8-2020

Development and Identification of Metrics to Predict the Impact of Dimension Reduction Techniques on Classical Machine Learning Algorithms for Still Highway Images

Wasim Akram Khan
Utah State University

DEVELOPMENT AND IDENTIFICATION OF METRICS TO PREDICT THE IMPACT OF DIMENSION REDUCTION TECHNIQUES ON CLASSICAL MACHINE LEARNING ALGORITHMS FOR STILL HIGHWAY IMAGES

by

Wasim Akram Khan

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

of

MASTER OF SCIENCE

in

Computer Science

Approved:

Douglas Galarus, Ph.D. Major Professor	John Edwards, Ph.D. Committee Member
Nicholas Flann, Ph.D. Committee Member	Janis L. Boettinger, Ph.D. Acting Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2020

ABSTRACT

Development and identification of metrics to predict the impact of dimension reduction techniques on classical machine learning algorithms for still highway images

by

Wasim Akram Khan, Master of Science

Utah State University, 2020

Major Professor: Douglas Galarus, Ph.D.
Department: Computer Science

The US Transportation System is heavily monitored by cameras and other sensors, generating large amounts of data which can be analyzed to extract actionable insights to improve transportation. Closed-Circuit TV cameras (CCTVs) monitor the highway and produce high dimensional images which make it challenging to apply machine learning algorithms. Dimension reduction techniques can help in dealing with a large amount of high dimensional data. Faster training and inference time on dimension-reduced data and smaller models which can be deployed on commodity hardware are a critical advantage of dimension reduction. This thesis explores the impact of dimension reduction on the performance of classical machine learning algorithms, and identifies and devises measures that best predict the impact. The dataset used is a time series of images from several camera feeds observing the traffic, weather and road conditions along the highways. Readings from other sensors were used to label images to facilitate the application of supervised machine learning algorithms.

(97 pages)

PUBLIC ABSTRACT

Development and identification of metrics to predict the impact of dimension reduction techniques on classical machine learning algorithms for still highway images

Wasim Akram Khan

We are witnessing an influx of data - images, texts, video, etc. Their high dimensionality and large volume make it challenging to apply machine learning to obtain actionable insight. This thesis explores several aspects pertaining to dimensional reduction: dimension reduction methods, metrics to measure distortion, image preprocessing, etc. Faster training and inference time on reduced data and smaller models which can be deployed on commodity hardware are a critical advantage of dimension reduction. For this study, classical machine learning methods were explored owing to their solid mathematical foundation and interpretability.

The dataset used is a time series of images from several camera feeds observing the traffic, weather and road conditions along highways. The time-series nature of dataset gives rise to interesting questions which are investigated in this work. For instance, can machine learning models trained on past data be used on future camera feed data? This is highly desirable and yet difficult due to the changing weather, road conditions, traffic conditions and scenery. Can dimension reduction models obtained from past data be used for reducing dimensionality of future data? This thesis also examines the difference between the performance of machine learning methods before and after application of dimension reduction. It tests some existing metrics to measure quality of dimension-reduced data set and introduces several new ones. It also examines the application of image pre-processing methods to boost the performance of classifiers. The classification performance with and without random sampling has been studied as well.

ACKNOWLEDGMENTS

It would be dishonest to claim this thesis as an individual effort. I would like to take this opportunity to thank everyone without whom this thesis would not have been realized. I have had the love and support of numerous people over the past two years and some even longer.

Firstly, I would like to express my deepest gratitude to my advisor Dr. Douglas Galarus for his invaluable contribution to this work. His faith in me, the guidance and expertise that he provided me through our meetings shaped the research and ultimately this thesis. I am deeply indebted to Dr. John Edwards for his ingenious suggestions and appropriate comments which helped to further clarify this thesis. I have had thought-provoking discussions with Dr. Flann for which I am eternally grateful.

I am grateful to the wonderful faculty and staff of the Computer Science department. I would like to acknowledge the assistance of Cora Price, Genie Hanson, Kaitlyn Fjeldsted, and Vicki Anderson for handling all the administrative requirements of my degrees allowing me to focus on my research and study.

I am blessed to have a wonderful family. They made me the person that I am today. Life without their unwavering love and support is just unimaginable. My labmates were every bit fun making my life lively and also helpful in shaping the thesis through meaningful discussions and supporting me on and off the research.

Wasim Akram Khan

CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

ACRONYMS

CCTV	Closed Circuit Television
DR	Dimension Reduction
HSV	Hue Saturation Variance
kNN	k-Nearest Neighbors
ML	Machine Learning
RGB	Red Green Blue
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

United States highways are monitored to improve and advance the state of transportation. Doing so generates a large amount of data - still CCTV images, live videos, and readings from numerous other sensors observing weather, road surfaces, and a multitude of other factors that impact transportation - in the range of terabytes or more. In this thesis, we examine still images generated by roadside CCTV cameras and use readings of weather and other sensors to label the images. Images convey a lot of information like the status of the roads, the weather in and around the place under observation, etc. Images are a classic example of high dimensional data; high dimensionality is a curse. It makes it computationally expensive to apply machine learning algorithms and other algorithms to extract useful insights from the data.

In this thesis, we reduce data sets into lower dimensions using dimension reduction techniques and then train classical machine learning algorithms on the reduced datasets. The performance of a classifier on reduced data may be better or worse than the classifier trained on the original, high-dimensional dataset depending on the reduction method used, the classifier type, and the size of the reduced data set. Classifier performance can be directly measured using recall, precision, accuracy, and f1 score versus the size of the reduced data set. Classifier performance can also be estimated by using proxy metrics measuring the preservation of the relative structure of the data set before and after dimension reduction.

Only a handful of classical machine learning classifiers are studied in this thesis. Studying all of them would not be feasible. Convolution Neural Networks which are very famous for dealing with images is not studied in this thesis. It is because this thesis does not aim at building successful image classifiers, rather it intends to develop and/or identify metrics to predict the impact of dimension reduction techniques on classical machine learning algorithms.

1.1 Background

Even though the capacity of storage mediums is increasing and costs are falling rapidly, it would be ideal to not have to store forever the images obtained from CCTV cameras monitoring roads. There also might be legal reasons to not store images beyond a certain duration for privacy or other reasons.

As we have the images and sensor readings for any particular location, we can merge both of them to label the images. It is possible to automatically generate road warnings or guidance using classifiers trained on the resulting labeled dataset. Since the data can be huge we are interested in training classifiers on projected (dimension reduced) dataset. The challenge, however, is preserving the optimal number of components in the projected data. If the dimension of the projected data is too low then the classifiers' performance will suffer and if it is too high then we waste compute resources for a minor reduction in the dimension of the dataset. The performance of classifier and the quality of a reduced dataset vary according to the method used to reduce dimension.

In this study, we measure the impact of dimension reduction on the classifiers' performance both directly using metrics based on a confusion matrix and indirectly using multiple metrics for preservation of local structure of the data set.

1.2 Related Works

There are a few related works, which we used for guidance and understanding. Maaten et. al [1] focus on the comparison of non-linear dimension reduction techniques and not on developing metrics that measure preservation of the underlying structure after dimension reduction. They use data sets that are artificial and thus do not necessarily tell us how the metrics fare on natural datasets. They also used natural datasets whose dimension is in the range of 1000 which is much less than the dimensionality of our dataset which is 506,880. The dimensions of their artificial datasets are also low compared to the datasets used in this thesis. Gracia et. al [2] also use data sets similar to those mentioned above in their experiments to compare several linear and non-linear dimension reduction methods using preservation of geometry as the objective function for measuring loss of quality. Neither [1]

nor [2] study time-series data.

The paper by Wang et. al [3] also studies the role of dimension reduction in classification in a qualitative manner. It studies the optimization of classification performance over dimension reduced dataset. It does not introduce any metric and focuses more on the runtime of classifiers on reduced data sets.

Keller et. al [4] study the impact of dimensionality reduction and feature selection on the classification performance but it focuses entirely on hyperspectral EnMAP data. They use the following metrics to measure the performance of the classifers: overall accuracy, Cohen's kappa coefficient, average completeness, average correctness, and average quality. It does not measure the quality of projection and is hyper-focused on a particular type of remote sensing dataset. In this thesis, we focus on a transportation dataset and also measure the quality of the dimension-reduced dataset.

The paper by Jelena et. al [5] compares different dimension reduction methods based on computation speed and accuracy. The accuracy measures are Stress, Spearman coefficient, and Shannon entropy. It uses three different groups of data- randomly generated clustered data, randomly generated nonclustered data and financial data. It does not deal with transportation images or time-series data and focuses on visualization. Nikkila et al. [6] also use metrics that measure topology preservation after dimension reduction on gene expression data to compare Self Organizing Map with Multidimensional Scaling and Hierarchical clustering. They study impact on visualization due to dimension reduction.

Obaid et. al [7] specifically report only time saving and classifiers' accuracy resulting from the usage of several pre-processing and dimension reduction techniques. In this thesis, we estimate the preservation of the structure of the data after dimension reduction as well as classifier performance on it. We do not examine computation time.

None of the related works above experiment on time-series data while our work is specifically on time-series data.

1.2.1 Metrics for Measurement of Quality of Projection

We opted for preservation of structure as the objective for projection quality of dimension reduction methods. There are many metrics to measure preservation of structure. Not all of them are general purpose and applicable to every method of dimension reduction. Many of them are very specific to the type of projection method used and the type of dataset on which reduction was carried out. In this thesis, we use some of the existing metrics to measure the projection quality and introduce several measures of our own. The metrics to measure preservation of structure are described in Chapter 3 'Dimension Reduction.'

CHAPTER 2

DATASET

This thesis is based on the exploration of novel datasets acquired from still CCTV cameras located on roadsides. Series of labeled images belonging to the same location form a dataset. There are datasets belonging to various locations. The availability of datasets from multiple locations helped us to test the generalization of algorithms and metrics from one site to another.

2.1 Acquisition of Image Dataset

The dataset consists of images from several locations in California whose roads/highways are monitored by CCTV cameras. There are multiple sensors placed along the roads to monitor the state of the roads. Weather is monitored and the status of the road surface is observed and recorded. The images and all the other information from the sensors are time-stamped.

2.2 Acquisition of Weather and Road Conditions Data

Thousands of files containing time-stamped sensor data for the locations monitored by the CCTV cameras were gathered from the California Department of Transportation (Caltrans) sensors. The files were collated into a single CSV (comma separated values) file to act as a central database. This allowed us to refer to just one file while labeling images. The records/rows were sorted according to locations and for each location, they were sorted in increasing order of the timestamps for labelling. An illustrative sample of the collated database is shown in Table 2.1.

recordDate	recordTime	essPrecipYesNo	essSurfaceStatus.1	locationName
3/7/2019	16:38:21	2	4	Spring Garden
1/25/2019	21:38:03	2	4	Spring Garden
1/27/2019	22:38:01	2	4	Spring Garden
12/4/2018	0:07:57	2	4	Spring Garden
11/8/2018	3:08:08	2	3	Spring Garden

Table 2.1: Sample weather.csv showing records of the Spring Garden site after collating weather and other sensor data from multiple files. Note that some columns were removed for display purposes.

2.3 Labeling Images

The central database has multiple attributes to explore. However, some attributes have more missing cells/records than others. The attributes which have been explored are precipitation (essPrecipYesNo), precipitation1hr (essPrecipitationOneHour), surfaceStatus (essSurfaceStatus.1), recordDate, and recordTime. The attribute recordDate and recordTime are not directly explored. They are combined to form a binary attribute named "daynight" which tells us whether it is day or night at the time of reading sensors. All of the attributes are detailed here [8]. One of the classification problems we subsequently address is that of determining day/night status. Thus, classification algorithms are trained individually on one of these four attributes.

The process of labeling images from a particular location involves the following step:

- Get the latitude and longitude of the location.
- For each image:
  - Read date and time stamp for the image. Compensate for the daylight savings and timezone differences.
  - Obtain the sunset and sunrise time.
  - Find the record containing the closest timestamp in the weather.csv file. Using binary search on pre-sorted timestamps would give optimal performance.
  - Read the values for the required attributes: precipitation, precipitation1hr, surfaceStatus.

It is important that the images are correctly labeled. To that end, a substantial portion of the images for one location was manually verified to confirm the correctness of the data collating and labeling scripts.

2.4 Properties of the given dataset

A (perfectly) balanced dataset for an attribute is a dataset containing an equal number of samples for each class of the attribute. For instance, for the attribute 'precipitation', the classes are 'Yes' and 'No' indicating precipitation and lack of precipitation respectively. This dataset would be balanced for the attribute 'precipitation' if the dataset contains an equal number of images having precipitation and lacking precipitation. (It does not).

A balanced dataset is considered ideal because while sampling to form training and testing sets both the sets would have equal or nearly equal samples of each class. This, in turn, implies that the classification algorithm is trained on each class equally; it compensates for errors while classifying all classes equally and results in performance metrics which make more sense in general. Our datasets have no such advantage. For instance, California, like almost any other place, witnesses fewer times having precipitation than without precipitation. It implies that there will be more images without precipitation than images with precipitation. This makes it harder for classifiers to learn what exactly constitutes precipitation in an image, as most of the images it sees are examples of lack of precipitation.

A few properties of the dataset have been enumerated below to clarify the nature of the dataset at disposal.

