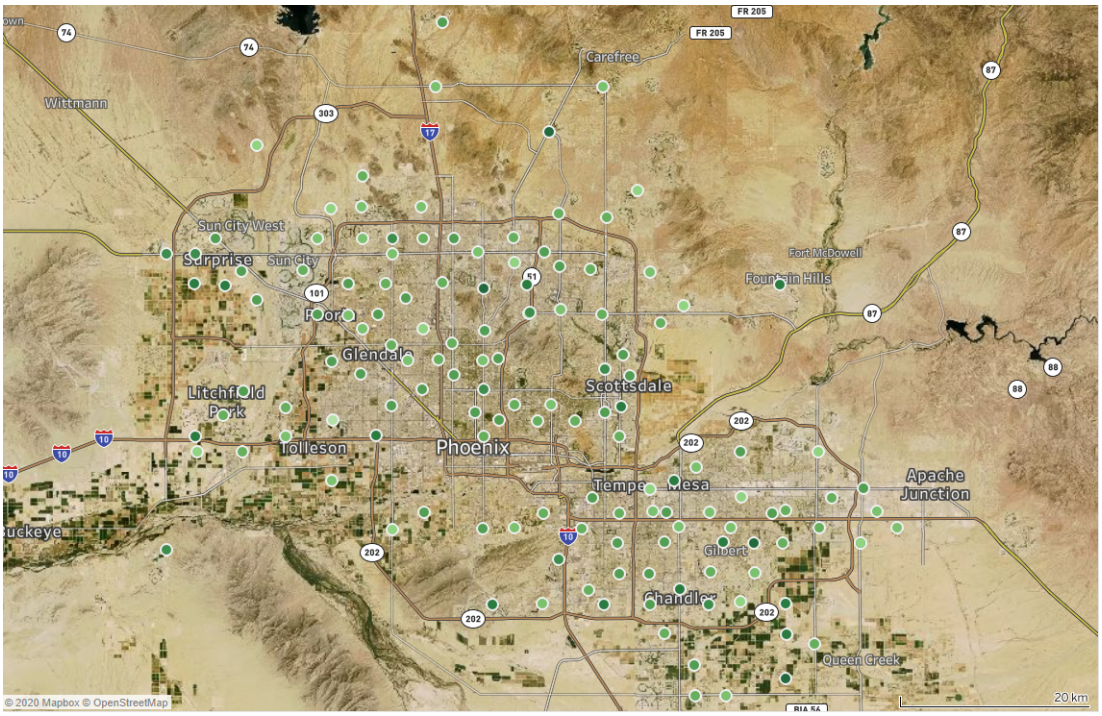


Figure 6.2: Walgreens Ratings

6.3.1 Traditional Urban Planning Systems

Traditional urban planning systems were developed very differently with the integration of traditional geographic information systems (GIS) for the purpose of location analysis. Birkin et al. [87] outlined a framework within which GIS can be linked with other analytical tools which could lead to the creation of a decision support system. One example mentioned in this article is the location-allocation model which can find the optimal locations for supply points given a spatially non-uniform pattern of demand.

Harris et al. [88] also describes possibilities for how GIS could be integrated with urban planning systems so that retail managers could then address a wider range of operational and strategic issues. The role of GIS would be to act as the display engine for urban

planning systems which produces maps and charts that summarize the urban planning system operations. Harris et al. further explains that the integration of GIS could also be used to generate variables relevant to the work of urban planning systems that are not directly accessible such as the average slope or elevation of an area or the proportion of areas in floodplains.

Another consideration to urban planning systems was described by Birkin et al. [89] which discusses refining models to capture complex types of consumer behavior to plan for new store openings, branches opening within shopping centers, incorporating shopping from the workplace, and spatial modeling to handle complicated consumer markets like petrol retailing. Birkin et al. expands on the classical model which incorporates spatial interactions of consumers (i.i. flow of people or money from residential areas to shopping centers and cost of travel and distance from residential areas to shopping centers). The extensions of the classical model include variables such as demand for a product in a particular zone and estimated sales for a product in a particular zone at a specified destination. While these models attempt to integrate spatial information into urban planning Birkin et al. states that to solve the challenging problems of modeling consumer behavior, it may require imaginative extensions of an existing framework, the introduction of completely new techniques, or a recourse to old-fashion but robust methods.

6.3.2 Modern Urban Planning Recommender Systems

Most modern recommender systems use collaborative or content-based filtering techniques allowing the customer to sift through items that may not be of interest and to help them navigate to personalized and relevant items faster [7]. Recommender engines are typically used for e-commerce for companies such as Netflix and Amazon. However, the same algorithms can be fine tuned to be used as recommender engines for urban planning as well.

A zone recommendation system was developed for physical businesses in an urban city, which uses both public business data from Facebook and urban planning data [90]. This approach consisted of classification algorithms, specifically Random Forest and Support

Vector Machines, that take in the metadata of a business and output a list of recommended zones to establish the business in. The zones refer to 55 urban planning areas, the boundaries of which are set by the Singapore government. Because their recommendations was only based on broad zones, the recommendations do not provide sufficiently granular information for business owners for specific locations. Take for example the business Starbucks. The recommender engine may predict that Washington, DC is a great location for Starbucks but the demographics of Washington, DC is diverse in which the location recommendation is not detailed enough as to where exactly the success of Starbucks would be.

There has been research in understanding the potential for data driven methods in urban planning. A recent study applied data driven methods for urban planning through extracting emotion information from twitter [91]. This work introduce a citizen-centric urban planning approach that uses tweets to assess citizens' perceptions of the city and associated emotions in an interdisciplinary manner. Their experimental results showed that they were able to identify tweets carrying emotions and that their approach bears potential to reveal new insights into citizens' perceptions of the city. Although not a recommender engine, their study shows how data driven techniques can improve urban planning. A 3-dimenstional multi-resolution framework, called Urbane, has been developed in collaboration with architects that enables a data-driven approach for decision making in the design of new urban development. This was accomplished by integrating multiple data layers and impact analysis techniques facilitating architects to explore and assess the effect of these attributes on the character and value of a neighborhood [92]. This study demonstrates the effectiveness of their data driven approach through a case study of development in Manhattan depicting how a data-driven understanding of the value and impact of speculative buildings can benefit the design-development process between architects, planners and developers.

Our approach spatially improves recommender engines using similar approaches in [66] and [67] but incorporates the machine learning algorithm collaborative filtering algorithms

Normalisation Baseline and truncated SVD. For our algorithm, the baseline estimate outputs are replacement values for when Ordinary Kriging returns a null value. The Kriging outputs are then fed into SVD as well as Baseline (for comparison purposes).

6.4 Methodology

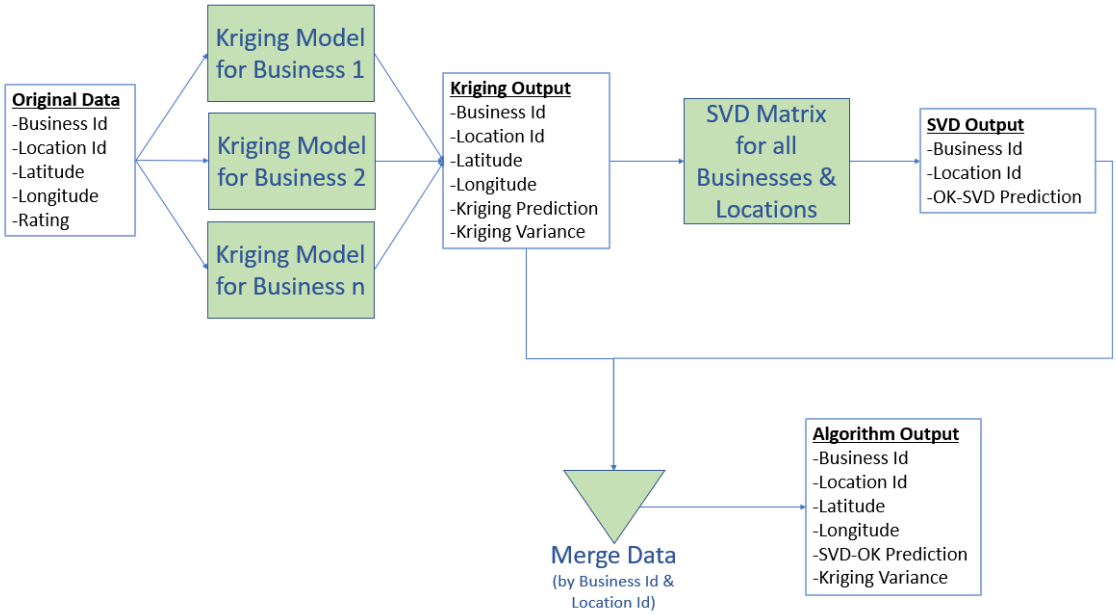


Figure 6.3: Algorithm Flowchart

While truncated SVD has proven its predictive power in many applications such as user-movie recommendation [4] and face recognition [5], it does not explicitly allow to model spatial locations and spatial auto-correlation. This limitation is evident due to SVD using linear algebra only, which is too limited to describe spatial trends. Therefore, in the following, we propose a hybrid approach which augments the power of latent factor modeling using truncated SVD with a kriging approach that gives spatial information appropriate

special treatment.

Ordinary kriging is based on the assumption that variation is random and spatially dependent, and that the underlying random process is intrinsically stationary with constant mean and a variance that depends only on separation in distance and direction between places and not on absolute position [6]. We believe that yelp ratings are stationary with constant mean but it is difficult to determine if there is trend in which the process should be modelled as a combination of a deterministic trend plus spatially correlated random residuals from the trend [6]. An experimental variogram that increases with increasing gradient as the lag distance increases usually signifies trend [6]. For example, Figure 6.5 shows the experimental variogram of the Yelp data set depicting the behavior of no trend which implies that OK is suitable for this data, not UK. In addition, we observed the difference in the root mean squared error (RMSE) to determine which type of kriging was most optimal for this data set. From Table 6.1 we see that OK performs better than UK thus we can infer that the data is stationary and that the mean and variance of the data stays constant.

In order to produced low errors in the prediction, our algorithm utilizes the strengths of each algorithm separately. For kriging to make accurate predictions, the kriging models must be segmented by business where all rating values are to be associated with the same business. If all businesses were kriged at once then an unknown point to be interpolated, such as Starbucks, would incorporate rating values of all stores nearby, such as McDonald's, Whole Foods, and Dick's Sporting Goods which are not related to Starbucks. Thus we apply kriging to each business individually. On the other hand, with the SVD algorithm, all businesses can be incorporated in the matrix to make predictions of the unknown points and is not required to be segmented by business name. Thus, our algorithm uses one large SVD matrix utilizing all the information from all businesses being analyzed to produce the predictions, which is a typical structure for SVD. The following subsections describe our approach, as shown in Figure 6.3, to enhance truncated SVD by ingesting spatial predictions.

6.4.1 Step 1 : Building Kriging Models

The first step in the OK-SVD algorithm is to build ordinary kriging models for every business individually as shown in Figure 6.3. The variograms will be generated using the observed ratings that are available in the data set. The points of interest will be interpolated using the kriging model. The points of interest values that have been interpolated from kriging will then be fed into the truncated SVD matrix cells.

6.4.2 Step 2 : Building SVD Matrix

Once OK models for each business is performed to produced the initial predictions, the outputs for each point of interest are fed into the large SVD matrix. Once truncated SVD is applied with the spatial predictions, we call the output OK-SVD as shown in Figure 6.4. We chose to feed kriging into SVD, rather than SVD into kriging since kriging by itself has proven to have a lower error when both approaches are compared individually, as shown in Table 6.1 in Section 6.5.3.

6.5 Experimentation

6.5.1 Yelp Data

In order to test the proposed algorithm, the data must have spatial information with each rating. Thus, we used the Yelp data from the Yelp Data Challenge [3]. The focus of this study was only around the businesses that were within the bounding box defined in Table 6.5. The columns used included business_id, name of business, latitude, longitude, and star rating. Once the data was cleaned, there were 13,801 data points remaining to execute the experiments for KARS. As described in Section 6.4, the kriging models were segmented by business name. From the cleaned data, 194 business names were utilized to build individual models in kriging. These 194 business names were also fed into the one large SVD matrix with a dimension of 4213 by 194 which creates a matrix of 817,322 cells.

[3] https://www.yelp.com/dataset/download

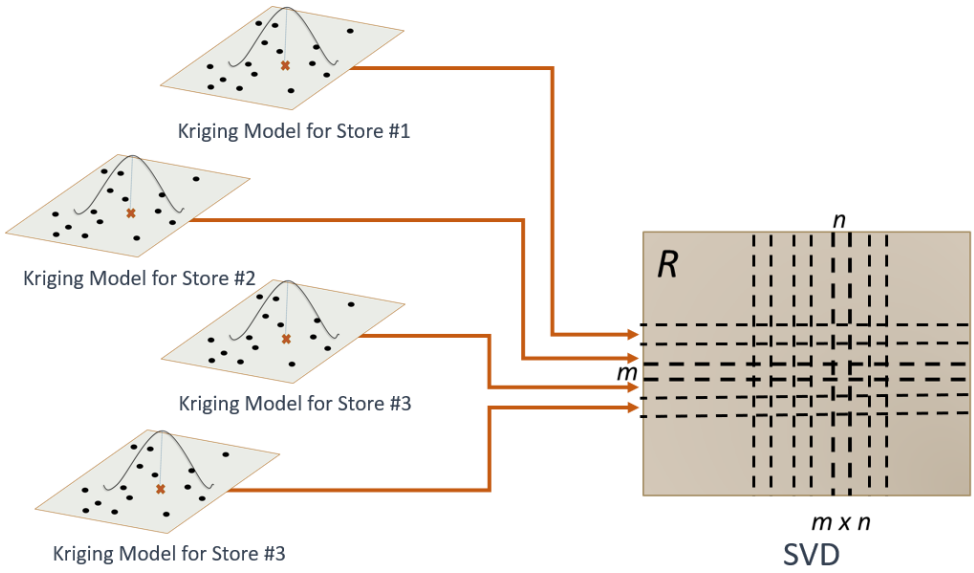


Figure 6.4: Algorithm Structure

6.5.2 Competitor Algorithms

This section describe the baseline approaches used in our experiments.