- Unbalanced.
- High dimensional

A dataset based on the Spring Garden site is the most used dataset for experiments in this thesis. Originally, the image capture-date ranged from 05/31/2018 to 02/12/2019,

which is dominated by summer which witnesses little to no precipitation causing the attributes to be highly unbalanced. To make the attributes more balanced, we reduced the dataset to use images captured from 01/01/2019 to 02/12/2019.

2.5 MNIST Dataset

We also use MNIST dataset to measure performance of classifier on dimension-reduced data set, and quality of projection of dimension reduction techniques. The results on MNIST dataset are used as general guidelines and not as ground truth to be achieved on our dataset. MNIST and our dataset have the following differences:

- Our dataset is time-series dataset while samples in MNIST are not time-related.
- The dimensionality of MNIST image is 28 * 28 while the dimensionality of an image in our dataset which is originally in RGB color space is 704 * 240 * 3.
- MNIST is a balanced dataset with roughly equal samples for each of the ten classes. Our dataset is highly unbalanced.

2.6 Motivation

The primary objective of this thesis is to study the impact of dimension reduction on classifiers. However, the temporal nature of the dataset opens up interesting avenues to explore. For instance, does a classifier trained to predict precipitation in images on last year's dataset work for images from the current year? Do reduction parameters extracted from the previous dataset preserve the local structure of the next batch of sequential data?

The purpose of this study is not to study the many different ways to mitigate the effects of unbalanced datasets. We explored classification performance on dimension reduced dataset for some of the attributes that were less unbalanced compared to others.

CHAPTER 3
DIMENSION REDUCTION

3.1 Introduction

This thesis primarily deals with still images obtained from roadside CCTV cameras. An image is basically made up of dots called pixels. For a grayscale image, each dot is just one value (generally 0-255) representing color intensity. The product of weight and height is the dimension of the image. For RGB color space, each pixel is made up of a combination of these three colors (Red, Green, Blue). The product of width, height, and the number of channels (3) is the dimension of the image in RGB color space. An image in a data set is called a data point or, simply, a point. The dimension-reduced image in the low-dimensional space is also called a data point.

A data point is described by several variables or parameters. The number of variables is the dimension of the data point. The primary basis of dimension reduction is that all of the variables that describe a data point may not be actually required to describe the data point. We call the number of parameters or variables required to describe a data point without any loss of information as the intrinsic dimension. The dimension of a data point can also be traded off for faster computation with minimum loss of information.

Dimension reduction (DR) is the process of reducing the number of variables that describe a point in a vector space using statistical methods. DR uses both linear and non-linear transformations to determine the intrinsic dimension of a dataset. In this thesis, we explore PCA (Principal Component Analysis) and Random Projection which use linear transformation and Image Scaling which may use either linear or non-linear transformation.

3.2 The case for Dimension Reduction

We are overwhelmed with data in all forms- images, text, video, and combinations of

these. Images, videos, text, etc. are examples of high-dimensional data. The high dimensionality of data is a curse. It makes visualizing the data difficult making it challenging to gain insights into the data. It also makes it computationally expensive to apply machine learning or other algorithms to extract insights from the data. Dimension reduction techniques are helpful in dealing with a large amount of high dimensional data.

Some of the benefits of dimension reduction are:

- The reduced dimensionality of the dataset results in faster training and inference times, which is a critical advantage of dimension reduction.
- It reduces the size of machine learning models, which, in turn, enables them to be run on less powerful, commodity hardware. It enables wide on-machine deployment of machine learning on commodity hardware.
- Wang et. al [3] state that DR methods can have a regularizing effect which helps to avoid overfitting. They further argue that DR can remove two types of noise from the input: (1) independent random noise, which is uncorrelated with the input and the label, and mostly perturbs points away from the data manifold. (2) Unwanted degrees of freedom, which are possibly nonlinear, along which the input changes but the (class) label does not.
- The data projected into lower dimension requires less storage capacity; storage of data, however, is a smaller concern due to the falling prices of storage mediums.

3.3 Dimension Reduction Methods

There are many dimension reduction methods. We explored Image Scaling which is specific to images, Random Projection, and Principal Components Analysis (PCA) in this thesis. These are described in more detail below.

3.3.1 Image Scaling

Image Scaling refers to the change (increment or reduction) of an image's height and width with (or without) its aspect ratio kept intact. For the purpose of this study, only a reduction in image height and width with aspect ratio kept intact is considered as Image Scaling. Increasing the size of the image does not lead to a reduction in the number of parameters that describe an image. Image Scaling is one of the obvious ways of reducing dimensionality of an image and equally ignored. While it is obvious that reducing the size of an image causes loss of detail, it is also obvious that the scaled-down image preserves features of the original image. Nearest Neighbor Interpolation [9] was used to scale down images in this thesis. It provides a good trade-off between computational complexity and quality of scaled image.

An advantage of Image Scaling is that it is intuitive. A drawback is that it treats all samples in the dataset individually. It fails to leverage the relation that might be present between two data points or the dataset as a whole.

3.3.2 Principal Components Analysis (PCA)

Invented in 1901 by Karl Pearson [10], PCA is one of the most used dimension reduction methods owing to its simplicity.

PCA can be intuitively thought of as trying to find new basis axes that are linear combinations of the original axes. The new basis axes are called principal components. The first basis axis explains the highest variability in the data and the variability explained decreases with each succeeding basis axis. The variability of a dataset explained by the principal components is called explained variance. The last few basis axes may hopefully explain little to no variability of the data set and are redundant leading to dimension reduction. The orthogonal transformation ensures that all of the principal components are orthogonal to each other. PCA assumes that the data set has Gaussian distribution [11].

Unlike Image Scaling and Random Projection, PCA exploits relationships among data points to project data into a lower dimension. The paper by Shlens [11] explains the concept of PCA with a rare combination of mathematical rigor and intuitiveness. Bhagoji et. al [12]

demonstrate the usage of PCA to build machine learning systems resilient against evasion attacks.

3.3.3 Random Projection

Random Projection is a simple and computationally fast method of dimension reduction. It is based on the Johnson-Lindenstrauss Lemma [13] which states that the pairwise distances can be almost preserved for a dataset of sufficiently high dimension when projected into a suitable lower dimension.

It also gives the lower bound on the number of components required to preserve the pairwise distances given the original dimension d. Given that we are willing to tolerate relative error ϵ, the number of components required to preserve pairwise distances is given by the equation 3.1

$$k = \lceil 4(\epsilon^2/2 - \epsilon^3/3)^{-1} log(d) \rceil \tag{3.1}$$

To project a d-dimensional data into k-dimensional data, we use a random projection matrix of dimension k*d where k << d.

$$X^{RP}_{k*N} = R_{k*d} X_{d*N} \tag{3.2}$$

where X_{d*N} is the original d-dimensional data, R_{k*d} is the randomly-generated projection matrix, X^{RP}_{k*N} is the projected k-dimensional data.

There are a few different ways to compute the random projection matrix. For this thesis, we used Sparse Random Projection [14] which is described by the following equation:

$$R_{i,j} = \begin{cases} 0, & \text{with probability } 1 - 1/s \\ \sqrt{\frac{s}{n\_component}}, & \text{with probability } 1/2s \\ -\sqrt{\frac{s}{n\_component}}, & \text{with probability } 1/2s \end{cases} \tag{3.3}$$

where s = 1 / density and n_component is the size of the projected subspace. Density

is the ratio of non-zero component in the random projection matrix to the total number of components in the matrix. The range of density is [0, 1]. We computed density as $1/\sqrt{n\_features}$ as recommended by Li et al [15]. n_features is the number of variables that define a data point or the dimensionality of the data point.

Random Projection only requires as input the number of components to preserve to generate the random projection matrix. If the random state of the random projection matrix generator is fixed, the same projection matrix is obtained every single time which is helpful in processing batches of data. In other words, it is easy to ensure that each batch of data will undergo the same random linear transformation while projecting them into lower dimensions.

3.4 Ranking Data Points

To measure the quality of projection, we also have to introduce metrics to measure the distance between data points. The following metrics described below were used to measure the distance between two data points:

3.4.1 Euclidean Distance

Euclidean distance between two N dimensional data points p and q is defined as the following:

$$d(p,q) = d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i^2)} \tag{3.4}$$

It is simple, widely used, and easy to understand.

3.4.2 Cosine

Cosine is a well-known metric to measure similarity of two data points. The distance between two points u, v is given by 1 - cos(u, v). The cosine of the angle between two

vectors $u, v \epsilon D$ is defined as the following

$$cos(u, v) = \frac{u.v}{|u||v|} \tag{3.5}$$

Given a transformation function f, two points in projected space would be f(u) and f(v). The cosine distance between the two points would be $\frac{f(u).f(v)}{|f(u)||f(v)|}$. This assumes that the origin is preserved during projection which may not necessarily be the case. Let's take a reference point w which can be safely assumed to be the origin in high-dimensional space and f(w) be the projected origin in low-dimensional space. So, the cosine distance in projected space would be the following:

$$cos(f(u), f(v)) = \frac{(f(u) - f(w)).(f(v) - f(w))}{|(f(u) - f(w))||(f(v) - f(w))|} \tag{3.6}$$

Cosine distance is widely used in Natural Language Processing [16]. Computing Euclidean distance is computationally cheaper than computing cosine distance for images. Both the distances can be used to compute k-Nearest Neighbors. The impact of the choice of distance metric is detailed in later sections.

3.5 Measurement of quality of dimension reduction methods

The quality of dimension reduction methods can be measured in different ways- the quality of point reconstructed from projected point, the preservation of local structure of the data, etc. In this paper, we measure the preservation of structure of the data as a measure of the quality of dimension reduction.

Maaten et. al [1] argue that measuring preservation of local structure is more important than measuring reconstruction error because for successful visualization or classification of data, its local structure needs to be preserved. An evaluation of the quality based on generalization errors, trustworthiness, and continuity has an important advantage over measuring reconstruction errors because a high reconstruction error does not necessarily imply that the dimensionality reduction technique performed poorly.

3.5.1 k-Nearest Neighbors (kNN)

The k-Nearest Neighbors algorithm is used to find k closest data points for all the data points in the dataset using a defined metric for distance. It is not a metric to measure quality of projection in itself but can be used with other metrics.

In this algorithm, for each point, the distance with every other point in the dataset is computed. Then, the points are sorted according to their distance. The first k neighbors are then considered as the k-nearest neighbors. It is an unsupervised algorithm since it does not require the dataset to be labeled.

This algorithm is useful to measure the impact of dimension reduction on the preservation of local structure of the data. First, the k nearest neighbors of each data point in the dataset are computed in the original high-dimensional space. Then, the k nearest neighbors of each point in the dataset are computed in the low-dimensional space. Then, several metrics described below were computed using the ranking of neighbors for each point in high-dimensional space and low-dimensional space.

3.5.2 Cosine Deviation

Cosine Deviation was developed in this thesis to measure the preservation of angles between data points after dimension reduction. Mathematically, to measure if the angles are preserved by the transformation function f, we compute the following:

$$CosineDeviation(u, v) = |cos(u, v) - cos(f(u), f(v))| \tag{3.7}$$

Computing cosines between data points has been detailed in the equation 3.6. When there is no preservation of cosine distance after dimension reduction, the Cosine Deviation of two data points u, v is 2, and ideal preservation results in a value of 0.

3.5.3 Trustworthiness

Trustworthiness [17] measures the proportion of points that are close together in the low-dimensional space but not in the high-dimensional space. It is defined as the following:

$$T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \epsilon U_i^{(k)}} (r(i,j) - k) \tag{3.8}$$

where n is the number of data points

k is the number of nearest neighbors to consider

$r(i,j)$ represents the rank of the data point j according to the pairwise distances between the high-dimensional data points.

The variable U_i^k indicates the set of points that are among the k nearest neighbors in the low-dimensional space but not in the high-dimensional space.

The value of trustworthiness ranges from 0 to 1. The ideal value of 1 is achieved when the k neighbors in the high and low-dimensional space are the same. One of the drawbacks of trustworthiness measure is that it is hard to tell when the measure evaluates to 0.

3.5.4 Continuity

Continuity [17] measures the proportion of points that are close together in the high-dimensional space but not in the low-dimensional space. Continuity [17] is defined using the equation below:

$$C(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \epsilon V_i^{(k)}} (r(i,j) - k) \tag{3.9}$$

where n is the number of data points

k is the number of nearest neighbors to consider

$r(i,j)$ represents the rank of the data point j according to the pairwise distances between the low-dimensional data points.

The variable V_i^k indicates the set of points that are among the k nearest neighbors in the

high-dimensional space but not in the low-dimensional space.

The value of continuity ranges from 0 to 1. The ideal value of 1 is achieved when the k neighbors in the high and low-dimensional space are the same. It has the same drawback as trustworthiness. It is hard to analyze when the metric evaluates to 0.

Trustworthiness and continuity are complements of each other. Given two clusters A and B, if you assume trustworthiness measures preservation of structure in B from A's perspective, then continuity measures the preservation of structure in A from B's perspective.

3.5.5 Ratio Preserved

Ratio Preserved is one of the new metrics developed in this thesis. It is defined as the number of neighbors common in both the original, high-dimensional space and the reduced, low-dimensional space divided by k (total number of neighbors considered for measurement of structure preservation).

$$RatioPreserved(i) = \frac{|H_i \bigcap L_i|}{k} \tag{3.10}$$

where H_i is the neighbors of i in higher dimension L_i is the neighbors of i in the lower dimension

Ratio Preserved ranges from 0 to 1. A value of 0 means that the neighbors of a data point in high dimension and low dimension space do not have any point in common. A value of 1 implies that a point has the same neighbors in both the high-dimensional space and the projected low-dimensional space.