Kriging-Augmented Recommender System (KARS-θ)

This denotes our approach as described in Section 6.4. KARS requires the fraction θ of values filled in the rating matrix as a parameter. As a special case, KARS-0 denotes the special case where only the cell $\mathcal{D}_{l,b}$ to be predicted is filled using kriging.

Ordinary Kriging (OK)

This approach uses the classic ordinary krigign approach as described in Equation 1.

Universal Kriging (UK)

Universal kriging is a more general approach to kriging that avoids the assumption of stationarity (i.e., having a constant rating mean) across space [6] but still assuming homoscedasticity (constant rating variance) across space.

SVD

This is the truncated SVD approach, as described in Section 2.2. This baseline does not use any kriging to spatially interpolate missing values.

Normalisation Baseline (Baseline)

As a spatially oblivious baseline estimate to estimate missing values in a matrix we use the following estimation approach [32]:

$$b_{i,j} = \mu + b_i + b_j \tag{6.1}$$

Where μ is the average of all ratings, b_i and b_j indicate the observed rating deviations of location l and business b respectively from the average. For example, if the average of all ratings is 3.6, location l has an average rating of 3.1 (thus having a deviation of b_i=-0.5), and business b having an average rating of 4.5 (thus having a deviation of $b_j = +0.9$), we would predict a rating of $b_{i,j}$ of $3.6 - 0.5 + 0.9 = 4.0$ for business b in location l.

In order to reduce the estimation bias for b_i and b_j for low sampling sizes, we use the baseline regularization approach proposed in [32]:

$$\min_{b*} \sum_{i,j \in \kappa} (r_{i,j} - \mu - b_i - b_j)^2 + \lambda_1 (\sum_i b_i^2 + \sum_j b_j^2), \tag{6.2}$$

where there first term $(r_{i,j} - \mu - b_i - b_j)^2$ minimizes the squared error, while the regularization term $\lambda_1(\sum_i b_i^2 + \sum_j b_j^2)$, avoids over-fitting by penalizing the magnitudes of the parameters [31].

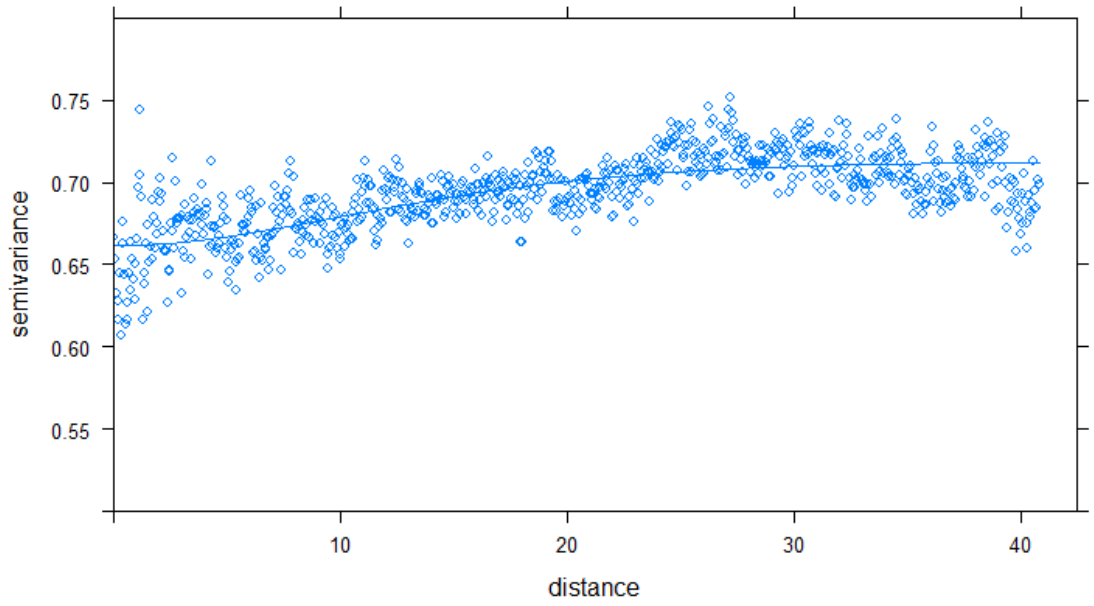


Figure 6.5: Yelp Variogram (in Miles)

In a nutshell, the idea is that for a given (item, user), we estimate a higher rating if the user generally gives high ratings, and the item generally receives high ratings. This baseline approach (denoted as "Baseline" in the remainder of this work) does not consider spatial location like kriging, nor is it able to leverage the similarity latent features of users and items like SVD.

6.5.3 Experimental Results

Table 6.1 shows us that OK performs significantly better than either the collaborative filtering Baseline estimate or the machine learning algorithm SVD. We can infer from these results that the data is much more spatially correlated than there is hidden patterns within the data. Because OK outperforms SVD or Baseline, the OK outputs are fed into SVD to further improve the estimates by finding hidden patterns within the OK predictions as shown in Figure 6.4. However, because OK will sometimes not have enough nearby points

to make predictions to all points, the Baseline output is adopted for those entries that are returned with a null value from Kriging since Baseline will always make a prediction for every point in the matrix. We use Baseline instead of SVD to replace the null values from Kriging because as shown in Table 6.1, Baseline alone does 6.7% better than SVD alone.

The Yelp data proves to have spatial correlation which can be improved upon by finding hidden patterns within the spatial predictions. As shown in Table 6.2, we found that the simple Baseline estimation alone performs better than the more complex algorithm SVD. Even though SVD is a more robust algorithm, the Baseline estimate is a remarkably effective approach when the data is relatively small and has low density [93]. However, when the OK estimates are fed into Basline and SVD, the simple Baseline performs similar to that of SVD when fed the kriging spatial predictions (shown in Table 6.2), thus we feed OK into SVD.

Table 6.1: Baseline Results

Method	RMSE
SVD	1.4084
Baseline	1.3143
Ordinary Kriging	0.7713
Universal Kriging	0.9625

Table 6.2: Our Algorithm

Method	RMSE	Improvement
KARS	0.6913	10.4%
OK-Baseline	0.6920	10.3%

6.5.4 KARS with Variance

Our algorithm, KARS, utilizes the strengths of Kriging first and then SVD. One unique strength of kriging to other spatial statistical methods is that a variance is calculated for every unobserved point which describes the uncertainty of each prediction. These OK outputs are then fed into the truncated SVD algorithm, however, when SVD is used, variances are not calculated since SVD is simply linear algebra. Although KARS feeds the OK predictions into the SVD matrix for further interpolation thus loosing the variance values at the end, we merge the SVD predictions back to the main data frame that contains the OK prediction and variances. From our experimentation's shown in Table 6.3, we found that using the variance from the Kriging portion of KARS, we can improve the error of the algorithm even further. When the SVD predictions are joined back to the main data frame that hold the kriging and kriging variance, we can filter the variance to different thresholds to improve KARS accuracy. As shown in Table 6.3, when we set the kriging variance threshold to at least 0.5, we increase the accuracy to 10.15%.

Table 6.3: KARS with Variance Threshold

Threshold	Improvement	Data Remaining
5	0.59%	99.76%
3	0.61%	98.8%
1	1.22%	91.89%
0.5	10.15%	55.05%
0.1	15.40%	9.25%

When applying the algorithm in this way, we *do* loose data since we only keep the predictions with very small variances, but the predictions that remain are more accurate as shown in Table 6.2 and because the predictions become more explainable. Incorporating the variance can be useful for different applications that do not need all results. For those use

cases that require all results, the variance can be indirectly used as a secondary resource. In this study, we examine two use cases which utilize the variance and will be discussed in the following sections.

6.5.5 Coding Language and Packages

This study was conducted using both the R and Python programming languages. The python package "surprise" was used to apply truncated SVD to the data. Specifically the SVD function, under the matrix factorization algorithms, within the "surprise" package was used to perform truncated SVD. More information on this technique can be found in [69] and [70]. For this function, Table 6.4 describes the hyperparameters used for this study.

Table 6.4: SVD Hyperparamter Values

Parameter	Value
Latent Factors	4
Epochs	15

The R package “gstat” was used in applying kriging to the data. The “variogram”, “fit.variogram”, and “krige” functions were utilized in making the predictions using the truncated SVD outputs. In order to efficiently run the algorithm to incorporate enough data points for kriging, but not to overwhelm the number of points used to fit the variogram function, the algorithm we structure in a unique way. For every store name being analyzed, if the number of training points was equal to or less than 65, a max distance was not specified in the krige function. However, if the number of training points was greater than 65, than a max distance was specified to 20 miles. Applying a max distance means only observations within a specified distance from the prediction location are used for prediction or simulation.

6.6 Applications for KARS

The KARS algorithm has the potential to be utilized in urban planning design. It can be applied to two different applications which will be discussed in the following subsections. With the use of the algorithm, city planners can utilize the robustness of machine learning and the intuitiveness of spatial statistics.

6.6.1 Application One: What to Build?

The first application KARS can be applied not only in determining which stores are most suitable for a specific area given a vacant lot, it can also address which stores may not perform well. This use case can be helpful in designing new town centers or shopping districts. Since our algorithm utilizes spatial correlations from kriging and hidden patterns from SVD, the predictions can aid urban design planners to determine optimal retail stores that could otherwise be overlooked.

Table 6.5: Bounding Box

Limits	X coordinate	Y coordinate
Minimum	33.10	-112.57
Maximum	33.87	-111.25

For this application, we used the data from Yelp in Arizona to apply our algorithm. We first specified our bounding box dimensions which are shown in Table 6.5. We then chose seven different locations for KARS to predict the most optimal stores for those locations given that there is no information at those specific points 6.6. The goal is to determine what stores would be best suited to be built in the seven locations given the information of stores surrounding the area within the bounding box specified. Figure 6.6 maps the seven different locations that have been analyzed for this use case and Table 6.6 lists their

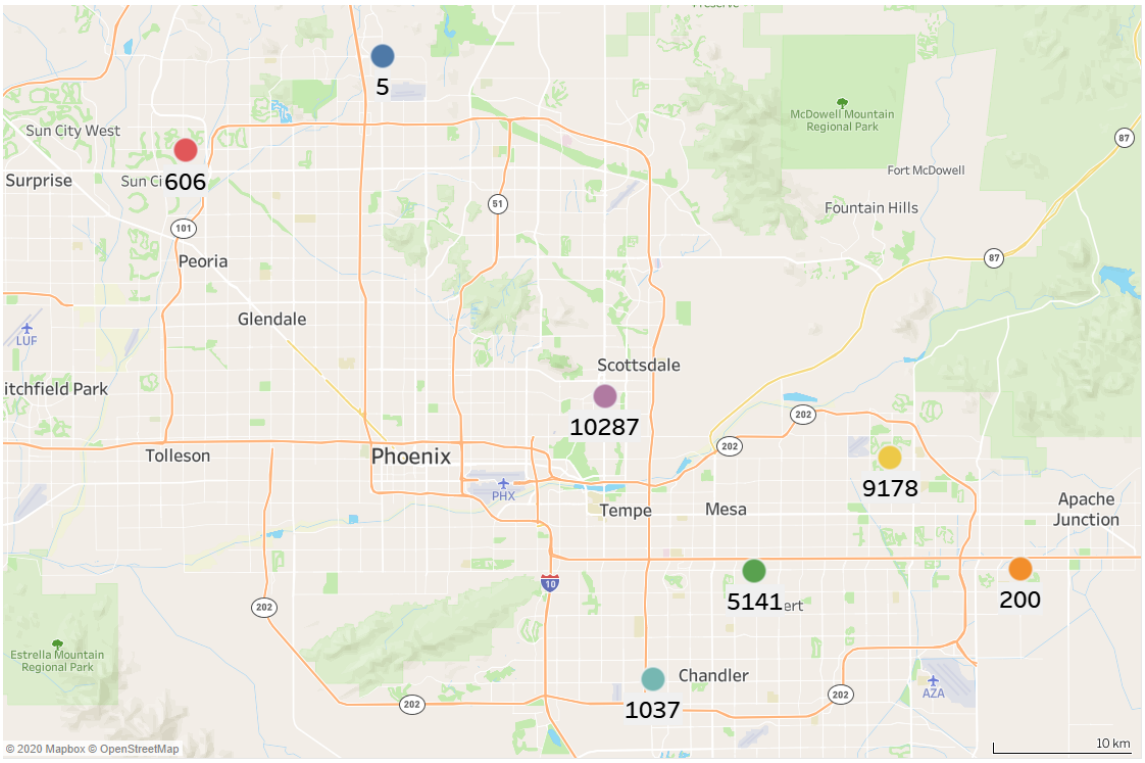


Figure 6.6: Map of Location Id's for Phoenix, AZ

latitude, longitude, and point color associated on the maps discussed in this section. The seven points were then evaluated using the KARS algorithm to find the optimal stores for each location.

Table 6.6: Seven Points for Analysis

Location ID	Map Point Color	Latitude	Longitude
5	Dark Blue	33.71348	-112.0992
200	Orange	33.37962	-111.6012
606	Red	33.65194	-112.2532
1037	Light Blue	33.30793	-111.8884
5141	Green	33.37856	-111.8089
9178	Yellow	33.45231	-111.7028
10287	Purple	33.49220	-111.9255

Table 6.7: Top Results for Location Id 10287

Index	KARS	Name	Variance	95% UL	95% LL
1	4.66	Desert Storage	0.07318	4.13	4.93
2	4.62	Henna Shoppe	0.01767	4.36	4.75
3	4.54	Life Storage	0.04217	4.13	4.74
4	4.47	Hollywood Beauty Eyebrow Threading	0.05324	4.02	4.70
5	4.44	All Mobile Matters	0.07031	3.92	4.70
6	4.14	European Wax Center	0.09564	3.54	4.45
7	4.14	Wildflower Bread Company	0.01102	3.93	4.24
8	4.12	Sunchain Tanning	0.09675	3.51	4.43
9	3.98	Fired Pie	0.08445	3.41	4.27
10	3.97	In-N-Out Burger	0.07132	3.45	4.24

For example, Figure 6.7 shows the top 20 results of the algorithm in descending order of the predicted rating value. In addition, the results show the predictions with the upper and lower bounds to allows the urban planners to have a more intuitive understanding of the prediction. For example, for location id 10287 the highest rating is for Desert Storage but the variance is 0.07318 while DSW Designer Shoe Warehouse is has a rating of only 3.86 but the variance is much smaller which allows urban planners to understand the confidence of the prediction.

Along with traditional market analysis, these results can enable urban planners to focus on specific retail stores with high predicted ratings and eliminate retail stores with very low ratings. Another factor to consider in urban planning is over crowding an area with too many similar or identical stores. SVD-OK makes its rating prediction based on spatial structure and hidden patterns and sometimes a store that is predicted to perform well based on the algorithm may be too close to another identical store. As an example, the store Pita Jungle is number 16 on the top list for location id 10287 (Figure 6.7) and has a rating score of 3.87. However, as shown in in Figure 6.7, the purple point represents location id 10287 while the dark grey points represent the actual Pita Jungle stores that were used to train the model. Since the proximity of the purple point (location id 10287) is relatively close

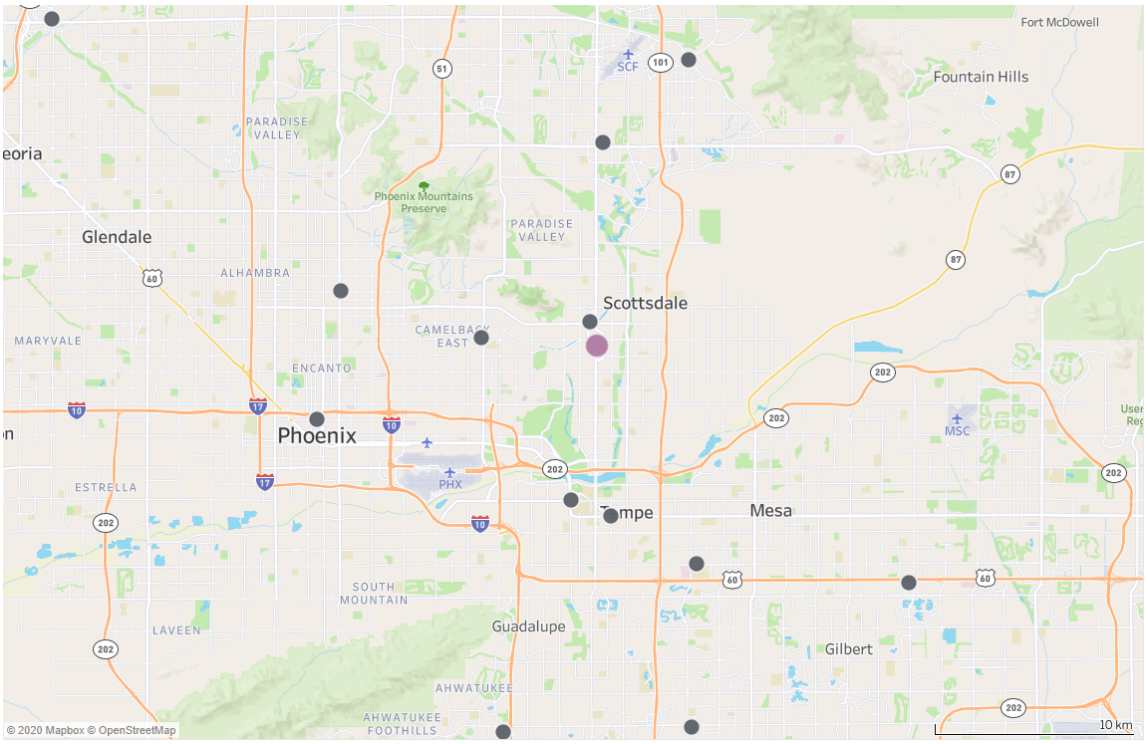


Figure 6.7: Proximity for Pita Jungle

to the gray points which are the chains of that business that already exist, urban planners can decide whether or not to use this predicted store as a potential candidate in designing their town centers.

Wildflower Bread Company is number seven on the top list for location id 10287 and has a predicted rating score of 4.14, which means that this store is predicted to perform well in this location. For this prediction, the other Wildflower Bread Company chains (gray points) may be a great enough distance away from the location id 10287 (purple point) to justify building another Wildflower Bread Company at that specific location. As shown in Figure 6.8, the closest Wildflower Bread Company to location id 10287 is 3.2 miles away. Since this location is projected to have a high rating for this particular store, this enables

urban planners in their decision making process to consider this business as a possibility.

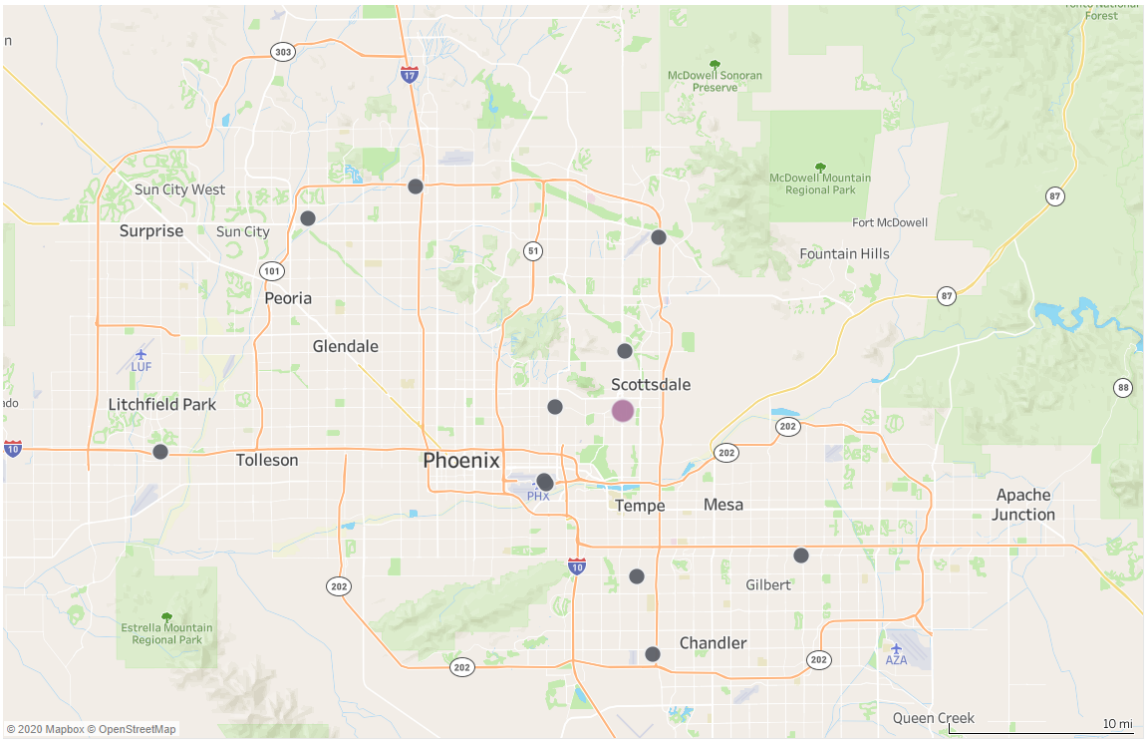
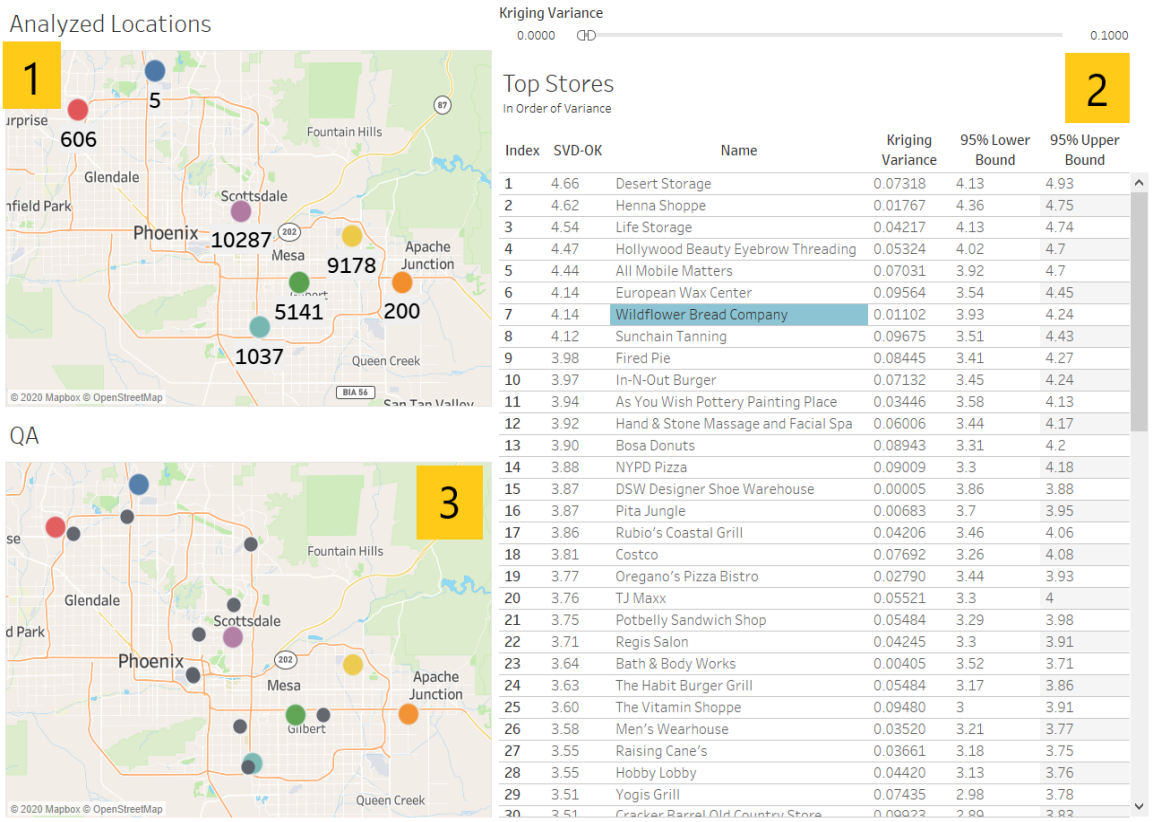


Figure 6.8: Proximity for Wildflower Bread Company

Once the Yelp data is ingested into the algorithm, the output is structured in a way that it can be used to view with the use of analytical dashboards. For this study we ingested the results in Tableau to show how planners could utilize the dashboard more easily during the analysis phase of their projects for building new town centers or city centers. Figure 6.9 shows the Tableau dashboard selected for location id 10287 and has three sections labeled 1-3. Section 1 shows the map of the desired locations to be analyzed. Section 2 allows for variance adjustments and the outputs from the KARS algorithm where the KARS value is ordered in descending order of variance. Urban planners can set the threshold to control and in Figure 6.9 the variance is set to 0.1. which means only results with variances 0.1 or less will be shown in the Top Stores list. Once the Top Stores list in section 2 is displayed, urban planners can select a store to visually see how far spaced other current stores with the

same name are located in comparison to the selected location id which is shown on a map in section 3. This enables decision makers to quickly evaluate which stores (with the most predicted certainty) have the highest/lowest projected ratings and then determine if other similar stores are a great enough distance to be further evaluated as a potential option.