Ratio Preserved is easy to compute and interpret. One drawback of the metric is that it does not take the sample size into consideration. Also, as k increases from 0 to N (number of samples), the value of Ratio Preserved rises for any arbitrary dataset and choice of dimension reduction method. However, it is very useful when the number of neighbors we are considering is small compared to the dataset size i.e. k << N.

This metric stems from the idea that we are only concerned with how a dimension reduction method alters the closest neighbors of each data point. Only the closest k neighbors influence the classification of a data point in the kNN Classifier algorithm. It is denoted as ratio_preserved_k in the results and figures where k is the number of neighbors considered.

The paper by Maaten et. al [1] suggests using one Nearest Neighbor (1NN) classifier to estimate the preservation of local structure after dimension reduction. We note that 1NN Classifier may not be a good measure because it implies that preserving the nearest neighbor results in good classification or vice-versa which may not be the case. So we propose computing Ratio Preserved with k = 1 as a better alternative because of the following reasons:

- Ratio Preserved only cares about estimating the preservation of structure not that if preserving the nearest neighbor leads to correct classification.
- It is possible that preserving the nearest neighbor does not lead to correct classification of the data point being classified leading to false and lower estimate of preservation of structure.

3.5.6 Distortion Factor

The Johnson-Lindenstrauss lemma states that given a dataset of sufficiently high dimensions, the pairwise distance can be preserved when projected to an optimal lower dimension. Mathematically the JL Lemma can be stated as,

$$(1-\epsilon)|u-v|^2 \leq |f(u)-f(v)|^2 \leq (1+\epsilon)|u-v|^2 \tag{3.11}$$

where $0 < \epsilon < 1$ and f is a linear transformation map which maps data point from original dimension into lower dimension.

This inequality can be rewritten as:

$$(1-\epsilon) \leq \frac{|f(u)-f(v)|^2}{|u-v|^2} \leq (1+\epsilon) \tag{3.12}$$

From the above equation, it is evident that ϵ bounds the distortion between data points in the original and projected dataset. The higher the value of ϵ the higher the distortion. The minimum value of ϵ that makes the above inequality true for a data set given the transformation function f can be considered as a measure of distortion. $\varepsilon\ \epsilon[0,\infty]$. Equation 3.12 is intended for use without pairwise distance scaling. We also considered allowing transformations that scale pairwise distance. Let's represent scale factor α relative to the relationship:

$$\alpha|u-v|^2 \leq |f(u)-f(v)|^2 \tag{3.13}$$

Then, the inequality 3.12 becomes:

$$(1-\epsilon) \leq \frac{|f(u)-f(v)|^2}{\alpha|u-v|^2} \leq (1+\epsilon) \tag{3.14}$$

Let D be our dataset; $u, v\epsilon D$; and f be a transformation function on D. Let α be the associated scale factor. In this thesis, we assume that our transformations preserve distance (to some degree, which we are trying to measure) up to a scale factor. Of course, not all transformations do this. We can estimate the scale factor as the expected value of the distance ratios (the ratio of the distance between data points in low-dimensional space to the distance between the same data points in high-dimensional space) over all pairs of points $u, v\epsilon D$ where $u \neq v$.

$$\alpha = E(\frac{|f(u)-f(v)|^2}{|u-v|^2}) \tag{3.15}$$

The estimate is not ideal, as we can see in the case of PCA. For PCA, the (theoretical) scale factor should be one, but because of distortion, the mean of the distance ratios

will likely not be one. Returning to the inequalities above, extended from the Johnson-Lindenstrauss lemma to account for scale factor, one way to define distortion would be the following equation:

$$Distortion1(f, D) = Var(\frac{|f(u) - f(v)|^2}{\alpha^2|u - v|^2}) = \frac{1}{\alpha^4}Var(\frac{|f(u) - f(v)|^2}{|u - v|^2}) \tag{3.16}$$

Note that α is computed as described above, estimating the scale factor in the transformation. The variance here is computed on the squares of the distances between points. Another way to define distortion, separate from the inequalities above, is to use the variance of the distances:

$$Distortion2(f, D) = Var(\frac{|f(u) - f(v)|}{\alpha|u - v|}) = \frac{1}{\alpha^2}Var(\frac{|f(u) - f(v)|}{|u - v|}) \tag{3.17}$$

Distortion1, Distortion2 $\epsilon[0, \infty]$. The lower the distortion value the better the pairwise distances are preserved after dimension reduction.

3.5.7 Spearman Rank Correlation Coefficient

Spearman Rank Correlation [18] is a non-parametric measure of rank correlation. It measures how monotonically the ranks in one array change with respect to another array. We use it to estimate the degree to which the relative order of neighbors of each data point is preserved after dimension reduction. When all the ranks are distinct, the following equation can be used to compute the Spearman Correlation.

$$r_s = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)} \tag{3.18}$$

where n is the number of data points in either array.

d_i is the difference in rank of ith point in two arrays.

Throughout this thesis, this metric has also been referred to as 'Spearman'. The idea that Spearman could be used to measure quality of projection came from the fact that we are dealing with arrays containing ranks of neighbors. The value of Spearman ranges from -1 to 1 where -1 represents the two arrays of ranks are in reverse order while a value of 1 represents that two arrays of ranks are identical.

Spearman is a strict measure of preservation of the structure. It has one drawback (when used with subsets of ranks) that it does not actually check the identity of the neighbors preserved. For instance, if the neighbors of data points a and b have index [1, 2, 3, 4, 5] and [10, 20, 30, 40, 50] respectively, then their Spearman is 1 even though a and b do not have common neighbors. This inspired the conception of Spearman Variant.

3.5.8 Spearman Variant

This thesis proposes the modification of the Spearman Rank Correlation Coefficient metric to better reflect the difference in the rankings of a data points' neighbor in the original high dimension and the reduced lower dimension for the measurement of projection quality. It also addresses the shortcomings of Spearman (when used with a subset of ranks).

For each point in the list of neighbors in high dimension, we compute the sum of the difference of ranks in high dimension and low dimension. Spearman variant is optimistic in nature. When a point that ideally should have been a neighbor is not found in the k nearest neighbors list, it assumes that it is the first point outside the k nearest neighbor i.e it is the k+1 th neighbor. If another neighbor that is expected to be in the list of neighbors for projected neighbor is missing, we assume it is the k+2 th neighbor and so on. To normalize the value of Spearman Variant, we divide the sum of differences by the maximum possible difference sum for k neighbors under our optimistic assumption. To compute Spearman Variant for the whole dataset, the score for each pair of data points should be averaged.

For points u, v with neighbors U and V respectively, the Spearman Variant can be computed as the following:

Algorithm 1: Spearman Variant

```
N = k
for i = 1 to k do
    if U[i] in V then
        diff += abs(i - rank_U[i]_in_V)
    else
        diff += N - i
        N += 1
    end
end
score = 1 - diff / (k * k)
```

The optimistic assumption of the Spearman Variant makes it computationally efficient because we do not have to determine the actual rank of missing neighbors. Like Spearman, Spearman Variant is also concerned with the relative order in which the neighbors of a data point are preserved.

3.5.9 Top k Normalized Distance

This metric developed in this thesis also considers the relative order in which the neighbors of a data point are preserved after dimension reduction.

Intuitively, this metric describes how much better the preservation of neighbors in the dimension-reduced dataset is compared to the worst possible preservation of neighbors.

If the order of neighbors are the same in the dimension-reduced dataset and the original high-dimensional dataset then the metric has value 1. The range of Top k Normalized Distance is [0, 1]. The higher the score the better preservation after dimension reduction.

The difference between the ranks can be computed in two ways using L1 norm and using L2 norm. Top k Normalized Distance using L1 norm and L2 norm is referred to as Top k L1 norm distance and Top k L2 norm distance respectively. L1 norm is defined as the following:

$$\|x\|_1 = |x| \tag{3.19}$$

Similarly, L2 norm is defined by the following equation:

$$\|x\|_2 = \sqrt{x_1^2 + ... + x_n^2} \tag{3.20}$$

For points u, v with neighbors U and V respectively the Top k Normalized Distance can be computed as the following:

Algorithm 2: Top k Normalized Distance

```
N = number of samples in dataset
total_diff = 0
total_max_diff = 0
for i = 1 to k do
    diff = abs(V[i] - U[i])
    max_diff = abs(N - 1 - i * 2)
    if norm == 'L2' then
        diff = diff * diff
        max_diff = max_diff * max_diff
    else
        pass
    end
    total_diff += diff
    total_max_diff += max_diff
end
score = 1 - total_diff / total_max_diff
```

3.6 Discussion and Results

The number of components preserved after dimension reduction, n_component, was chosen so that the initial improvement in classification performance and projection quality could be captured easily. Both the quality of projection and the classifier performance peak and plateau after the first few n_components implying that fewer observations are required as the size of reduced dataset increases. The results of quality of projection metrics for

various dataset are presented here.

3.6.1 Results on MNIST dataset (Benchmark)

The MNIST [19] dataset was used for the purpose of demonstrating the changes in projection quality on varying n_component and subsequently measure it using the metrics discussed above although it is different from our dataset in many ways like dimensionality and the fact that it is a balanced dataset. Even though it might seem that transportation images for a specific location are mostly the same and thus different from MNIST. The transportation images surely do have variations. The variation among the images is caused by changing time of the day, changing climate throughout the year, weather change throughout the day, movement of people, vehicle, animal, etc.

The number of MNIST dataset samples used is 1000 which was sampled randomly and PCA was used for dimension reduction.

Compared to other metrics for quality of projection, Trustworthiness and Continuity propose aggressive n_components for equivalent projection quality. Trustworthiness and Continuity assume the worst for the missing neighbors. So, when the neighbors start to appear in the k nearest neighbors list after dimension reduction, the metric results in higher values. The difference is not large when the number of nearest neighbors considered changes from 10 to 20.

Ratio Preserved shows us that as n_component increases, the proportion of common neighbors in high-dimensional space and low-dimensional space increases. The increase is significant for the first few n_component and plateaus at n_component > 200. Figure 3.3 shows that neighbors in both the Euclidean space and cosine space are preserved equally well. The value of Ratio Preserved was computed for k = 1, 10, and 20. Ratio preserved with k = 1 has special significance as it indicates if the nearest neighbor is preserved or not. As stated previously, as the number of neighbors considered (k) increases, the value of Ratio Preserved increases too which is evident in the comparative Figure 3.3.

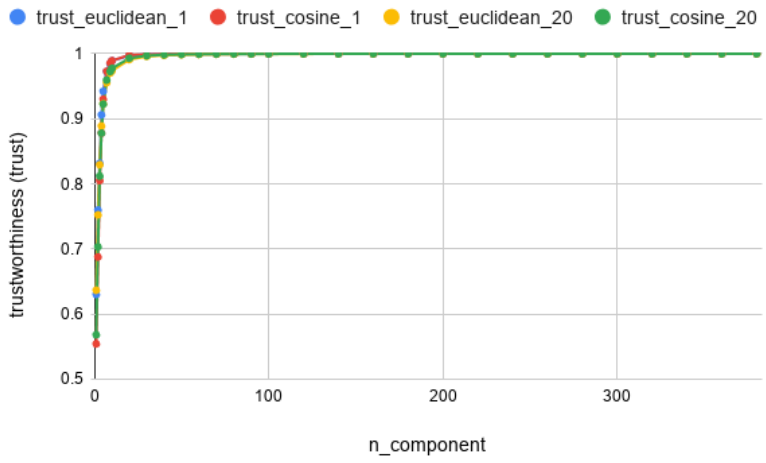

Fig. 3.1: Trustworthiness versus n_component. Euclidean and cosine distances were used to compute kNN with k = 1, 20. The range of vertical axis has been shortened for display purposes.

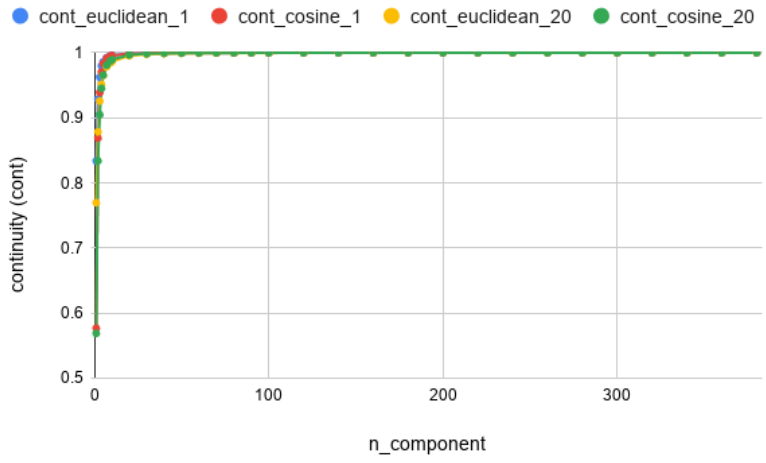

Fig. 3.2: Continuity versus n_component. Euclidean and cosine distances were used to compute kNN with k = 1, 20. The range of vertical axis has been shortened for display purposes.

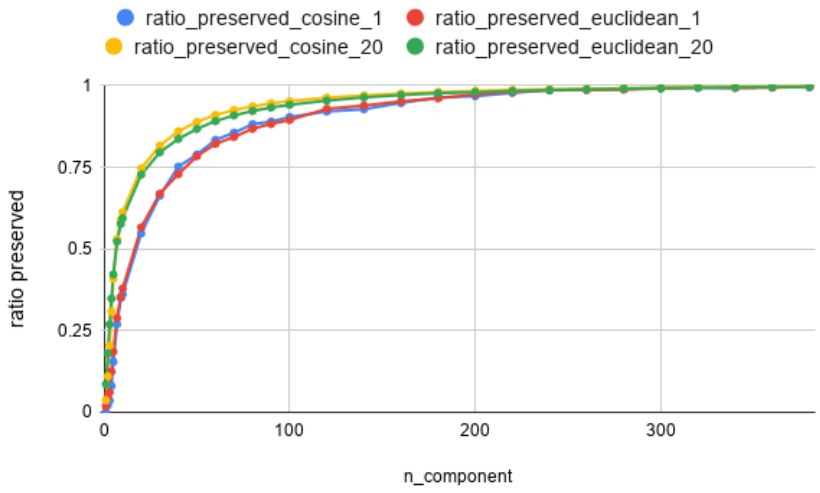

Fig. 3.3: Ratio Preserved vs n_component. Euclidean and cosine distances were used to compute kNN with k = 1, 20. The neighbors of data points are preserved equally well in both Euclidean space and cosine space.

Spearman estimates the degree to which the relative order of neighbors of each data point is preserved. The quality of projection increases gradually on increasing n_component as shown in fig: 3.4. It can be noted that spearman_10 > spearman_N which implies that it is easier to preserve the nearest neighbors and challenging to preserve the farthest neighbors. Also, spearman_10 (considering the nearest 10 neighbors) increases faster than spearman_N (considering all the N-1 neighbors) in figure 3.4. spearman_N tells us that the ranking of all the neighbors are not preserved perfectly even at n_component = 500 and initially (n_component < 200) the neighbors are preserved in somewhat random order.

Figure 3.5 shows quality of projection measured by Spearman Variant increases rapidly plateauing at n_component > 200. Spearman Variant behaves like Ratio Preserved except that it also takes into account the relative order of preservation of the neighbors. It increases on increasing the number of neighbors considered but does not become 1 until the neighbors are preserved in perfect order. So unlike Ratio Preserved which is favorable to use when k << N, Spearman Variant can also be used for k $\epsilon[1, N]$.

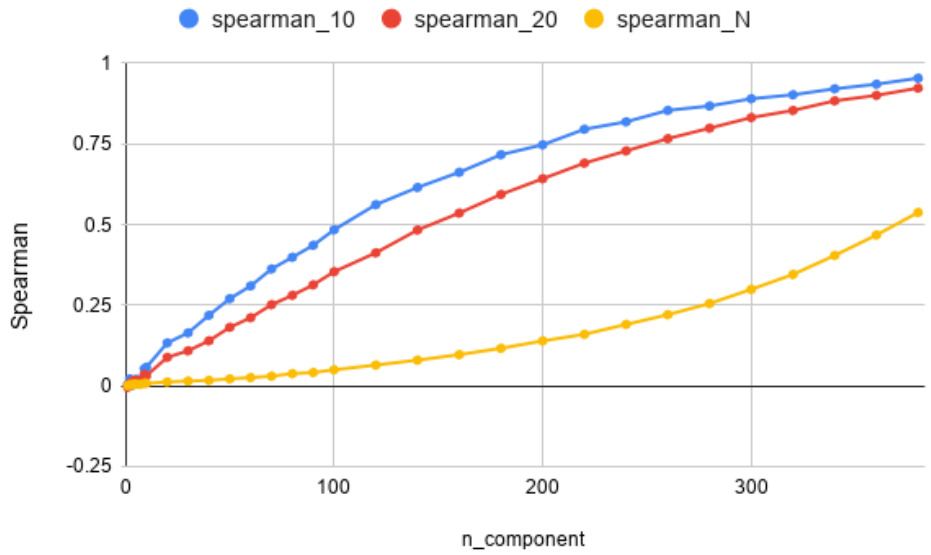

Fig. 3.4: Comparing the Spearman for different values of k (10, 20, N). Euclidean distance was used to compute kNN.

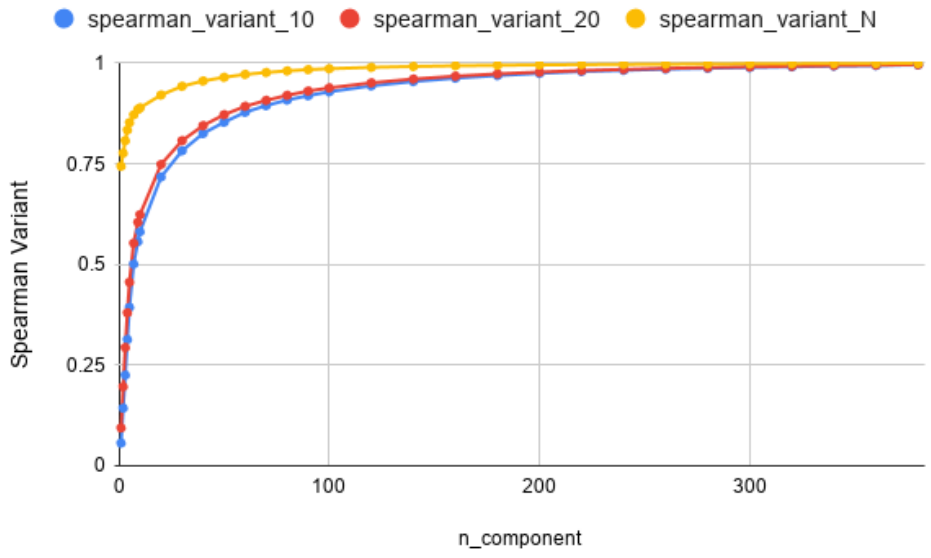

Fig. 3.5: Comparing the Spearman for different values of k (10, 20, N). Euclidean distance was used to compute kNN.

The results of measure Top k L1 Normalized Distance and Top k L2 Normalized distance are shown in figure 3.6 and Figure 3.7 respectively. They were computed for the following values of neighbors considered: 10, 20 and N. Like Trustworthiness and Continuity, Top k Normalized Distance also assumes the worst for the missing neighbors. The estimates for the quality of projection tell us how the current structure preservation compares with the worst i.e. if neighbors for all the data points were reversed in order. The kNN was computed using Euclidean distance as distance metric.

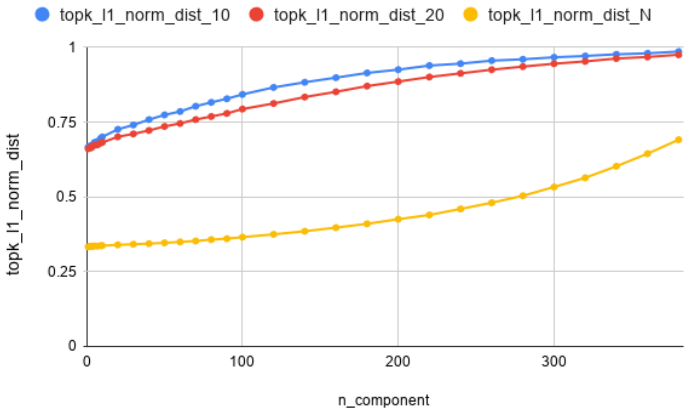

Fig. 3.6: Top k L1 norm distance versus n_component for k = 10, 20, N

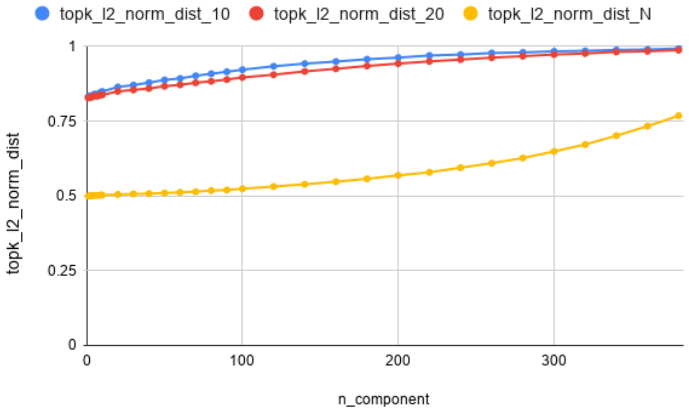

Fig. 3.7: Top k L2 norm distance versus n_component for k = 10, 20, N

The results for Distortion1 and Distortion2 are shown in Figure 3.8 and Figure 3.9 respectively.

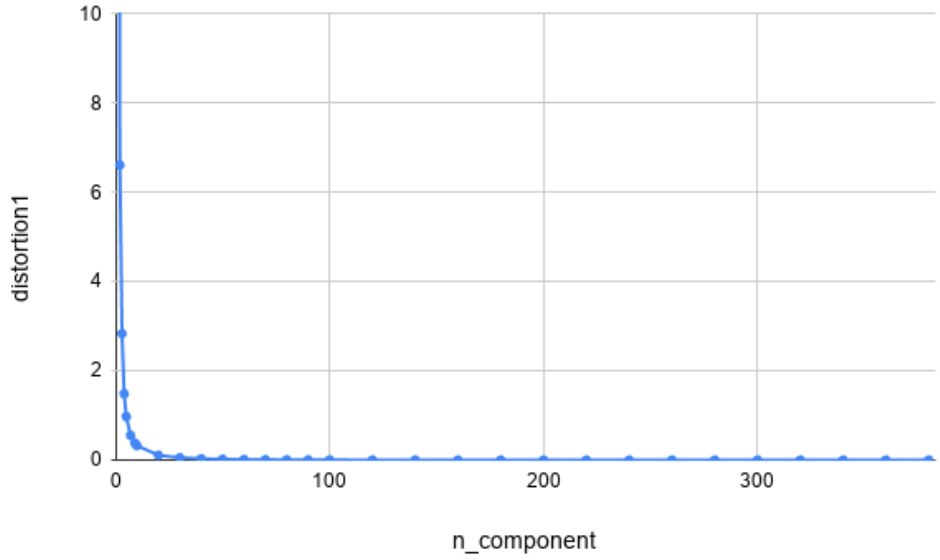

Fig. 3.8: Distortion1 versus n_component. The value of Distortion1 at n_component = 1 is 40.6 which is out of bounds in this graph.

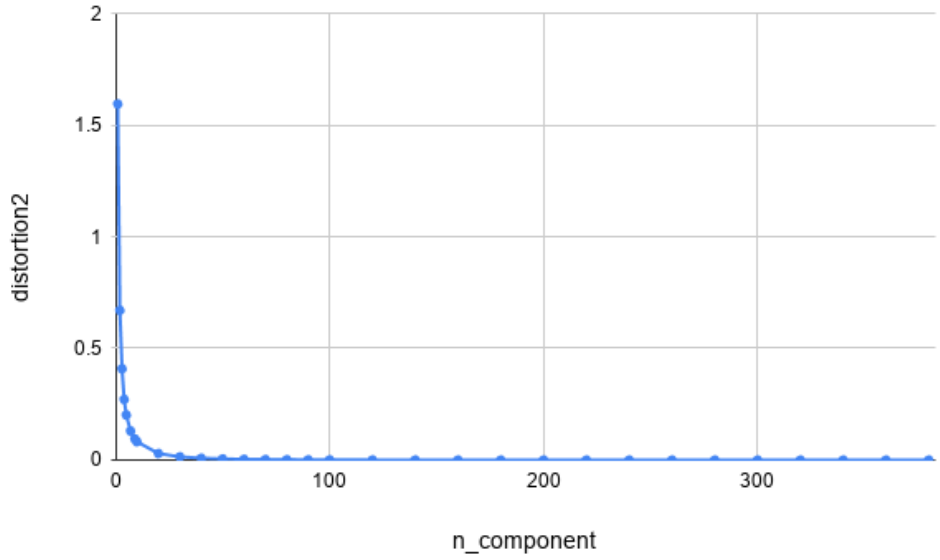

Fig. 3.9: Distortion2 versus n_component.

Distortion Factors (Distortion1 and Distortion2) are based on distance ratios (the ratio of the distance between two data points in the low-dimensional space to the distance between them in the high-dimensional space). Even though the local structure of the whole dataset is somewhat preserved when the number of components preserved is very low (1 or so), distance ratio is < 1 leading to a very high distortion factor. As n_component increases, the distance ratio swiftly increases towards a value of 1. The quality of projection rapidly increases as shown by the decreasing value of Distortion Factors.

3.6.2 Results on Spring Garden Dataset

The sample size for all the experiments below in this section was 1000 and PCA was used for dimension reduction. The results of metrics measuring the quality of dimension reduction for Spring Garden is given below. Euclidean distance was used to compute kNN unless mentioned explicitly.