Index	SVD-OK	Name	Kriging Variance	95% Lower Bound	95% Upper Bound
1	4.66	Desert Storage	0.07318	4.13	4.93
2	4.62	Henna Shoppe	0.01767	4.36	4.75
3	4.54	Life Storage	0.04217	4.13	4.74
4	4.47	Hollywood Beauty Eyebrow Threading	0.05324	4.02	4.7
5	4.44	All Mobile Matters	0.07031	3.92	4.7
6	4.14	European Wax Center	0.09564	3.54	4.45
7	4.14	Wildflower Bread Company	0.01102	3.93	4.24
8	4.12	Sunchain Tanning	0.09675	3.51	4.43
9	3.98	Fired Pie	0.08445	3.41	4.27
10	3.97	In-N-Out Burger	0.07132	3.45	4.24
11	3.94	As You Wish Pottery Painting Place	0.03446	3.58	4.13
12	3.92	Hand & Stone Massage and Facial Spa	0.06006	3.44	4.17
13	3.90	Bosa Donuts	0.08943	3.31	4.2
14	3.88	NYPD Pizza	0.09009	3.3	4.18
15	3.87	DSW Designer Shoe Warehouse	0.00005	3.86	3.88
16	3.87	Pita Jungle	0.00683	3.7	3.95
17	3.86	Rubio's Coastal Grill	0.04206	3.46	4.06
18	3.81	Costco	0.07692	3.26	4.08
19	3.77	Oregano's Pizza Bistro	0.02790	3.44	3.93
20	3.76	TJ Maxx	0.05521	3.3	4
21	3.75	Potbelly Sandwich Shop	0.05484	3.29	3.98
22	3.71	Regis Salon	0.04245	3.3	3.91
23	3.64	Bath & Body Works	0.00405	3.52	3.71
24	3.63	The Habit Burger Grill	0.05484	3.17	3.86
25	3.60	The Vitamin Shoppe	0.09480	3	3.91
26	3.58	Men's Wearhouse	0.03520	3.21	3.77
27	3.55	Raising Cane's	0.03661	3.18	3.75
28	3.55	Hobby Lobby	0.04420	3.13	3.76
29	3.51	Yogis Grill	0.07435	2.98	3.78

Figure 6.9: Tableau Dashboard for Location Id 10287

6.6.2 Application Two: Where to Build a Walgreens?

Another application in data-driven urban planning is using the KARS algorithm for business's looking to find the optimal location for their next store by predicting where the highest rating would be. In this use case we explore where an optimal store should be located given the bounding box specified in Figure 6.5. In this use case we create a grid of

10,374 points that span the bounding box. We use the Yelp data to train our model, and use the grid points to make predictions in each location. The goal is to find the highest rating prediction with a reasonable variance to depict where an optimal location for a number of businesses could be. For this application we focus on four different stores which each have a different number of training points shown in Table 6.8. The purpose of this table is to show how well the algorithm performs given a sufficient amount of data (Walgreens) versus a very small amount of data (Bath & Body Works).

Table 6.8: Selected Businesses

Store Name	Count
Walgreens	142
Supercuts	56
One Stop Nutrition	23
Bath & Body Works	15

As with the application in subsection 6.6.2, once the algorithm is applied to the data and connected back to the main data frame, the data is appropriately structured to be ingested into a BI visualization tool to build dashboards. In this study, for this application, we built a dashboard in Tableau in order to display the predictions for the area of interest and the variance associated with each prediction. Some predictions do not have variances calculated since those points are to great a distance from any training points. These points still have a predicted value from KARS but have no variance. Thus if a user chooses too, the prediction can be used, however, for the dashboards shown in maps 6.10 and 6.11, the predicted ratings of all locations within the bounding box that have a variance calculated from the kriging portion of KARS are displayed.

This application can be useful for businesses looking to expand their business and find optimal locations based on the status of the stores currently opened in the area of interest. The dashboard is particularly useful because it gives the user a quick visual of the spatial range of possibly star ratings and gives the user control over the variance threshold.

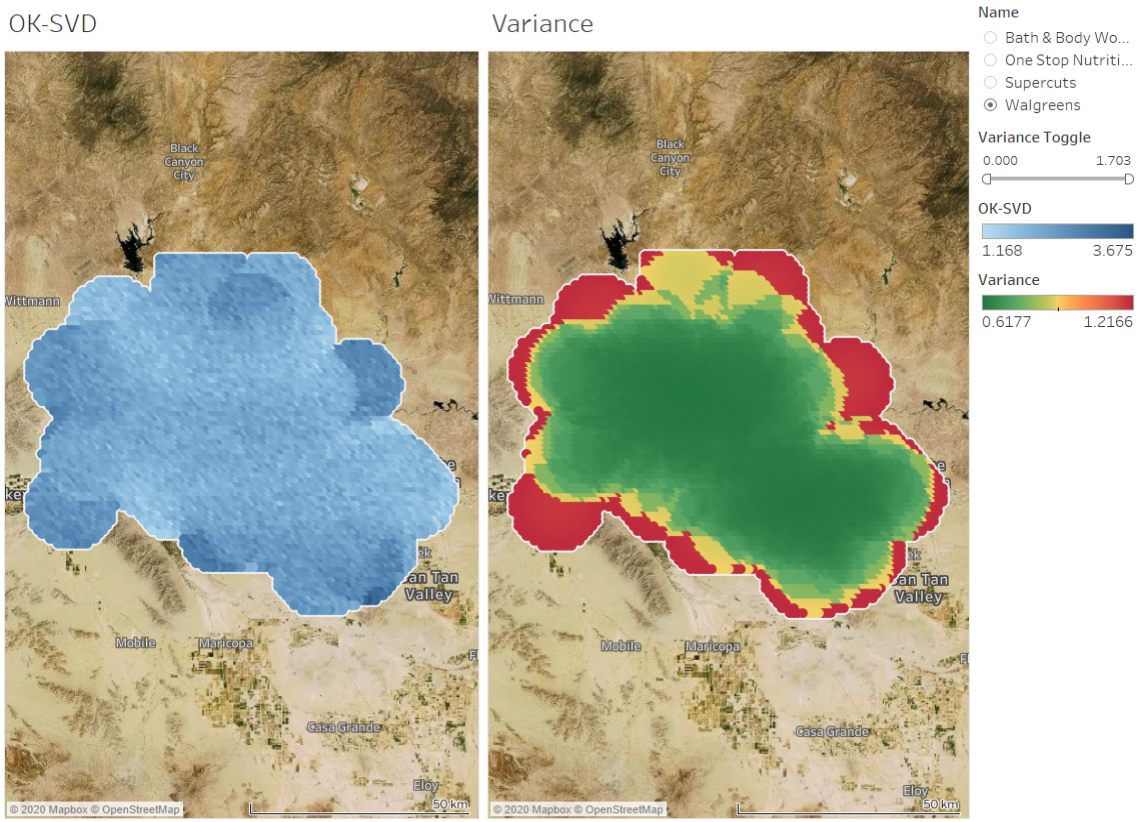


Figure 6.10: Tableau Dashboard for Walgreens

6.7 Conclusion and Future Work

Predicting how well a retail store will perform is a challenging task that depends not only on the type of store , but also on its location. We propose a system that leverages the predictive power of a latent-feature based recommender engine using truncated singular value decomposition (SVD). As such systems are spatially oblivious, we integrate the result into ordinary kriging. This way, kriging fits a model to a data set in which gaps are filled by SVD.

Our experimental evaluation shows that our proposed integrated approach KARS outperforms both standalone SVD, standalone ordinary kriging, and universal kriging. Our work shows that, given only basic information about a store, the combination of SVD and

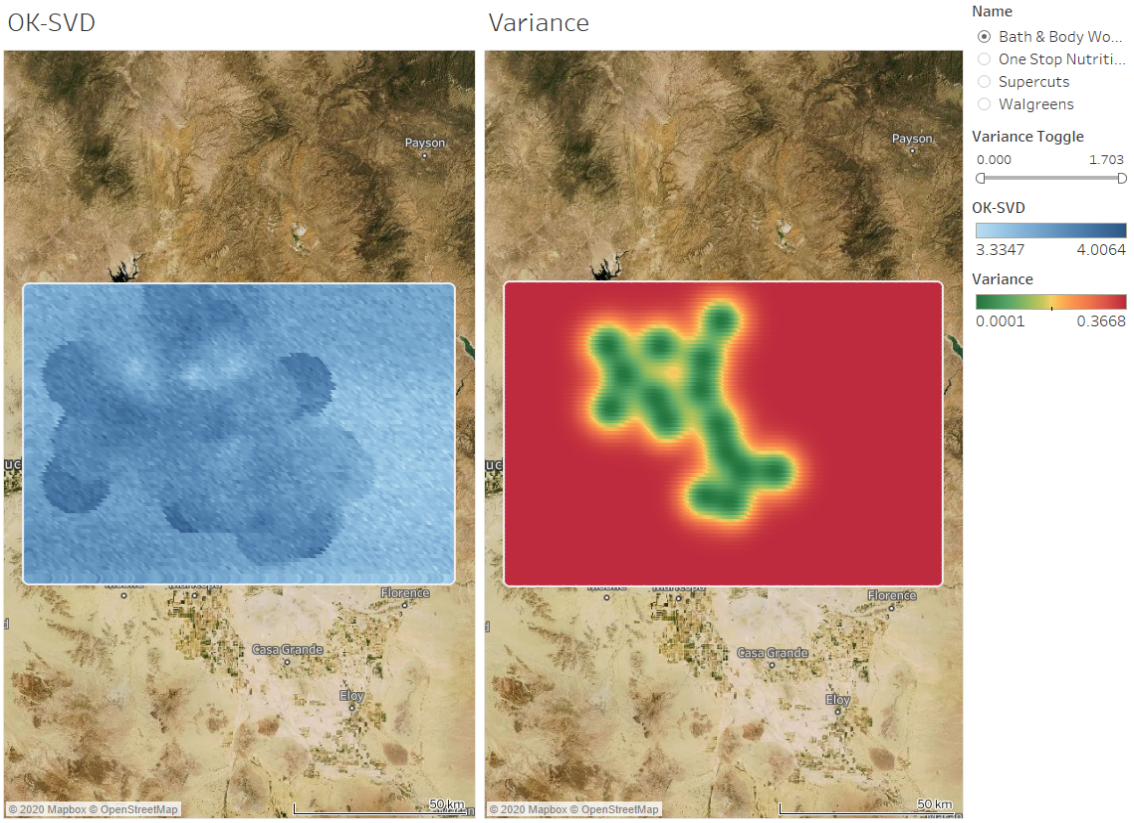


Figure 6.11: Tableau Dashboard for Bath & Body Works

kriging yields the best of both worlds of recommender systems and spatial interpolation. In the future, we plan to apply this approach on a user level. Instead of aiding urban planning designers and businesses in finding the optimal store and location, we plan to use the algorithm to recommend stores to users based on their rating history.

Chapter 7: Additional Work & Future Considerations

This chapter discusses topics that could be further explored using the algorithms developed in this thesis to enhance recommender systems using spatial statistics. While some sections may have run actual mini-experiments with data that was utilized in this thesis, other sections are purely theoretical and act as frameworks for future research.

7.0.1 Solving the Issue of Sparse Matrices

Introduction

Since the re-emergence of SVD in 2015, this algorithm has gained momentum due to its remarkable performance in recommender systems. With such promising results, a great deal of research has been done and many enhancements have been made to the algorithm working to improve various aspects of it, specifically the acute challenge in the lack of information which causes extremely sparse rating matrices. The reason for this challenge is because, in real-life scenarios, users for a given application do not always rate consistently. For example, Netflix and Movilens data sets have 1.18% and 6.3% density respectively which signifies that only a few items are rated by users [4, 94]. Thus, a number of studies have been done to address and mitigate the issue of sparsity.

A study published an algorithm which combined the classic matrix factorization method with a specific rating elicitation strategy to best approximate a given matrix with missing values in order to create a higher density matrix to improve the challenge of sparse matrices. A new matrix factorization model, called Enhanced SVD (ESVD), was developed which incorporates the classic matrix factorization algorithms with ratings completion inspired by active learning. Active learning algorithms are effective in reducing the sparsity problem for recommender systems by requesting users to give ratings to some items when they

enter the systems [15]. Unlike traditional active learning that queries only new users for a certain number of ratings in each iteration, ESVD predicts these specific ratings for all the users at the same time (one iteration) based on the result of applying matrix factorization algorithms to this sub-matrix. After these ratings are added to the original rating matrix, a more accurate matrix factorization model could be trained. This study also proposed multi-layer versions of ESVD which obtains the fillings of the matrix incrementally through multiple matrix factorization on different sub-matrices.

Another study [16] attempting to mitigate the same problem of the ever increasing sparsity of a matrix proposed a new hybrid model by generalizing a contractive auto-encoder paradigm into the matrix factorization framework with good scalability and computational efficiency. This study states that most approaches that aim to alleviate a sparse matrix depend on hand-crafted feature engineering, which they believe to be noise-prone and biased by different feature extraction and selection schemes. Instead, the 2017 study jointly models content information as representations of effectiveness and compactness, and make use of side information (implicit user feedback) to make accurate recommendations.

A study proposed a recommender algorithm based on a factorized matrix composed of user preferences associated to the movies' genres/categories. They found that the advantage of using such a user-genre matrix factorization model (instead of a user-item matrix) is that it requires less computational resources, as the matrix will be less sparse and at lower dimension, which is a fundamental problem that may reduce the quality of the predictions generated by recommender systems [17]. The user-genre factorized matrix is used in two directions: i) to infer latent factors of genres, as a specific category may have different concepts, each one with distinct levels to the user; and ii) to enrich new users' profiles with predictions of weights for those categories which are still absent in their set of preferences.

While these strategies have shown improvements to the original SVD algorithm for recommender systems, they are oblivious of spatial information. If a data set does have a degree of spatial autocorrelation, then the methods described above would not be able to capture the spatial structure. Thus, this study proposes an approach to improving the

sparsity of a matrix using ordinary kriging. The data set that is used for this study is from the Yelp Data Challenge that was also used in Chapter 6.