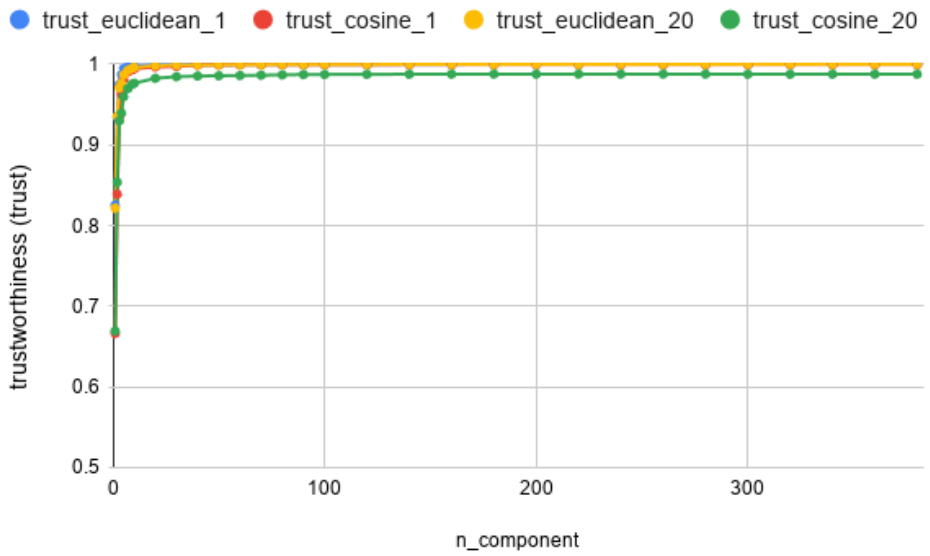

Fig. 3.10: Trustworthiness versus n_component. Euclidean and cosine distances were used to compute kNN with k = 1, 20. The range of vertical axis has been shortened for display purposes.

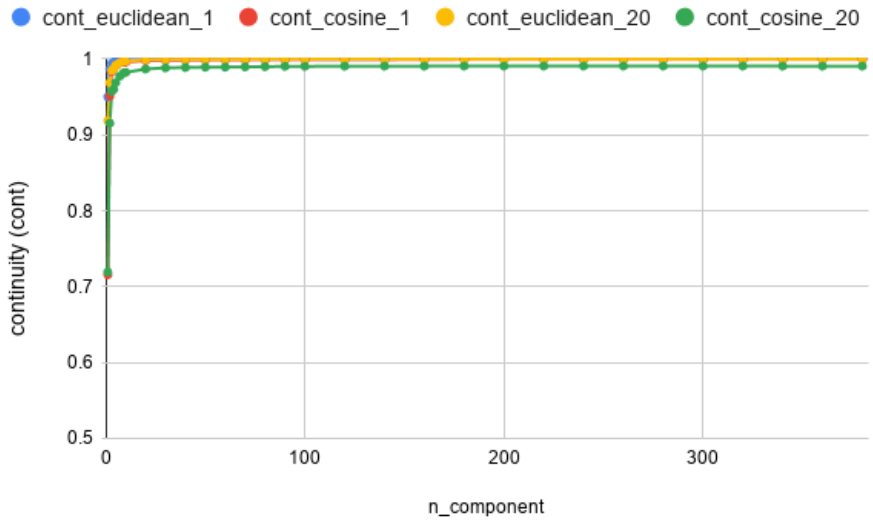

Fig. 3.11: Continuity versus n_component. Euclidean and cosine distances were used to compute kNN with k = 1, 20. The range of vertical axis has been shortened for display purposes.

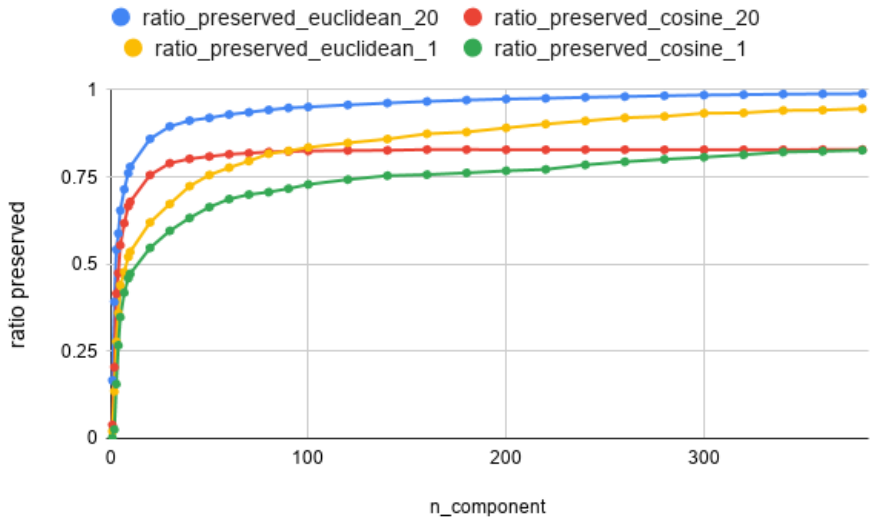

Fig. 3.12: Comparing Ratio Preserved vs n_component for k = 1, 20. kNN was computed using Euclidean distance and cosine distance metrics as indicated by the labels.

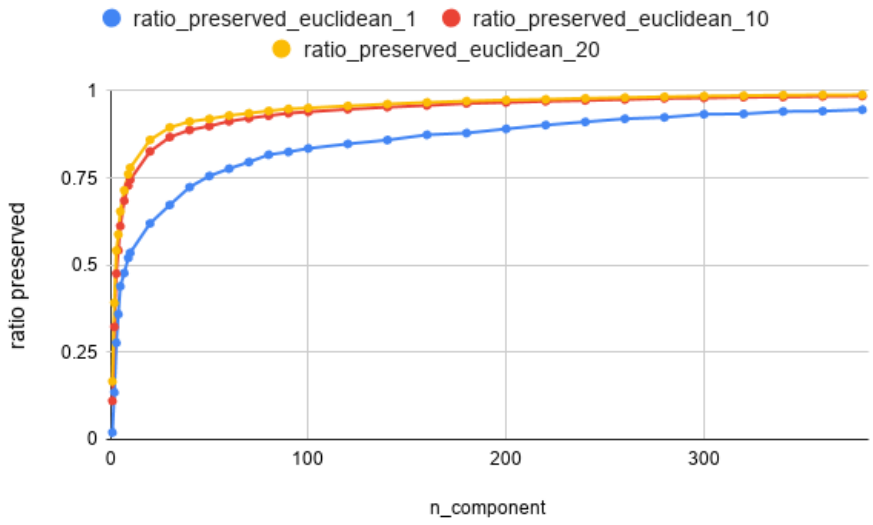

Fig. 3.13: Comparing Ratio Preserved for different values of k (1, 10, 20). Euclidean distance was used to compute kNN.

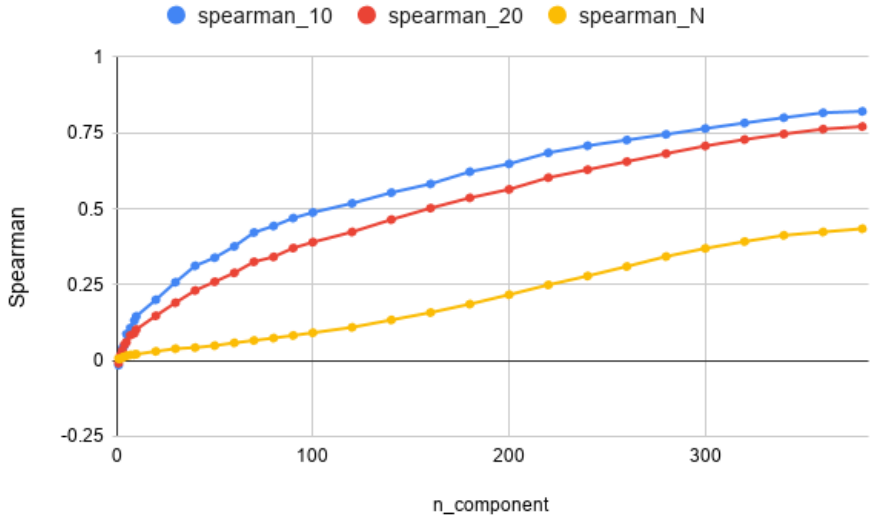

Fig. 3.14: Comparing Spearman for different values of k (10, 20, N). Euclidean distance was used to compute kNN.

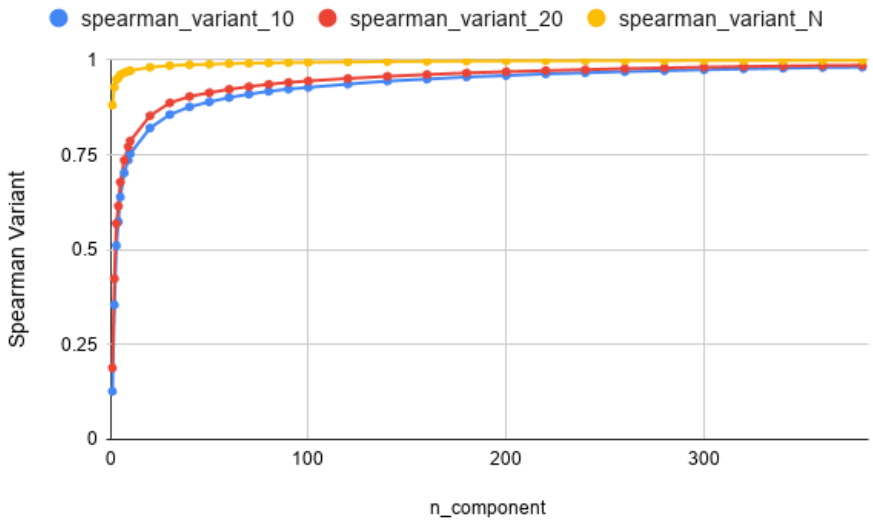

Fig. 3.15: Comparing Spearman Variant for different values of k (10, 20, N). Euclidean distance was used to compute kNN.

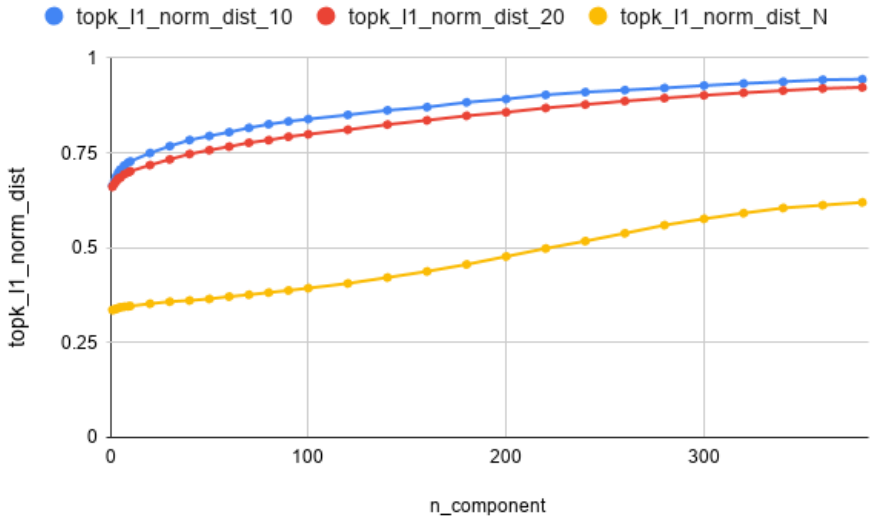

Fig. 3.16: Top k L1 Normalized Distance versus n_component for k = 10, 20, N

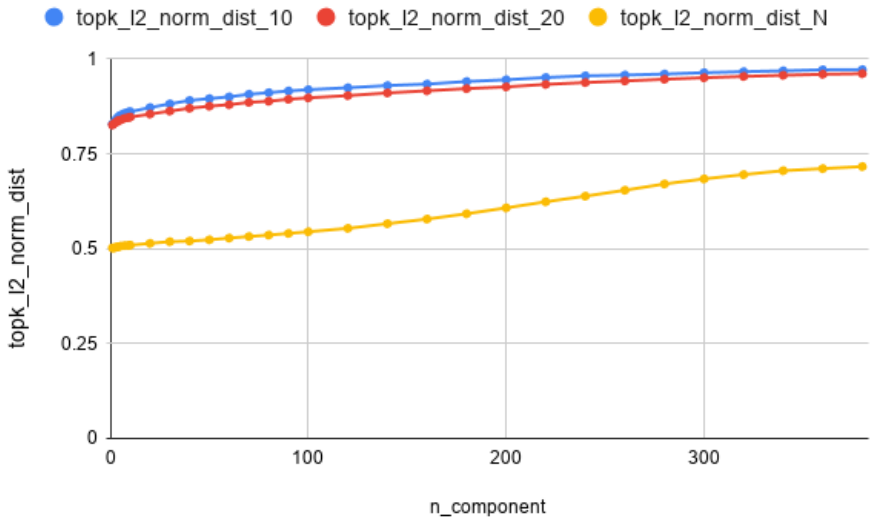

Fig. 3.17: Top k L2 Normalized Distance versus n_component for k = 10, 20, N

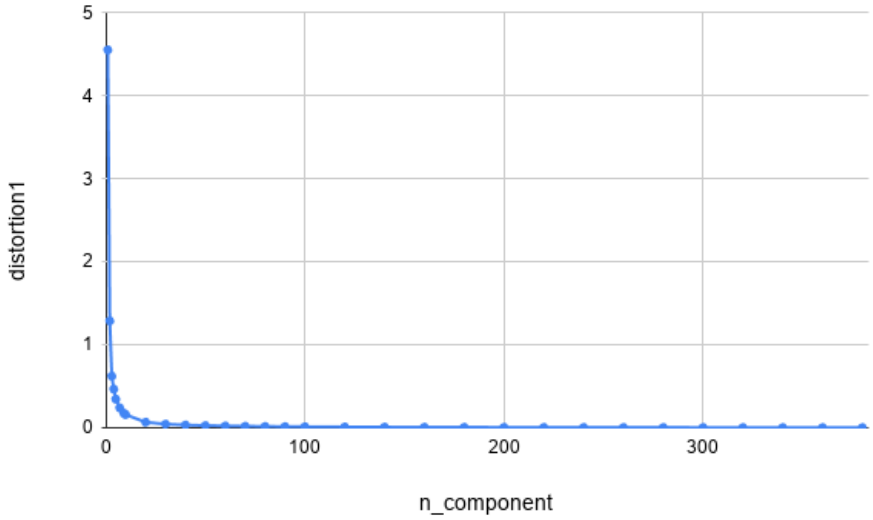

Fig. 3.18: Distortion1 versus number of components preserved (n_component)

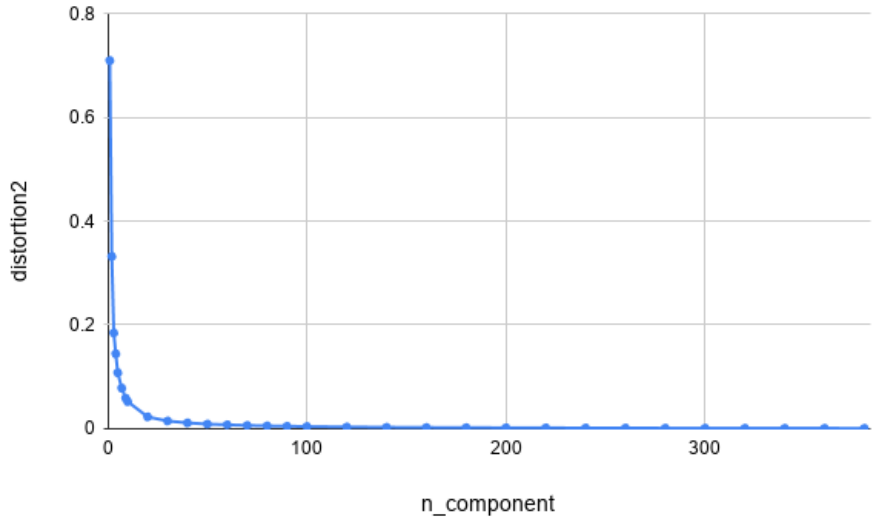

Fig. 3.19: Distortion2 versus number of components preserved (n_component)

3.6.3 Comparison with benchmark: The similarities and differences

There are more similarities than differences between the results of metrics measuring quality of projection even though the two datasets are different in many ways. Most of the metrics have similar results on both the datasets. The cosine distances between two data points are preserved better in MNIST dataset than in the Spring Garden dataset for any given n_component as shown in the results of Trustworthiness, Continuity, and Ratio Preserved metrics.

3.6.4 Comparison of the Dimension Reduction methods

We computed the metrics for quality of projection for several values of n_component for PCA, RP, and Image Scaling. We plotted metrics for measuring quality of projection against n_component for all the methods of dimension reduction used. We found a unanimous result that PCA has the best quality of projection, followed by Random Projection and Image Scaling has the worst quality of projection. The same conclusion holds true for the Spring Garden dataset. All the graphs in this section were plotted for MNIST dataset.