Methodology

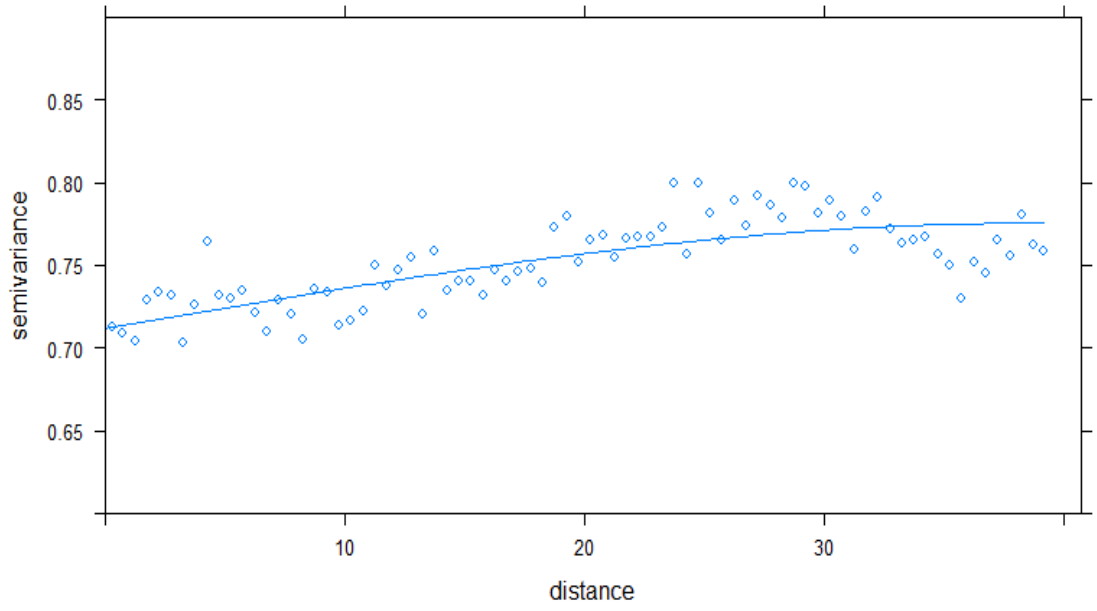


Figure 7.1: Variogram(in Miles)

Typically the denser the matrix is, the better the matrix factorization model is obtained [15]. When a data set, such as the one used in this study, has a degree of spatial autocorrelation, than filling in the original matrix with spatial predictions can improve the predictions of SVD. From visual inspection in Figure 7.1, the covariance function can explain the spatial structure of the data. In Chapter 6, ordinary kriging was integrated with SVD by replacing the cells of interest with kriging values (without increasing the density of the matrix) and found that spatial statistics can improve SVD in this way. In this study, the aim is to

enhance the SVD algorithm by creating a denser matrix by randomly selecting unknown cells and adding values to them derived from the kriging interpolation as shown in 7.3.

In order for ordinary kriging to interpolate points that are not in the data, a random subset of the original data can be extracted and for each unique location, all store names should be replicated. This enables kriging to utilize the points given in the original data, and then perform kriging on the blue cells **and** the orange cells as shown in Figure 7.2. Once this is completed, the predictions are fed into the SVD matrix, resulting in a denser matrix. At this point, SVD can be performed as usual to make predictions on the blue cells (refer to Figure 7.2).

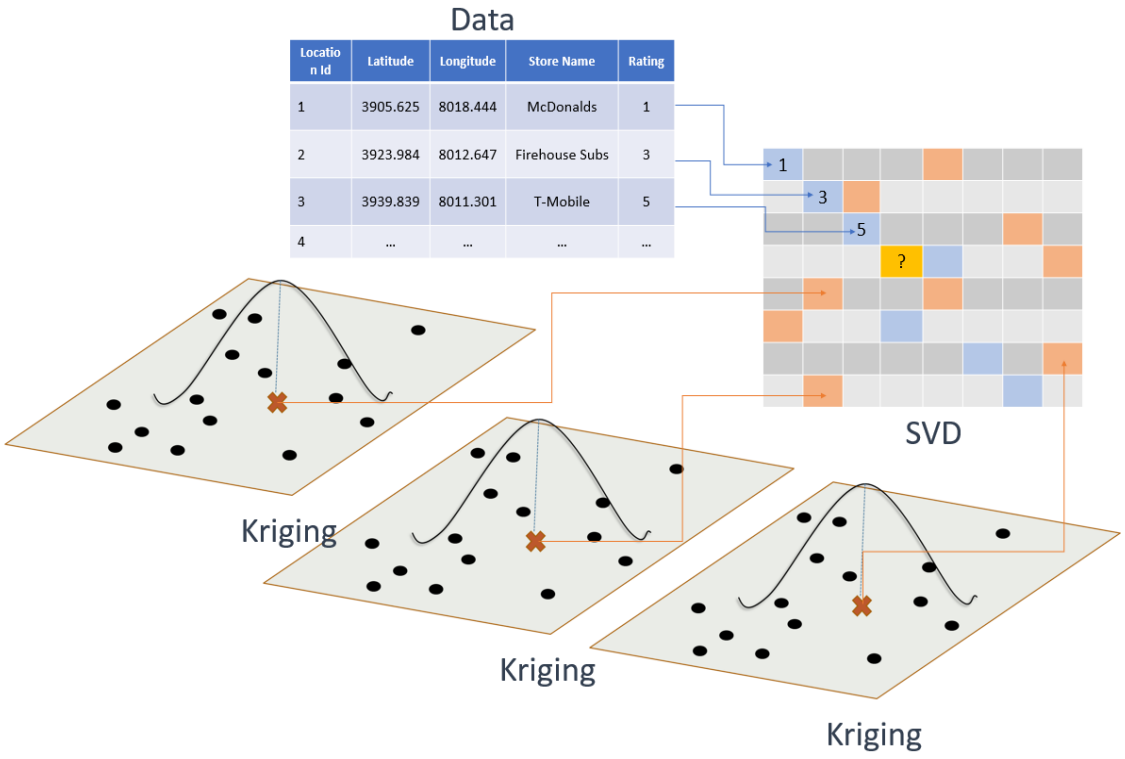


Figure 7.2: Increasing SVD Density

Experimentation

Depending on the density of the matrix, the error of the predictions can significantly decrease. In this study, the SVD matrix was filled by ordinary kriging values in different percentages to control the density of the matrix and test the improvement of the denser matrices compared to the normal SVD as a baseline. The normal SVD is not enhanced with any ordinary kriging interpolated values and is denoted as 0% dense to show that no information has been added other than what is already given to us in the Yelp data set.

Table 7.1: Density Results

Matrix Density	Extra Ratings	RMSE
0%	0	0.8258
10%	161,756	0.6874
45%	688,839	0.6643
100%	1,533,618	0.6555

As shown in Table 7.1, even with a slight increase in density at 10% there is already a significant improvement in the Root Mean Squared Error (RMSE). By filling the matrix by 45%, there is over a 20% improvement in RMSE. These results show that with spatially autocorrelated data, this approach can be utilized to better improve SVD recommender engines by feature engineering other cells in the matrix to reduce sparsity.

While these results show that the method of adding spatial data to random cells in the matrix yields large improvements over SVD, we have found that predicting the points of intrest with ordinary kriging alone still returns better results in comparison to even a matrix with a density of 100% with an RMSE of 0.6531 .

However, when the KARS algorithm, described in Section 6.4, is enhanced by filling the random cells in the SVD matrix with spatially interpolated values, KARS performs slightly better as shown in Figure 7.3. We modified the KARS algorithm and enhanced it by adding

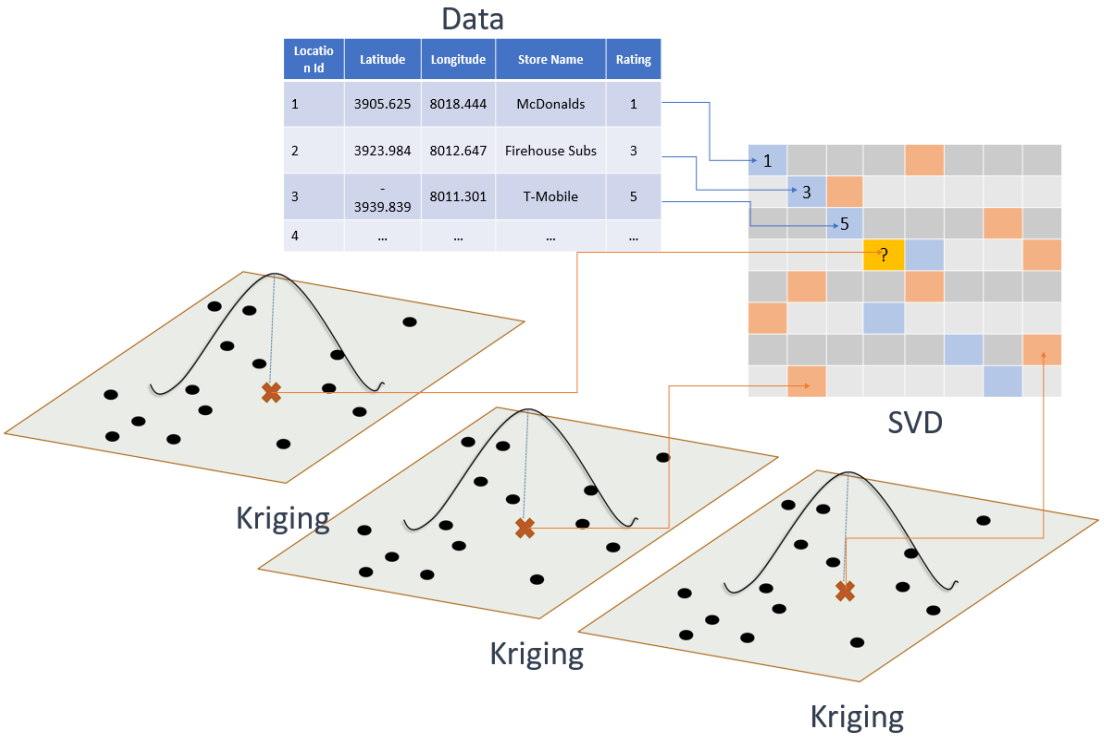


Figure 7.3: Enhancing KARS Algorithm

kriging interpolations values for random points within the matrix in addition to the points of interest. We increased the density of the matrix by 5% (KARS-5), 10% (KARS-10), 15% (KARS-15), 25% (KARS-25), 50% (KARS-50), and 65% (KARS-65) and measured its improvement over KARS as shown in Figure 7.3. We found that filling the matrix with spatially interpolated values for the points of interest **and** random points by 15%, we were able to maximize the increase in accuracy for KARS by reducing the RMSE by 0.80% as shown in Figure 7.4.

Coding Language and Packages

This study was conducted using both the R and Python programming languages. The python package "surprise" was used to apply truncated SVD to the data. Specifically the

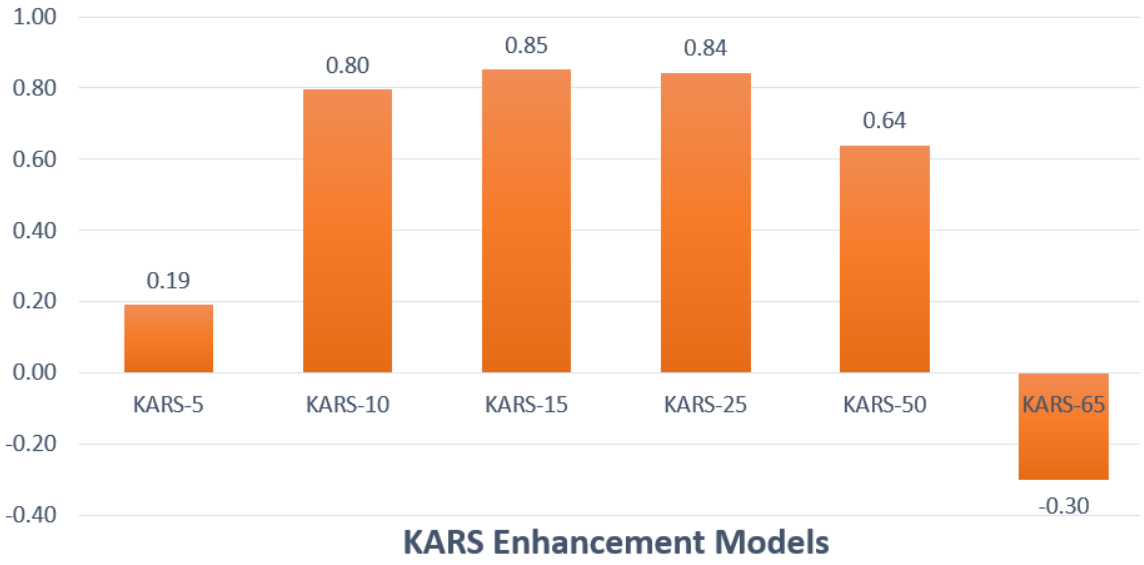


Figure 7.4: RMSE Percent Improvement over KARS

NMF function, under the matrix factorization algorithms, within the "surprise" package was used to to perform truncated SVD. This function ensures that the user and item vectors stay positive. More information on this technique can be found in [69] and [70]. For this function, Table 7.2 describes the hyperparameters used for this study. The R package "gstat" was used in applying kriging to the data. The "variogram", "fit.variogram", and "krige" functions were utilized in making the predictions using the truncated SVD outputs.