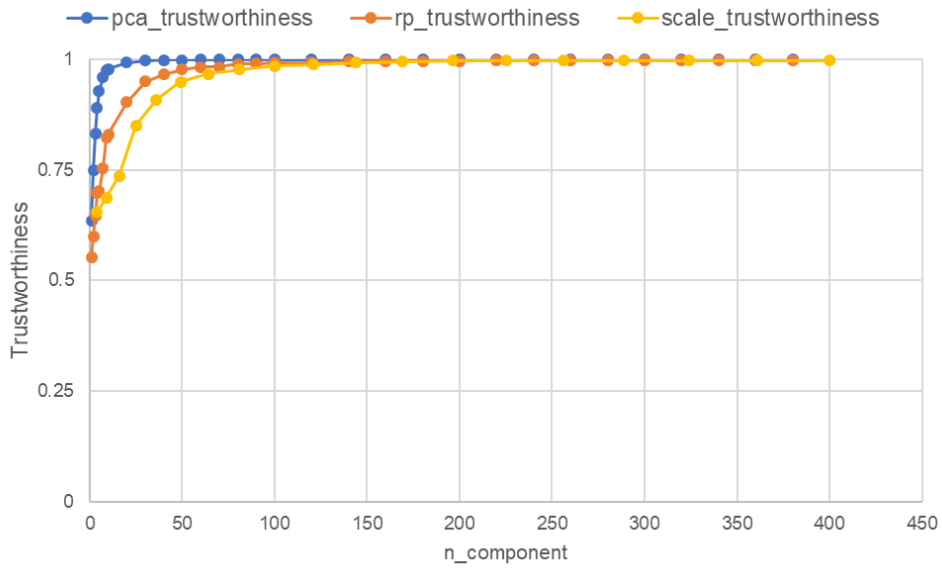

Fig. 3.20: Trustworthiness versus n_component for PCA, RP, and Image Scaling

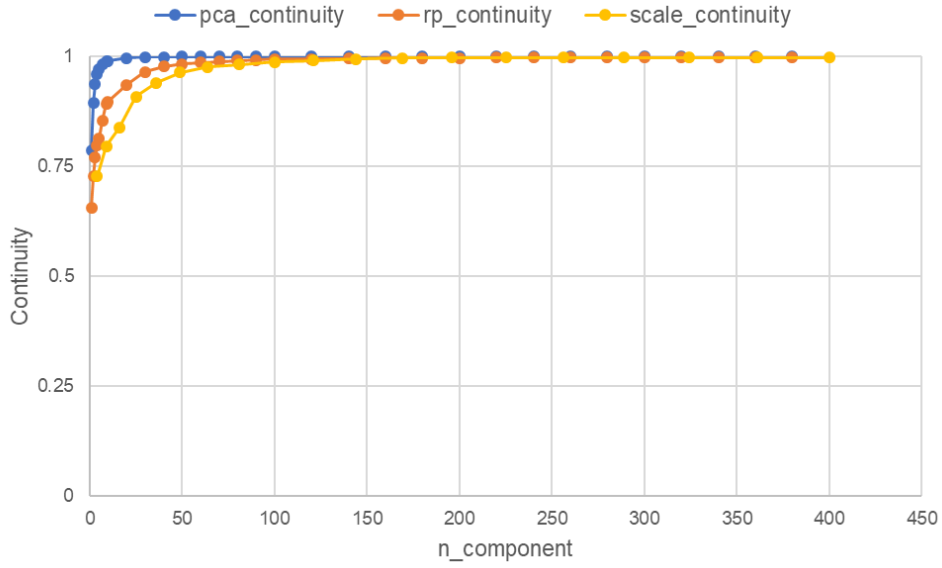

Fig. 3.21: Continuity versus n_component for PCA, RP, and Image Scaling

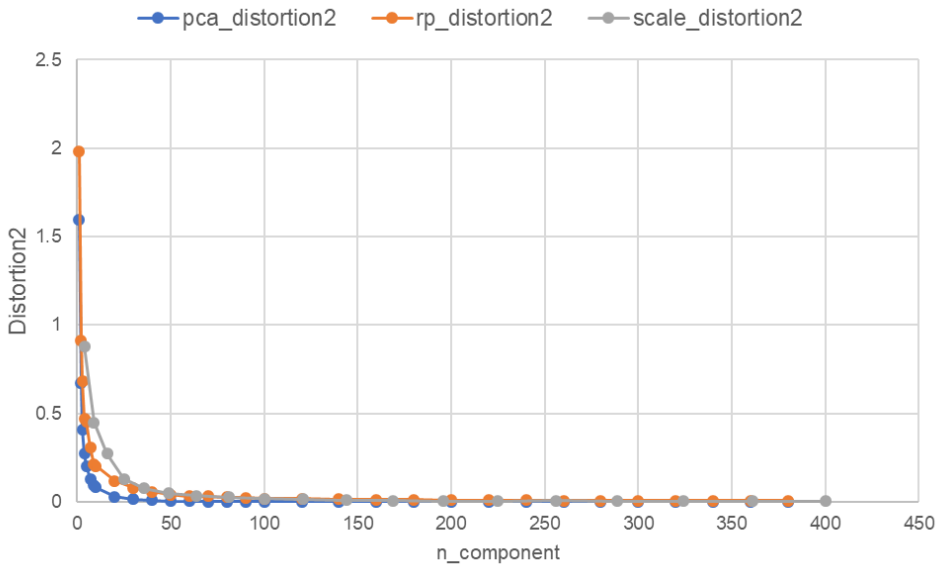

Fig. 3.22: Distortion2 versus n_component for PCA, RP, and Image Scaling

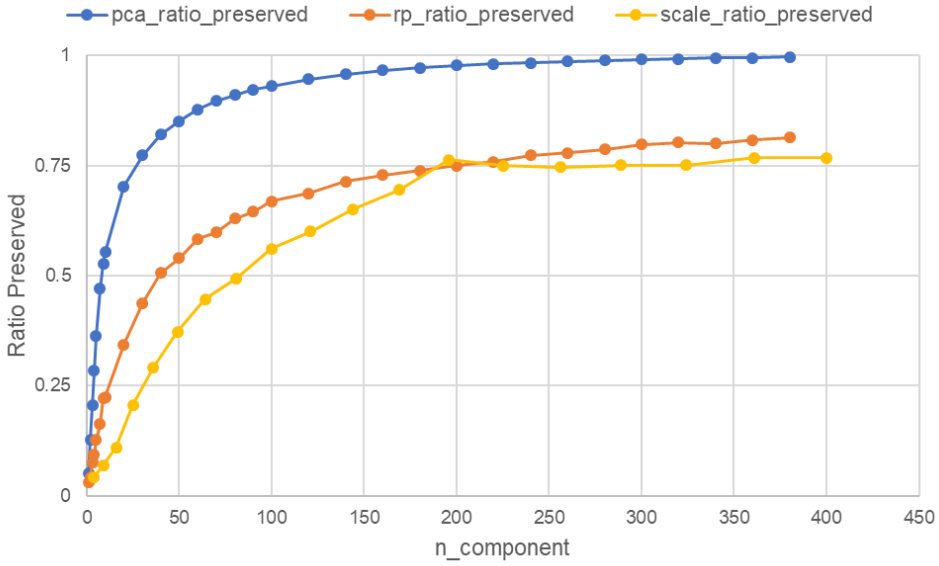

Fig. 3.23: Ratio Preserved versus n_component for PCA, RP, and Image Scaling

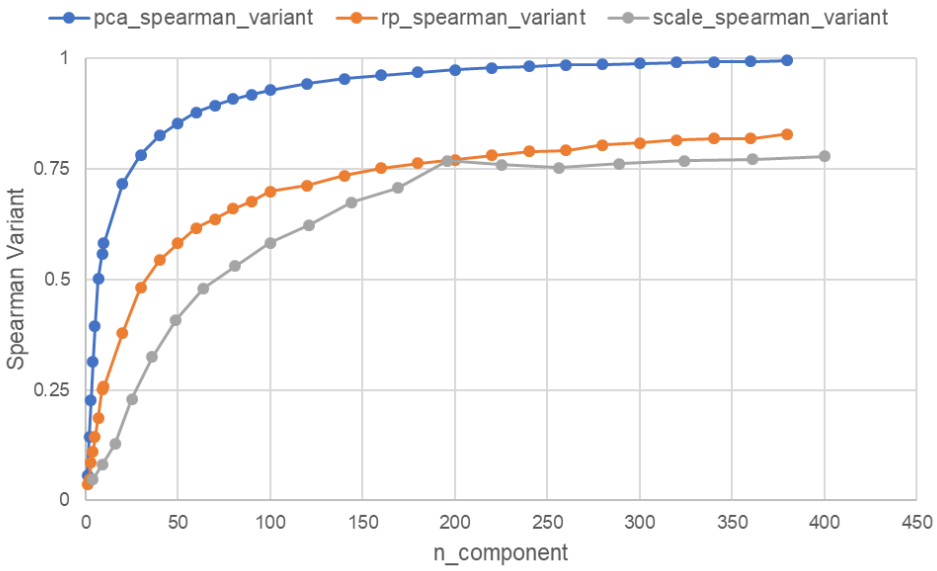

Fig. 3.24: Spearman Variant versus n_component for PCA, RP, and Image Scaling

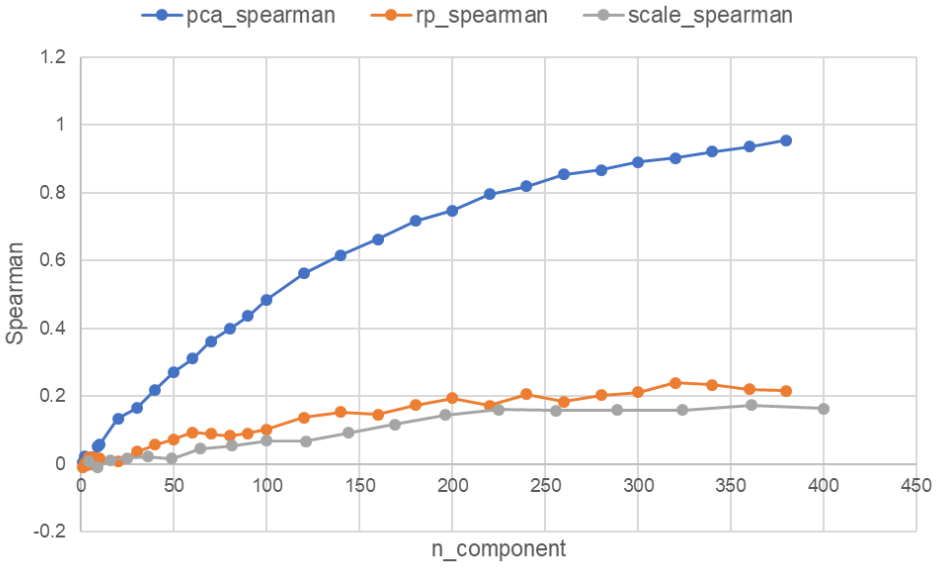

Fig. 3.25: Spearman versus n_component for PCA, RP, and Image Scaling

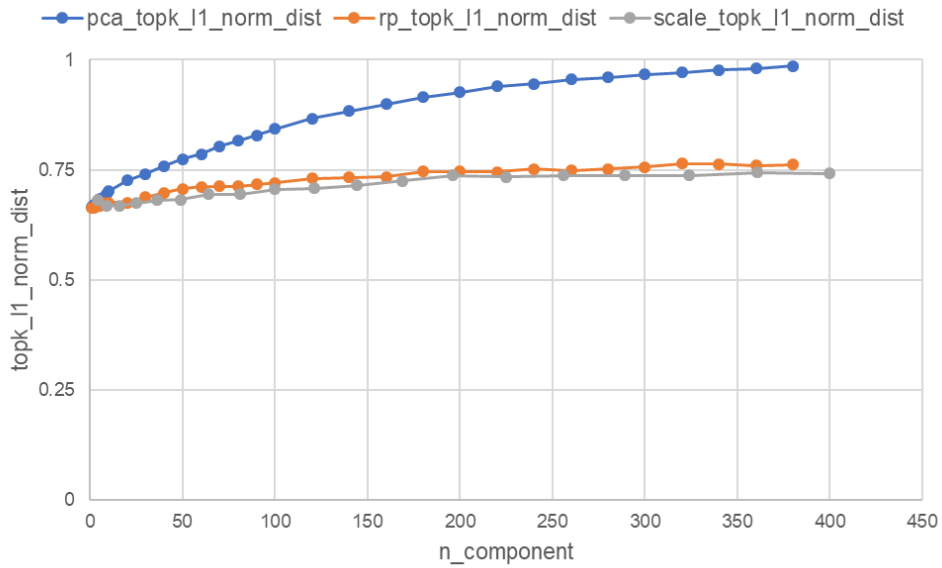

Fig. 3.26: Top K L1 Normalized Distance (top_k_l1_norm_dist) versus n_component for PCA, RP, and Image Scaling

3.6.5 Why do different metrics peak and plateau at different n_components?

First, they measure different aspects of preservation of structure of a dataset. For instance, Ratio Preserved is only concerned with how k nearest neighbors of a data point are preserved. It does not take into account sample size or the relative order in which the neighbors are preserved. Spearman also measures the preservation of structure but it estimates the preservation of order of neighbors for each data point. One major difference between Spearman Variant and Top k normalized distance is the way they treat their missing neighbors. Spearman Variant assumes optimistically that the first missing neighbor is the k+1 th neighbor and the second missing neighbor is the k+2th neighbor and so on. Top k Normalized Distance pessimistically assumes the worst i.e. the first missing neighbor might as well be the last neighbor and the second missing neighbor is in the second to last position in the list of the nearest neighbors after dimension reduction and so on. Trustworthiness and Continuity do not account for the neighbors that are actually k nearest neighbors before as well as after dimension of the dataset. They only deal with the missing nearest neighbors.

3.7 Conclusion

Some of the notable conclusions reached in this chapter are.

- As the size of the reduced dataset increases, the quality of projection increases i.e. the local structure of the dataset is better preserved.
- Random Projection has erratic performance throughout all n_components owing to its nature of generating the transformation matrix randomly. Even though the projection of quality gradually increases on increasing n_component, it cannot be ensured that projection quality is greater for each successive n_component.
- PCA outperforms Random Projection on every metric. Given the same number of components, it preserves the structure of dataset better than Random Projection.
- Common statistical tools like Spearman Rank Correlation can be used to measure quality of projection. It is perfect to estimate the order of preservation of neighbors of each data point.
- Both PCA and Image Scaling plateau at n_component > 200 while the quality of projection for Random Projection keeps increasing on increasing n_component.
- Among all quality of projection metrics, Trustworthiness and Continuity suggest overly optimistic and aggressive n_component for dimension reduction for a given value of quality of projection.
- For any metric estimating projection of quality, Image Scaling requires the highest n_component to achieve equivalent projection quality which means that PCA obtains better feature representation for given output dimension.
- Using cosine as a distance metric while computing neighbors yields a similar result to that of using Euclidean distance as a distance metric. It also signifies that pairwise distance is preserved both for Euclidean distance metric and cosine distance metric when projected to lower dimension with optimal dimensionality.

CHAPTER 4

Image Preprocessing

4.1 Introduction

In this chapter, we investigate the usage of appropriate image representation for all of the experiments in this thesis. In this chapter, we can treat the classifiers as black boxes. They are discussed in detail in the next chapter 'Image Classification.'

Pre-processing is performed in the early stages of machine learning pipeline. It is done to represent the data better to subsequently improve classification performance. In this thesis, pre-processing is mostly image processing. Image processing is the process of manipulating images using some algorithms to achieve a desirable effect.

Considering that images of a single dataset belong to the same location, the images taken on similar days (weather and lighting) should be very similar except for the movement of people, animals, or vehicles. That is not the case, however. The change in time of the day and seasonal variation cause variation among images of the same location. Even images taken in sequence can be very much different from each other making the application of algorithms difficult. For instance, it is comparatively easy to sample representative images when there is less variation in images. Sampling representative images, in turn, lead to better classification performances because a classifier trained on a highly representative training set is bound to perform well on test sets.

Lack of data is often presented as a problem. However, the availability of gigabytes of domain-specific datasets may also be a problem. The problem is that the available dataset is not representative enough; that is there is not enough variance in the dataset or, simply, not enough differences between two data points in the dataset. We also face the same problem. Even though there are thousands of images taken from the same fixed spot overlooking the same roads, there is always a chance that a newly captured image has not been seen before.

With the application of image processing, the intent was to account for and compensate lighting differences. To that end, several color spaces and other techniques such as histogram equalization were explored. Note that the choice of image processing techniques applied to the dataset has repercussions on both computational complexity and classification performance.

4.1.1 Color Spaces

Fig. 4.1: RGB image and its' channels. a, b, c and d indicate all RGB channel, red channel, green channel and blue channel image respectively.

Fig. 4.2: HSV image and its' channels. a, b, c and d indicate all HSV channel, hue channel, saturation channel and variance channel image respectively.

Processing images is computationally expensive. A 704 * 240 color image implies there

Fig. 4.3: RGB image on left and equivalent grayscale image on right.

are 704 * 240 * 3 (=506880) 8-bit integers in the image. It is desirable to reduce the dimensionality of the dataset to reduce subsequent computations on the image.

Also, as the CCTVs are fixed with only some preset movements allowed we would expect similar images except for vehicular, human, and animal movements which is not the case. Even for the same time of the day, the variation in lighting causes the scenery to appear different.

To compensate for lighting differences, we explored several color spaces. A color space is a representation of colors. In the RGB color space, different values of the Red component, Green component, and Blue component combine to represent a color. Similarly, in the HSV color space, a color is represented by a combination of Hue, Saturation, and Variance values. A grayscale image is obtained by computing an average or weighted average of R, G, B values for every coordinate. A grayscale image can be stored in one-third of the memory required for HSV or RGB images and has only one-third of the parameters required to represent images in RGB or HSV color space. This is useful because it reduces computations on the image. Each color space has its merit and drawbacks. For instance, the HSV color space is particularly suitable to deal with changing illumination, the RGB color space is suitable for viewing images on screens, the CMYK (Cyan, Magenta, Yellow, and Black) for printing any document, and so on.

We performed several operations on images like color space conversion and histogram equalization to reduce the variability between images to make the training dataset representative enough.

4.1.2 Histogram Equalization

Fig. 4.4: RGB image on left and equivalent histogram equalized image on right.

Histogram equalization [20] is a process of adjusting contrast in images. It results in contrast-balanced images. Histogram equalization is usually applied to grayscale images because they are single-channel images. The idea behind using histogram equalized images is that resulting images will have less lighting difference.

4.1.3 Single Channel vs Multi-channel Images

A single pixel in a typical image is represented as an (R, G, B) triple. This implies that the image is made of three channels and a pixel is represented by the values of all three layers at that position. While all the channels in total have more information compared to a single channel, each channel can be considered to represent the same information but differently.

Using just one of the single channels of the RGB or HSV color space or histogram equalized or grayscale images reduces the dimension of the image. The images explored and exploited in this thesis were originally captured and stored in the RGB color space.

4.2 Discussion and Results

All the experiments were carried on the Spring Garden dataset consisting only of daytime images. The classifiers were trained on the Precipitation1hr attribute as that is the most balanced attribute besides the daynight attribute. The dataset consists of 1600 images and the test to train split was 25 to 75. Only SVM and kNN classifiers were used for classification. The last point on any graphs below shows the performance of classifiers on data set without dimension reduction.

4.2.1 Comparison of HSV Channels

Among the H, S, and V channels in HSV color space, the hue channel (H) has the best performance in the Figure 4.5 and 4.6. The accuracy and f1 score of the classifier is shown separately for the sake of clarity as differences are very small and all the lines clutter when shown together.

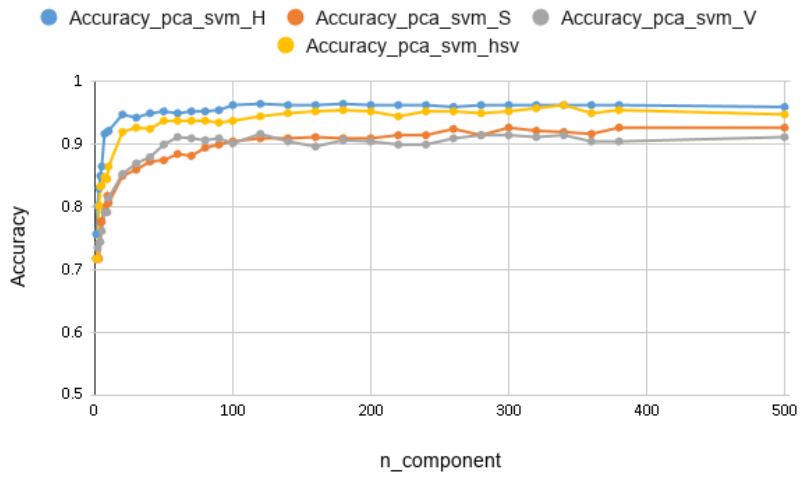

Fig. 4.5: Accuracy for H, S, V channels of the HSV color space. The hue channel has a clear lead.

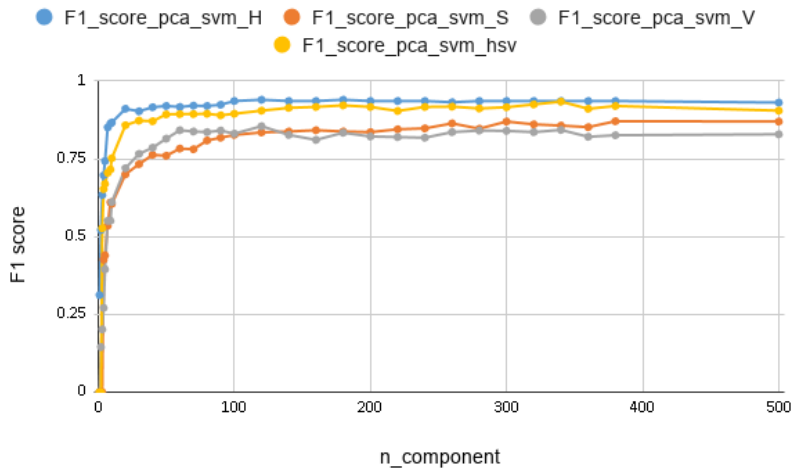

Fig. 4.6: F1 score for H, S, V channels of the HSV color space.

4.2.2 Comparison of RGB Channels

The accuracy and f1 score of the classifiers is shown in two separate Figures 4.7 and 4.8 respectively for the sake of clarity as differences are very small and all the lines clutter when shown together.

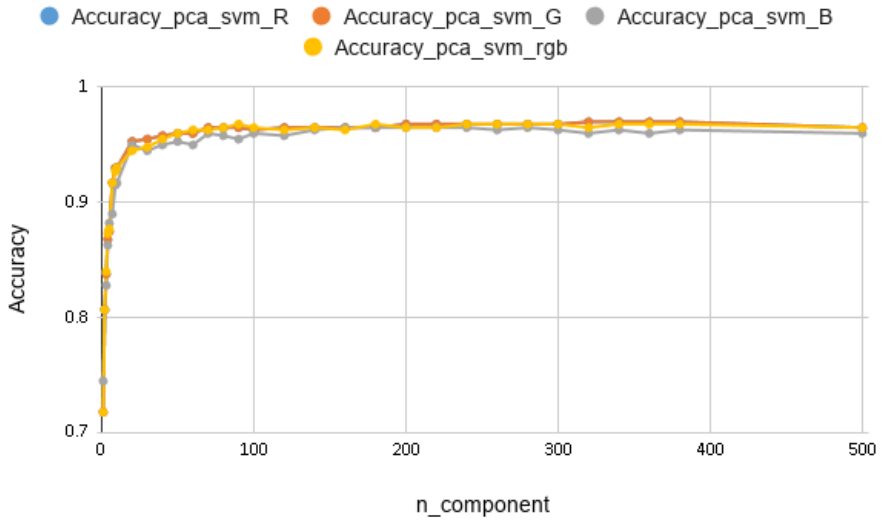

Fig. 4.7: Accuracy for R, G, B channels of the RGB color space.