Table 7.2: SVD Hyperparameter Values

Parameter	Value
Latent Factors	5
Epochs	70
Learning Rate for b_u	5e-8

Conclusion

Matrix sparsity is an acute challenge in many recommender engines. Due to the lack of explicit information from users, SVD matrices can be imbalanced. In this study, we propose a method to approximate missing values in the matrix to improve the SVD predictions. Since Yelp data can be spatially autocorrelated, as shown in Figure 7.1, adding implicit data from ordinary kriging predictions and increasing the density of the original matrix enhances SVD predictions. From the experimentation, it is evident that even with a small increase in density (9%), SVD can increase in accuracy by 10%.

7.0.2 Solving the Cold Start Problem

Introduction

While recommender systems have become significant tools in e-commerce, the aim is to present relevant items that best meet the preferences of users through a mathematical approach. In doing so, there are a number of acute challenges that can make it difficult to integrate recommender systems in real world scenarios. Subsection 7.0.1, discussed the challenge of sparse matrices while this subsection address the cold start problem faced in recommender systems. The cold start problem is that the recommender engine cannot draw inferences for users or items for which it does not have sufficient information or when there is a new user added to the system which has no rating values on any item. Due to the lack of information, recommender systems cannot make a prediction since they cannot find any pattern between the new user and users already in the matrix that have rated items. For this specific issue, there have been a number of methods developed to address the new user or cold start problem in recommender systems.

A study proposed a method for when a new user is added to the system, using demographic information about a user as an alternative input for a recommender system (such as age, gender, static location). This method assumes that users with similar demographic attributes will rate items similarly and obtains groups of user having similar demographic attributes forming a neighborhood from which newly recommended items are generated [95]. Opinion classification is another technique that was applied to address the cold start problem in recommender systems. Researchers in this study state that ratings are difficult to collect, as they are not spontaneously given by users. However, blogs, forums, social networks, etc. are freely available. Since texts very often carry opinions on topics easy to recognize, the study utilized this idea and propose a method to perform recommendations from unstructured textual data from completely different websites highlighting that it is possible to feed a recommendation system with ratings in this way [96]. Another study made use of association rules and clustering technique for solving cold start problem by

combining two existing approaches in a sequential manner. They first apply association rule technique to expand the user profile as suggested. With the help of this expanded user profile, they apply clustering techniques for recommendation focusing on new item recommendation [97].

While there have been many studies to address the cold start problem, there have been few that utilize spatial elements in their methods. Due to the recent increase in location-based social networks (LBSNs) there has been a large, ongoing amount of geographical user data. Data created from LBSN provides unprecedented opportunity to study user movement to socially understand their behaviors to improve location based applications such as recommender systems. For the cold start problem, when a new user enters into the system, previous works have attempted to utilize demographics, opinion, mining, and other techniques as mentioned above. However, because of the recent growth in location data on users, spatially aware methods can be employed to intelligently overcome the cold start problem.

One study attempted to utilize spatial information to address the cold start problem in recommendation by capturing the correlations between social networks and geographical distance on LBSNs with a geo-social correlation model. A number of factors were utilized to calculate social correlations on LBSNs consisting of user similarity between local non-friends, location frequency, and user frequency. The study found that their approach properly models the geo-social correlations of a user's cold-start check-ins and significantly improves the location recommendation results [98]. Another study employed spatial elements with a novel method based on fuzzy geographically clustering to solve the Cold-Start problem in recommender systems occurring when a new user enters into the system. This methodology takes into consideration theory of Intuitionistic Fuzzy Sets and Spatial Interaction – Modification Model for demographic and geographic data. Comparative experimentation was performed on benchmark data sets showing that the proposed method obtains better accuracy than the baseline defined in the study [99].

Methodology

Many attempts have been made to incorporate spatial data into recommender systems for solving the issue of a new user entering the system (cold start problem) as described in Section 7.0.2. Yet, the use of the spatial statistics technique, ordinary kriging, to address the issue has not yet been conducted. This subsection will describe the novel methodology for how to employ kriging to aid recommender systems that are traditionally spatially unaware as shown in Figure 7.5.

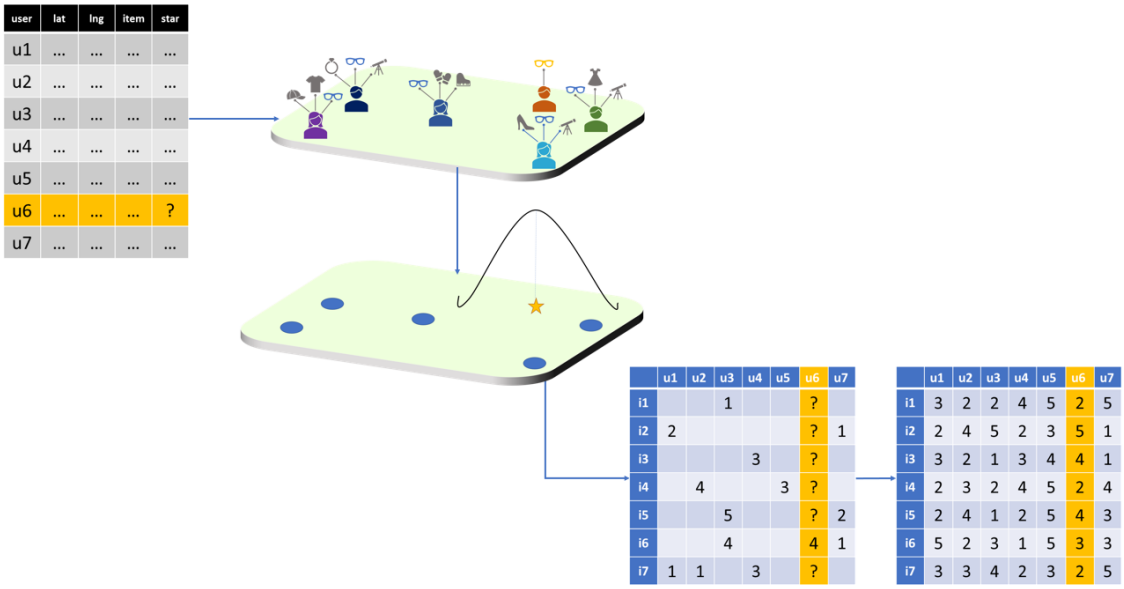


Figure 7.5: Cold Start Methodology Flowchart

When a new user enters into the system and does not have ratings to find hidden patterns, the spatial coordinates of that user can be used in various ways. A novice approach can be to use the inverse distance weight (IDW) which simply assigns closer observed points with more weight and less weight to points further away but an advanced approach would be to use kriging, which is the method of choice in this thesis for addressing the cold start

problem.

The recommender system explained in this section will first determine if a user is new by identifying that the user has no history of rating any items as shown in the gray table in Figure 7.5. The system will then randomly select an item to interpolate ratings since no items have been rated by the new user. Once a random item derived from the system is selected, the system will select the new user and all other users already in the system that have rated the item that is within a certain distance to the new user. Ordinary kriging will then be performed to interpolate a rating value for the item for the new user based on the spatial structure of all other ratings from other nearby users. Once this information is collected, the spatially aware prediction derived from kriging will be used to replace a blank user-item cell in the SVD matrix (denoted with an asterisk in Figure 7.5). At this point, SVD can be performed with the new user since there is now an existing rating, and once completed, ratings for all other items for the new user will be predicted which will allow the system to make recommendations for the new user. If desired, the process described in Figure 7.5 can be run again for other items for the new user so that SVD can run on a denser matrix.

Conclusion

This section has proposed a novel framework for solving the cold-start problem in recommender systems by incorporating spatial elements and spatial statistics. By using the method of kriging, new users that enter the system can be evaluated by understanding how users near them rate an item in the system. By understanding the spatial structure for the new user, a prediction for that user can be made and fed into the SVD matrix to ensure that the user column is not empty which allows SVD factorize the matrix and make new predictions for all other items in the system for the new user. The method described in this thesis can be implemented numerous times in order to create a denser matrix, which could lead to better prediction results.

7.0.3 Applying KARS to User Ratings

Introduction

In this thesis, the integration of kriging and SVD were studied in Section 4 and Section 5 to build an algorithm that makes predictions of spatially correlated data. Later, the integration of the two algorithms was studied in a recommender engine approach. The recommender system KARS was developed in Section 6 as a recommender system for aiding urban planning designers in recommending businesses at specific locations. This spatial recommendation system can also be applied in another way. Instead of recommending businesses at locations, KARS can be used to recommend *users* to *items* based on the location of users. As mentioned in Sections -Related work KDD- there has been extensive research on improving these methods by integrating spatial elements to recommender engines. Although they enrich the recommendation system with simple explicit and task-specific spatial features, KARS goes an additional step ahead by leveraging regression kriging, a state-of-the-art approach to augment spatial statistics. This study will theoretically describe how KARS can be leveraged in a new way; to build a recommender system from the view point of users, making suggestions of items to users based on hidden patterns and spatial autocorrelation as generalized in Figure 7.6.

Most recommender systems use collaborative or content-based filtering techniques allowing the customer to sift through items that may not be of interest and to help them navigate to personalized and relevant items faster. Recommender engines are typically used for e-commerce for companies such as Netflix and Amazon. This theoretical approach can be utilized by companies that want to or already use recommender engines to sell their products more efficiently. Instead of using spatial elements in a naive way, such as inverse distance weighting (IDW) which adds more weights to observed values that are closer to the point being evaluated and less weight on those further away, KARS can calculate the weights of the observed points by calculating their spatial correlation in conjunction with SVD which finds the latent features of the data.

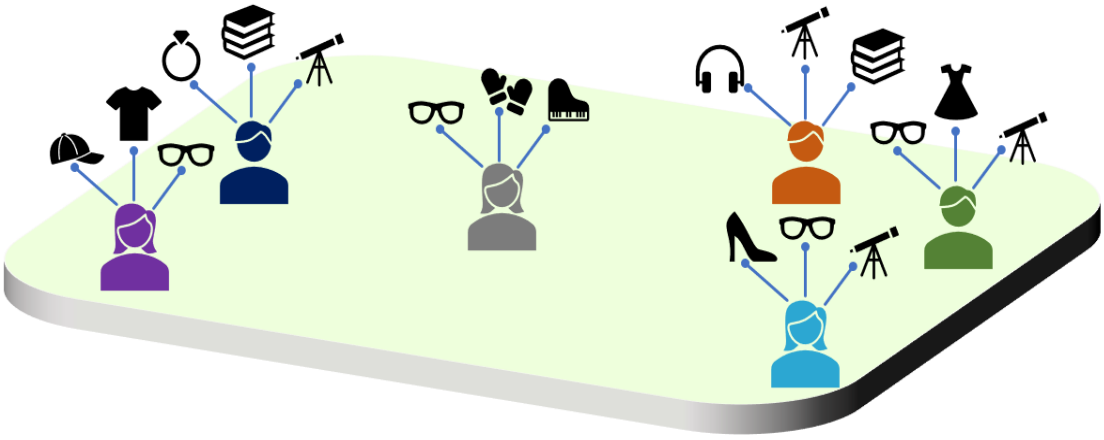


Figure 7.6: User-Item Spatial Recommender Engine

Methodology

Imagine that a cluster of users are purchasing telescopes as shown in Figure 7.6. The recommender system, KARS, can determine if their is spatial autocorrelation for this item. Those results are fed into SVD for further analysis to find the hidden patterns within those spatially-aware predictions. As shown in Table 6.2, this approach improves SVD by 50.6% and improves OK by 10.4%. By leveraging both the kriging and truncated SVD algorithms, KARS can find if users at closer distances with similar interests would find the telescope worth purchasing by calculating their spatial autocorrelation. In contrast, KARS can find if users at greater distances would also find the telescope worth purchasing by calculating the hidden factors using SVD.

In order to execute the recommender system, the data must flow as described in Figure 7.7. The data has a list of user id's with associated latitudes and longitudes, all the items that they have rated, and the ratings the user has given those items. These columns of information are then fed into kriging models. A krigig model is built for every item that is being analyzed for user recommendation. In order to fairly and effectively apply kriging to this type of data set, separate models must be made for each item so that the kriging

algorithm can distinguish the rating for item 1 as different than item 2. Once each model makes a spatial prediction for selected users, the kriging prediction and variance for every user is 1) stored into a dataframe and 2) processed into the SVD algorithm. Unlike kriging, which is required to evaluate user predictions by looking at each item separately in order to make accurate predictions, the SVD matrix has the power to incorporate all items and all users in one shot. Although SVD does not have the spatial awareness that kriging does, it does have the power to evaluate all items and users at once and find hidden patterns between them all in order to make strong item recommendations for the selected users.

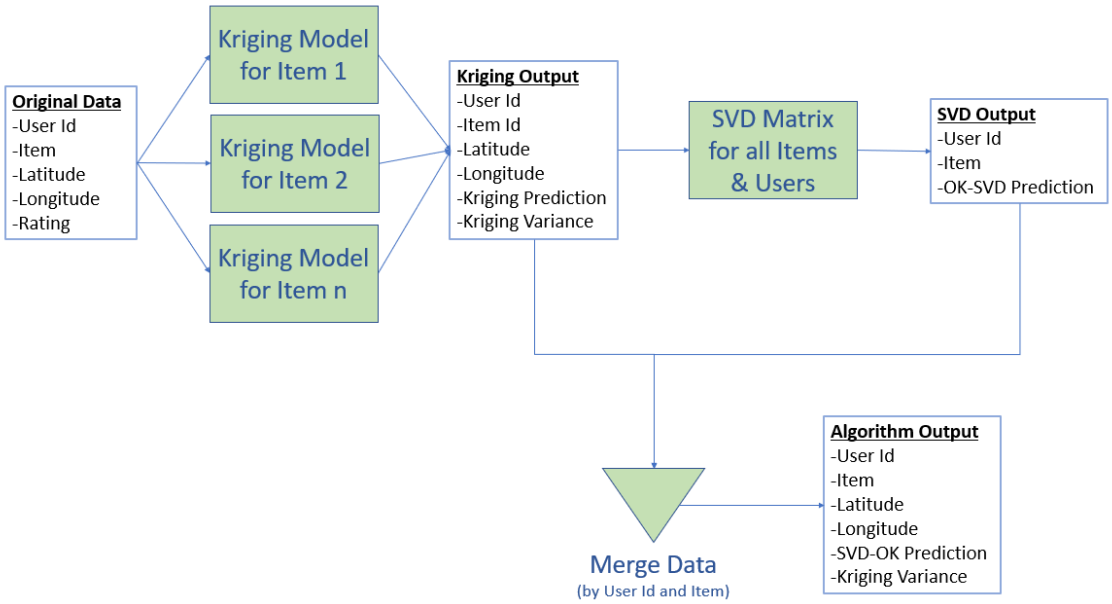


Figure 7.7: Process Map for KARS

Once the kriging process and SVD process is complete, the outputs for each process are stored and merged by user id and item. This connects information from kriging such as latitude, longitude, and variance, to the KARS prediction made by the SVD process. The merged data is than grouped by user so to evaluate each user individually. The KARS

prediction for each user is set in ascending order. Then the variance, which is derived from the kriging process, is utilized to allow the KARS results to be interpretable. This is achieved at the decision point where a variance threshold is set by the system. The smaller the variance, the tighter the fit of the upper and lower bounds for each prediction. This recommender system not only integrates spatial statistics to matrix factorization approaches for recommender engines, but it also makes the predictions more explainable since each predicted rating is associated with a variance which is calculated during the kriging process.

Coding Language and Packages

Both the R and Python programming languages are suggested to be used to execute this study. The python package "surprise" is used to apply truncated SVD to the data. Specifically the SVD function, under the matrix factorization algorithms, within the "surprise" package is used to perform truncated SVD. More information on this technique can be found in [69] and [70]. For this function, Table 7.3 describes the hyperparameters that can be used for this study.

Table 7.3: SVD Hyperparamter Values

Parameter	Value
Latent Factors	4
Epochs	15

The R package “gstat” is used in applying kriging to the data. The “variogram”, “fit.variogram”, and “krige” functions are utilized in making the predictions using the truncated SVD outputs. In order to efficiently run the algorithm to incorporate enough data points for kriging, but not to overwhelm the number of points used to fit the variogram function, the algorithm we structure in a unique way. For every item name being analyzed,

if the number of training points is equal to or less than 65, a max distance is not specified in the krige function. However, if the number of training points is greater than 65, than a max distance is specified to 20 miles. Applying a max distance means only observations within a specified distance from the prediction location are used for prediction or simulation.

Conclusion

Recommender engines have become an integral part of our e-commerce. With the progression of the recommender systems, there have been many improvements to it's implementation. In addition, the availability of big data in today's world and the intensive development of geo-referencing has opened up new ways to employ spatial data, specifically in recommender engines as is described in (Spatially Aware Recommender Systems Section).

The recommender engine that is proposed in this dissertation, KARS, has been tested and evaluated in Section 6. With promising results, this section describes a theoretical approach to implementing KARS in a different light where users are being evaluated by space and latent features so to recommend relevant items that could be of interest to each user. Combined, our approach leverages both the strong predictive power of matrix factorization with the consideration of location and spatial auto-correlation of kriging. Many e-commerce businesses such as Netflix, Amazon, and YouTube can apply the theoretical approach described in this section to strengthen their current systems.

Chapter 8: Final Conclusion

This dissertation is an ensemble of publishable research papers that detail each algorithm developed. In Chapter 4, the research focused on integrating ordinary kriging and SVD by a deep neural network which is referred to in this dissertation as HNN [1]. This algorithm was developed as a proof of concept to determine if there is merit in combining kriging and SVD. In Chapter 5, the two algorithms were further researched, feeding SVD into kriging, which is referred to as SVD-RK. Lastly, in Chapter 6, the recommender system, KARS, was developed where OK predictions were fed into an SVD matrix.

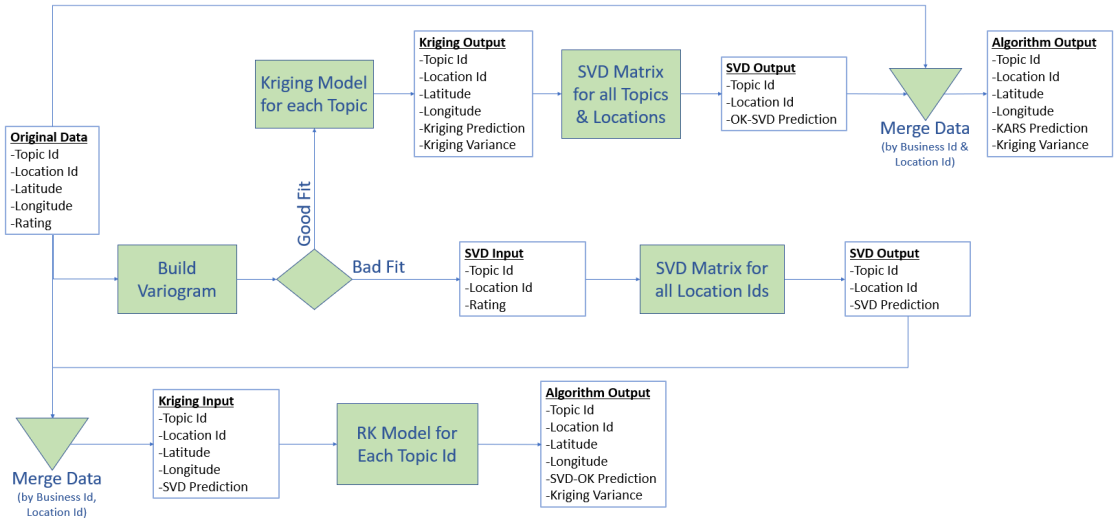


Figure 8.1: Algorithms Workflow

Topic Id is a generic term in this workflow which can be tailored based on the data used. The Topic Id for the Zillow data set is the bedroom-bathroom pair combination and for the Yelp data set is the business id.

While these methods have shown improvement upon kriging and SVD individually, it

is important to know when to use each algorithm. As shown in Chapter 4, the spatial autocorrelation was not as strong in predicting values in comparison to SVD. Thus, when there is minimal spatial autocorrelation with strong latent features, HNN is the preferred algorithm to use. However, for the data sets in Chapter 5 and 6, the empirical testing showed that OK is the strongest baseline in predicting values thus, variograms were created for these data sets to understand the spatial structure further. In this dissertation, it was found that when there is a bad variogram fit and displays a degree of trend, then the SVD-RK algorithm should be used and when the variogram fits the data well, the KARS algorithm should be implemented as shown in Figure 8.1.

For Chapter 5, the workflow in Figure 5.4, shows that if the variogram fits the data well, the strategy was to employ OK, otherwise, if the variogram depicts trend, the SVD-RK algorithm should be utilized. In Chapter 6, we developed KARS and found that instead of using OK when the variogram shows a strong fit to the data we can use KARS to further enhance the prediction power.

Table 8.1: MAE for Algorithms

County	Variogram Fit	OK MAE	UK MAE	GWR MAE	SVD MAE	HNN MAE	SVD-RK MAE	KARS MAE
3101	Good	83,500	158,000	88,200	128,300	$>$ 1,000,000	85,100	**76,400**
1286	Bad	80,600	83,700	83,700	96,800	$>$ 1,000,000	**66,900**	69,500
2061	Bad	59,100	68,500	73,600	74,400	$>$ 1,000,000	**55,800**	55,900

To further ensure that KARS can improve the prediction of the data when the variogram fits the data well better than OK alone, we applied KARS to the Los Angeles county (county id 3101) in the Zillow data set where the fit of the variogram (shown in Figure 5.6a) describes the spatial structure of the data well. We found that KARS performed better than OK and SVD-RK (show in Table 8.1). In contrast, the variogram fit for the other two counties in

the Zillow data set (county id's 1286 and 2061) depicted trend in their variogram thus the KARS algorithm did not perform as well as SVD-RK. These results further conclude that based on the fit of the variogram, we can decide which of the two algorithms, SVD-RK and KARS, can be applied to a data set. In addition, we compared KARS and SVD-RK to the other spatial statistics baselines and truncated SVD baseline to ensure that the ensemble algorithm outperformed the individual approaches shown in Table 8.1.

As a final analysis, all algorithms that were discussed or developed in the dissertation were tested on one data set to analyze their performances (shown in Table 8.2). The data set that was utilized is the Yelp data set that was used in this Chapter 6. Each algorithm uses the same training set to train the algorithm, and the same test set to produce the results.

Table 8.2: Alternative Experiments

Algorithm	RMSE
KARS	**0.6913**
OK	0.7713
GWR	0.8335
HNN	0.8461
UK	0.9625
Baseline	1.3143
SVD	1.4084
SVD-RK	2.8450

Among the three algorithms that were developed in this dissertation, the KARS algorithm was found to produce the strongest results for the Yelp data which was the expected result. Based on the algorithms workflow shown in Figure 8.1, since the variogram fit is good, the KARS algorithm should be utilized. As expected, HNN and SVD-RK did not perform will in comparison to KARS. The three algorithms developed in this dissertation

have the ability to improve recommendation systems that are spatially correlated. By integrating kriging with Simon Funk's SVD, recommender engines can aid in making better predictions users in the traditional recommender engine role, but can also extend to spatial recommendations such as location recommendations.

Bibliography

[1] A. Sikder and A. Züfle, "Emotion predictions in geo-textual data using spatial statistics and recommendation systems," in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-based Recommendations, Geosocial Networks and Geoadvertising*. ACM, 2019, p. 5.

[2] S. Mohammad and P. D. Turney, "Crowdsourcing a word-emotion association lexicon," *Computational Intelligence*, vol. 29, pp. 436–465, 2013.

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[4] J. Bennett, S. Lanning *et al.*, "The netflix prize," in *Proceedings of KDD cup and workshop*, vol. 2007. New York, NY, USA., 2007, p. 35.

[5] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2691–2698.

[6] M. A. Oliver and R. Webster, *Basic steps in geostatistics: the variogram and kriging*. Springer, 2015, vol. 106.

[7] F. P. Lousame and E. Sánchez, "A taxonomy of collaborative-based recommender systems," in *Web Personalization in Intelligent Environments*. Springer, 2009, pp. 81–117.

[8] X. Zhang and H. Wang, "Study on recommender systems for business-to-business electronic commerce," *Communications of the IIMA*, vol. 5, no. 4, p. 8, 2005.

[9] A. Tuzhilin, Y. Koren, J. Bennett, C. Elkan, and D. Lemire, *Large-scale recommender systems and the netflix prize competition*. Association for Computing Machinery, New York, NY, United States, 2008.

[10] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: introduction and challenges," in *Recommender systems handbook*. Springer, 2015, pp. 1–34.

[11] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, aug 2009. [Online]. Available: http://dx.doi.org/10.1109/MC.2009.263

[12] M. Schedl, P. Knees, B. McFee, D. Bogdanov, and M. Kaminskas, "Music recommender systems," in *Recommender systems handbook*. Springer, 2015, pp. 453–492.

[13] I. Guy, "Social recommender systems," in *Recommender systems handbook*. Springer, 2015, pp. 511–543.

[14] N. Lathia, "The anatomy of mobile location-based recommender systems," in *Recommender Systems Handbook*. Springer, 2015, pp. 493–510.

[15] X. Guan, C.-T. Li, and Y. Guan, "Matrix factorization with rating completion: An enhanced svd model for collaborative filtering recommender systems," *IEEE access*, vol. 5, pp. 27 668–27 678, 2017.

[16] S. Zhang, L. Yao, and X. Xu, "Autosvd++: An efficient hybrid collaborative filtering model via contractive auto-encoders," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 957–960.

[17] M. G. Manzato, "Discovering latent factors from movies genres for enhanced recommendation," in *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 2012, pp. 249–252.

[18] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 447–456.

[19] B. B. Trangmar, R. S. Yost, and G. Uehara, "Application of geostatistics to spatial studies of soil properties," in *Advances in agronomy*. Elsevier, 1986, vol. 38, pp. 45–94.

[20] G. C. Sigua and W. H. Hudnall, "Kriging analysis of soil properties," *Journal of Soils and Sediments*, vol. 8, no. 3, p. 193, 2008.

[21] Columbia University Mailman School of Public Health, "Geographically weighted regression. https://www.mailman.columbia.edu/research/population-health-methods/geographically-weighted-regression."

[22] Q. Meng, "Regression kriging versus geographically weighted regression for spatial interpolation," *International journal of advanced remote sensing and GIS*, vol. 3, no. 1, pp. 606–615, 2014.

[23] E. J. Pebesma, "Multivariable geostatistics in s: the gstat package," *Computers & Geosciences*, vol. 30, no. 7, pp. 683–691, 2004.

[24] K. Komnitsas and K. Modis, "Soil risk assessment of as and zn contamination in a coal mining region using geostatisretics," *Science of the Total Environment*, vol. 371, no. 1-3, pp. 190–196, 2006.

[25] H. Zhou, *Computer Modeling for Injection Molding: Simulation, Optimization, and Control*. Wiley, 2012.

[26] Columbia University Mailman School of Public Health, "Population health methods: Kriging. https://www.mailman.columbia.edu/research/population-health-methods/kriging."

[27] "How kriging works," http://desktop.arcgis.com/en/arcmap/10.3/tools/3d-analyst-toolbox/how-kriging-works.htm, March 2019.

[28] B. Matérn, *Spatial variation*. Springer Science & Business Media, 2013, vol. 36.

[29] J. A. Hoeting, R. A. Davis, A. A. Merton, and S. E. Thompson, "Model selection for geostatistical models." *Ecological applications : a publication of the Ecological Society of America*, vol. 16 1, pp. 87–98, 2006.

[30] C. C. Aggarwal *et al.*, *Recommender systems*. Springer, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-29659-3,doi={10.1007/978-3-319-29659-3}

[31] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 1, p. 1, 2010.

[32] W. Ma, J. Shi, and R. Zhao, "Normalizing item-based collaborative filter using context-aware scaled baseline predictor," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[33] S. Whitener, "How Your Emotions Influence Your Decisions," *Forbes*, 2018. [Online]. Available: https://www.forbes.com/sites/forbescoachescouncil/2018/05/09/how-your-emotions-influence-your-decisions/#485ef2b43fda

[34] N. A. Malhotra, A. J. Healy, and C. Mo, "Personal emotions and political decision making: Implications for voter competence," *SSRN Electronic Journal*, 07 2009. [Online]. Available: https://www.gsb.stanford.edu/faculty-research/working-papers/personal-emotions-political-decision-making-implications-voter

[35] W. R. Tobler, "A computer movie simulating urban growth in the detroit region," *Economic geography*, vol. 46, no. sup1, pp. 234–240, 1970.

[36] R. Cellmer, "The possibilities and limitations of geostatistical methods in real estate market analyses," *Real Estate Management and Valuation*, vol. 22, no. 3, pp. 54–62, 2014.

[37] M. Kuntz and M. Helbich, "Geostatistical mapping of real estate prices: an empirical comparison of kriging and cokriging," *International Journal of Geographical Information Science*, vol. 28, no. 9, pp. 1904–1921, 2014.

[38] R. K. Pace, R. Barry, and C. F. Sirmans, "Spatial statistics and real estate," *The Journal of Real Estate Finance and Economics*, vol. 17, no. 1, pp. 5–13, 1998.

[39] K. Wakil, R. Bakhtyar, K. Ali, and K. Alaadin, "Improving web movie recommender system based on emotions," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 2, pp. 218–226, 2015.

[40] S. Deng, D. Wang, X. Li, and G. Xu, "Exploring user emotion in microblogs for music recommendation," *Expert Syst. Appl.*, vol. 42, no. 23, pp. 9284–9293, Dec. 2015. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2015.08.029

[41] Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li, "Idoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization," *Future Generation Computer Systems*, 6 2015.

[42] F. Bohnert, D. F. Schmidt, and I. Zukerman, "Spatial processes for recommender systems," in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[43] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, pp. 129–142, 01 2015.

[44] W. Wang, H. Yin, S. W. Sadiq, L. Chen, M. Xie, and X. Zhou, "Spore: A sequential personalized spatial item recommender system," *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pp. 954–965, 2016.

[45] G. Yang and A. Züfle, "Spatio-temporal site recommendation," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2016, pp. 1173–1178.

[46] S. Klettner, H. Huang, M. Schmidt, and G. Gartner, "Crowdsourcing affective responses to space," *Kartographische Nachrichten*, vol. 63, pp. 66–73, 04 2013.

[47] "Yelp open dataset: An all-purpose dataset for learning," https://www.yelp.com/dataset, Accessed January, 2018.

[48] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *JMLR*, vol. 3, no. Jan, pp. 993–1022, 2003.

[49] D. Ayata, Y. Yaslan, and M. Kamasak, "Emotion based music recommendation system using wearable physiological sensors," *IEEE Transactions on Consumer Electronics*, vol. PP, pp. 1–1, 06 2018.

[50] A. Conner-Simons and R. Gordon, "Detecting emotions with wireless signals," 09 2016.

[51] S. M. Mohammad and P. D. Turney, "Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon," in *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, ser. CAAGET '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 26–34. [Online]. Available: http://dl.acm.org/citation.cfm?id=1860631.1860635

[52] F. S. Tabak and V. Evrim, "Comparison of emotion lexicons," in *2016 HONET-ICT*, Oct 2016, pp. 154–158.

[53] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[54] J. Kim and H. Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3261–3281, 2011.

[55] M. Mayo, "Neural network foundations, explained: Activation function," https://www.kdnuggets.com/2017/09/neural-network-foundations-explained-activation-function.html, March 2019.

[56] N. Kang, "Multi-Layer Neural Networks with Sigmoid Function— Deep Learning for Rookies (2)," *Towards Data Science*, 2017.

[57] D. F. Specht, "A general regression neural network," *IEEE transactions on neural networks*, vol. 2, no. 6, pp. 568–576, 1991.

[58] S. Gibbons and S. Machin, "Valuing school quality, better transport, and lower crime: evidence from house prices," *oxford review of Economic Policy*, vol. 24, no. 1, pp. 99–119, 2008.

[59] D. R. Haurin and D. Brasington, "School quality and real house prices: Inter-and intrametropolitan effects," *Journal of Housing economics*, vol. 5, no. 4, pp. 351–368, 1996.

[60] A. Adair, S. McGreal, A. Smyth, J. Cooper, and T. Ryley, "House prices and accessibility: The testing of relationships within the belfast urban area," *Housing studies*, vol. 15, no. 5, pp. 699–716, 2000.

[61] A. K. Lynch and D. W. Rasmussen, "Measuring the impact of crime on house prices," *Applied Economics*, vol. 33, no. 15, pp. 1981–1989, 2001.

[62] J. V. Duca, J. Muellbauer, and A. Murphy, "Housing markets and the financial crisis of 2007–2009: lessons for the future," *Journal of financial stability*, vol. 6, no. 4, pp. 203–217, 2010.

[63] J. Das, S. Majumder, and P. Gupta, "Spatially aware recommendations using kd trees," *IET Conference Proceedings*, 2013.

[64] J. Li, A. D. Heap, A. Potter, and J. J. Daniell, "Application of machine learning methods to spatial interpolation of environmental variables," *Environmental Modelling & Software*, vol. 26, no. 12, pp. 1647–1659, 2011.

[65] T. Hengl, G. B. Heuvelink, and D. G. Rossiter, "About regression-kriging: From equations to case studies," *Computers & geosciences*, vol. 33, no. 10, pp. 1301–1315, 2007.

[66] J. M. Tadić, V. Ilić, and S. Biraud, "Examination of geostatistical and machine-learning techniques as interpolators in anisotropic atmospheric environments," *Atmospheric Environment*, vol. 111, pp. 28–38, 2015.

[67] F. Veronesi and C. Schillaci, "Comparison between geostatistical and machine learning models as predictors of topsoil organic carbon with a focus on local uncertainty estimation," *Ecological Indicators*, vol. 101, pp. 1032–1044, 2019.

[68] "Welcome to surprise' documentation!" https://surprise.readthedocs.io/en/stable/index.htmlhtml, 2015.

[69] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[70] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273–1284, 2014.

[71] "gstat: Spatial and spatio-temporal geostatistical modelling, prediction and simulation," https://cran.r-project.org/web/packages/gstat/index.html, 2020.

[72] A. Sikder, "svd_kriging," Nov 2019. [Online]. Available: https://github.com/datadj28/svd_kriging

[73] Y. Murayama, *Progress in geospatial analysis*. Springer Science & Business Media, 2012.

[74] T. M. Oshan, Z. Li, W. Kang, L. J. Wolf, and A. S. Fotheringham, "mgwr: A python implementation of multiscale geographically weighted regression for investigating process spatial heterogeneity and scale," *ISPRS International Journal of Geo-Information*, vol. 8, no. 6, p. 269, 2019.

[75] C. Lloyd, "Nonstationary models for exploring and mapping monthly precipitation in the united kingdom," *International Journal of Climatology: A Journal of the Royal Meteorological Society*, vol. 30, no. 3, pp. 390–405, 2010.

[76] P. Harris, A. S. Fotheringham, and S. Juggins, "Robust geographically weighted regression: a technique for quantifying spatial relationships between freshwater acidification critical loads and catchment attributes," *Annals of the Association of American Geographers*, vol. 100, no. 2, pp. 286–306, 2010.

[77] M. M. Fischer and A. Getis, *Handbook of applied spatial analysis: software tools, methods and applications*. Springer Science & Business Media, 2010.

[78] P. Harris, A. Fotheringham, R. Crespo, and M. Charlton, "The use of geographically weighted regression for spatial prediction: an evaluation of models using simulated data sets," *Mathematical Geosciences*, vol. 42, no. 6, pp. 657–680, 2010.

[79] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning geographical preferences for point-of-interest recommendation," in *ACM SIGKDD*, 2013, pp. 1043–1051.

[80] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen, "Lcars: a location-content-aware recommender system," in *ACM SIGKDD*, 2013, pp. 221–229.

[81] M. Sarwat, J. J. Levandoski, A. Eldawy, and M. F. Mokbel, "Lars*: An efficient and scalable location-aware recommender system," *TKDE*, vol. 26, no. 6, pp. 1384–1399, 2014.

[82] W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, and X. Zhou, "Geo-sage: A geographical sparse additive generative model for spatial item recommendation," in *ACM SIGKDD*, 2015, pp. 1255–1264.

[83] M. Beyard, A. Kramer, B. Leonard, M. PAWLUKIEVICZ, D. Schwanke, and N. Yoo, "Ten principles for developing successful town centers," *Urban Land Institute. Washington*, p. 30, 2007.

[84] "The quiet revolution happening in the suburbs," https://www.governing.com/topics/urban/gov-suburban-urban-changes.html, 2017.

[85] R. Keil, "Welcome to the suburban revolution," *Suburban Constellations*, pp. 8–15, 2013.

[86] L. C. Johnson, "Style wars: revolution in the suburbs?" *Australian Geographer*, vol. 37, no. 2, pp. 259–277, 2006.

[87] M. Birkin, G. Clarke, and M. Clarke, "Gis for business and service planning," *Geographical Information Systems*, vol. 2, pp. 709–722, 1999.

[88] B. Harris and M. Batty, "Locational models, geographic information and planning support systems," *Journal of Planning Education and Research*, vol. 12, no. 3, pp. 184–198, 1993.

[89] M. Birkin, G. Clarke, M. Clarke, and R. Culf, "Using spatial models to solve difficult retail location problems," *Applied GIS and spatial analysis*, pp. 35–56, 2004.

[90] J. Lin, R. J. Oentaryo, E.-P. Lim, C. Vu, A. Vu, A. T. Kwee, and P. K. Prasetyo, "A business zone recommender system based on facebook and urban planning data," in *European Conference on Information Retrieval*. Springer, 2016, pp. 641–647.

[91] B. Resch, A. Summa, P. Zeile, and M. Strube, "Citizen-centric urban planning through extracting emotion information from twitter in an interdisciplinary space-time-linguistics algorithm," *Urban Planning*, vol. 1, no. 2, pp. 114–127, 2016.

[92] N. Ferreira, M. Lage, H. Doraiswamy, H. Vo, L. Wilson, H. Werner, M. Park, and C. Silva, "Urbane: A 3d framework to support data driven decision making in urban development," in *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2015, pp. 97–104.

[93] J. Lee, M. Sun, and G. Lebanon, "A comparative study of collaborative filtering algorithms," *arXiv preprint arXiv:1205.3193*, 2012.

[94] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl, "Movielens unplugged: experiences with an occasionally connected recommender system," in *Proceedings of the 8th international conference on Intelligent user interfaces*, 2003, pp. 263–266.

[95] L. Safoury and A. Salah, "Exploiting user demographic attributes for solving cold-start problem in recommender system," *Lecture Notes on Software Engineering*, vol. 1, no. 3, pp. 303–307, 2013.

[96] D. Poirier, F. Fessant, and I. Tellier, "Reducing the cold-start problem in content recommendation through opinion classification," in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1. IEEE, 2010, pp. 204–207.

[97] H. Sobhanam and A. Mariappan, "Addressing cold start problem in recommender systems using association rules and clustering technique," in *2013 International Conference on Computer Communication and Informatics*. IEEE, 2013, pp. 1–5.

[98] H. Gao, J. Tang, and H. Liu, "Addressing the cold-start problem in location recommendation using geo-social correlations," *Data Mining and Knowledge Discovery*, vol. 29, no. 2, pp. 299–323, 2015.

[99] K. M. Cuong, N. T. H. Minh, N. Van Canh *et al.*, "An application of fuzzy geographically clustering for solving the cold-start problem in recommender systems," in *2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR)*. IEEE, 2013, pp. 44–49.

Curriculum Vitae

Aisha Sikder is a PhD. student in the Earth Systems and Geoinformation Sciences program at George Mason University. She received a Bachelors of Science in Civil Engineering from Geroge Mason University in May of 2012, and graduated from George Mason University with a Master's of Science in Systems Engineering in May 2015. Aisha has worked with the Intelligence Community and Federal Agencies since 2014 and has pursued a career in Systems Engineering and Data Science and currently works with Perspecta Engineering since 2016.

Technical Skills

- Python & R
- Tableau
- Neo4j NoSQL Database
- MongoDB NoSQL Database
- SQL/Oracle
- Selenium
- RegEx/NLP
- Informatica Power Center (ETL)
- Alteryx