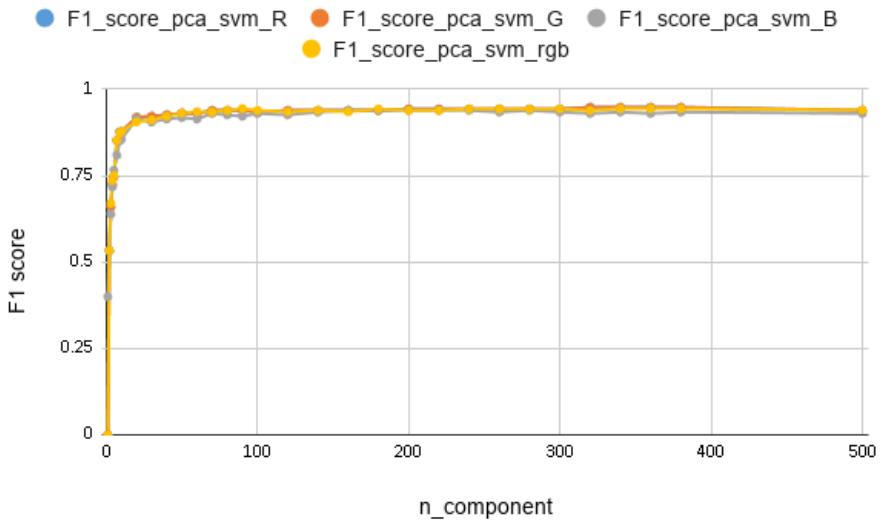

Fig. 4.8: F1 score for R, G, B channels of the RGB color space.

It evident that the blue channel is the lowest performer while the performance of the rest of the channels including full RGB is very similar. On closer inspection of Figures 4.7 and 4.8, it was found that the green channel has the best performance, but with a lead of less than 1%. This might not be the case everywhere but it can be considered that full RGB images perform at par with the best performing channel (R, G, or B) image.

4.2.3 Comparison of HSV versus RGB

The usage of full RGB color space results in 1-2% better performance than the usage of full HSV color space. The usage of RGB results in higher accuracy and a higher F1 score compared to the classifier using HSV color space. RGB is the default color space that images are most often stored which means no color space conversion is required to use RGB images while conversion is required to use HSV color space images.

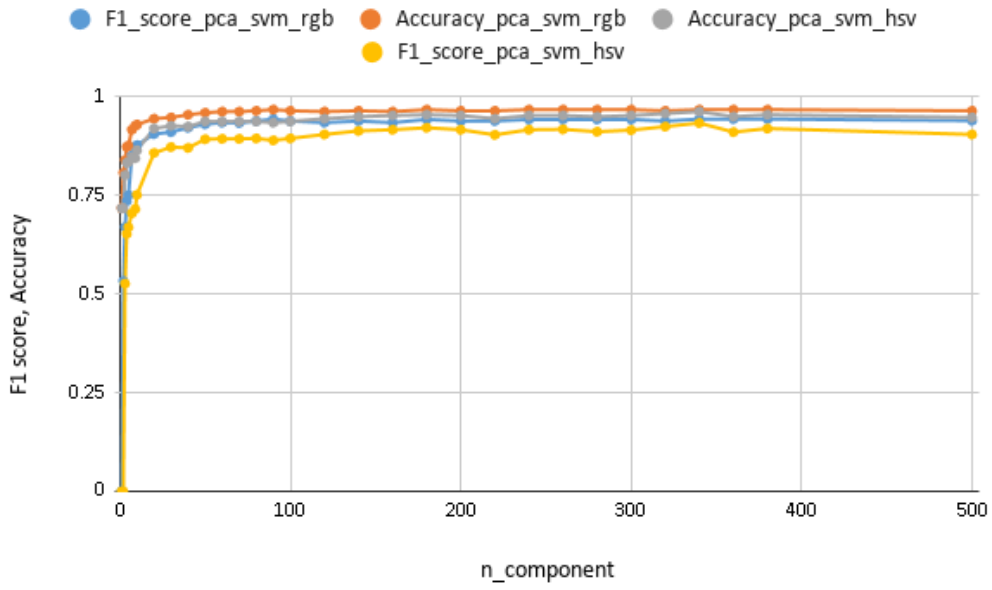

Fig. 4.9: Full Channel RGB versus full channel HSV. RGB has a couple of percentage point advantage compared to HSV.

4.2.4 Comparison of Single Channels

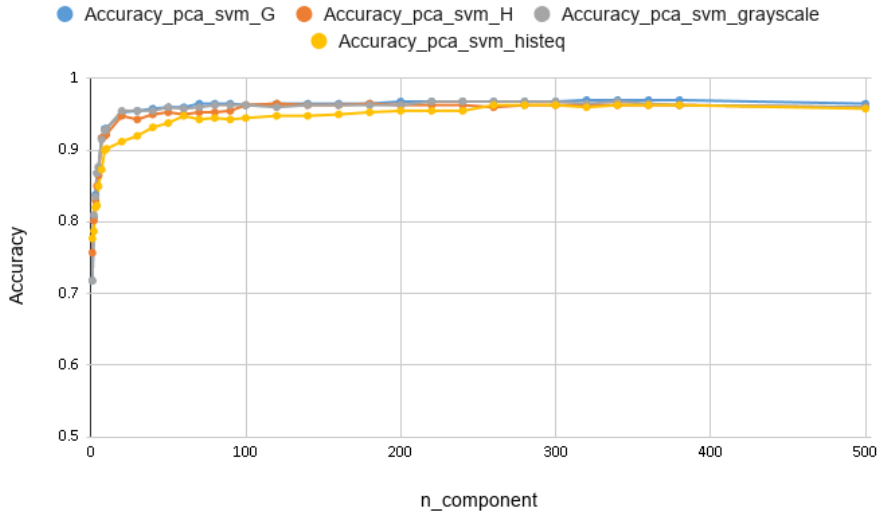

Fig. 4.10: Comparison of accuracy for Green, Hue, Grayscale, HistEq Image.

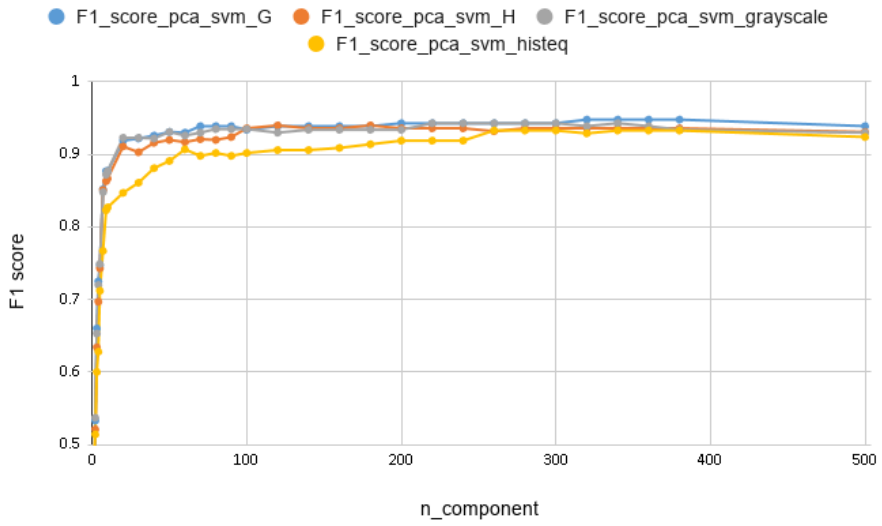

Fig. 4.11: Comparison of f1 score for Green, Hue, Grayscale, HistEq Image.

Among the R, G, B channels in RGB color space, Green has the best performance by a tiny margin, and the Hue channel clearly leads the performance among Hue, Saturation,

and Variance channel. The accuracy and f1 score of the classifier is shown in two separate graphs for clarity's sake as differences are very small and all the lines will clutter when shown together. The histeq (histogram equalized) image and hue channel have poor performance compared to grayscale image and green channel image; Green channel has the lead among them all on comparing both accuracy and f1 score. The classifier using grayscale images closely follows the performance of the classifier using the green channel image.

4.2.5 Performance on grayscale images

Classifiers trained on grayscale images perform on par with (or better than) other single channels or full channel RGB image while being more interpretable than the individual channels like Red, Green, or Blue. The performance of classifiers trained on grayscale images is more consistent than others across all attributes. The computational gains from not using full channel images or histogram equalized images are significant and it does not come at the cost of classification performance too.

4.3 Conclusion

The results of the image processing on the dataset are surprising. The HSV color space known for dealing with lighting changes in images did not help and the single-channel images outperformed multi-channeled images.

- RGB color space has better performance than HSV color space for classification of the attributes explored in this thesis.
- Using single-channel images like grayscale, green channel, or red channel leads to better or at par performance than using all channel RGB image.
- Using single-channel images like histogram equalized, grayscale or individual channels in the RGB color space leads to smaller model size and faster training and inference time (by a factor of at least 3).

- Classifiers trained on grayscale images performed on par or better than classifiers trained on full RGB channel images or other single-channel images while also being more interpretable.

- Since, the methods that we explored for compensating lighting difference did not work out, adding more data and exploring other image processing options are the only way to achieve better classification performance on the dataset explored in this thesis.

CHAPTER 5

Image Classification

5.1 Introduction

The data set that has been used is a time-series data set consisting of images from CCTV observing the roads and highways and sensors readings for weather and road conditions along the highways.

Our goal is to investigate how dimension reduction impacts classification and identify measures that best predict the impact. We explored a few classical machine learning algorithms to measure their performance on projected data sets. Two of the attributes that we build classifiers for are weather-related. Guerra et. al [21] state that a lack of discriminating features among various weather conditions can make it challenging to classify.

The classification algorithms that have been explored have solid mathematical foundations beneath them and have been part of numerous researches. They are listed below:

5.1.1 kNN Classifier

The kNN Classifier is a supervised classification algorithm based on the output of k-Nearest Neighbors (kNN). Once the kNN is obtained for a data point, it is classified according to the labels of its neighbors. In other words, the class assigned to a data point is determined by the class of its k nearest neighbors. The majority class of the neighbors can be assigned to be the class of the data point which is to be classified. The distance of the neighboring data points can also be taken into consideration during the voting. Such voting is termed as weighed voting.

5.1.2 Support Vector Machines

Support Vector Machines [22] (SVM) is a supervised classification algorithm. Given

a set of labeled data with binary class, SVM finds an optimal hyperplane that separates the two classes. By optimal hyperplane, we mean that the points belonging to either class are the farthest possible from the hyperplane. The points belonging to either class that is closest to the hyperplane are called support vectors. If you draw lines on either side of the hyperplane which contains the support vectors, then the distance between these two lines is called margin. SVM can also be considered as an algorithm that maximizes the margin between two classes.

The one implicit assumption in the above description of SVM is that the data set is linearly separable. When the data set is not linearly separable, SVM applies a clever trick which projects the data set into higher dimensions in which the data set is linearly separable; this trick is called kernel trick which was introduced in [22].

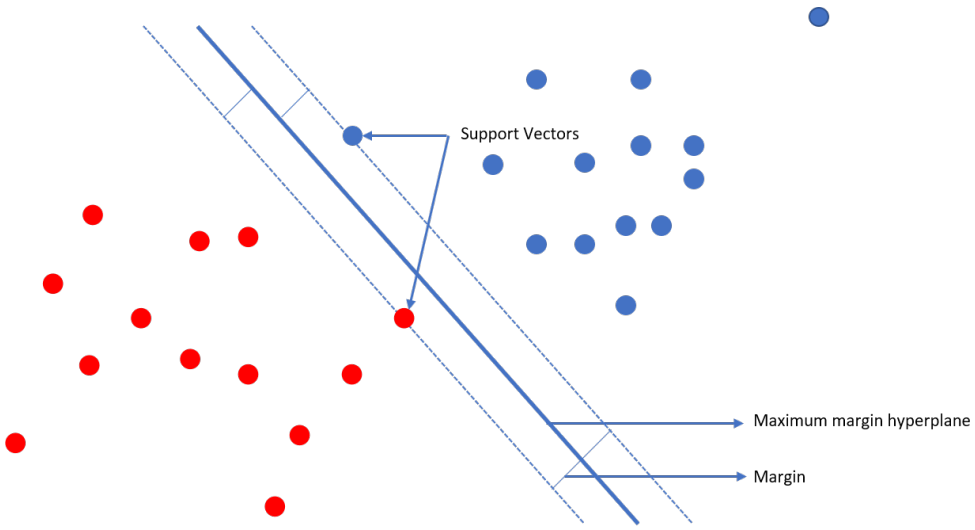

Fig. 5.1: Depiction of SVM hyperplane, margin and support vectors

Trying to find the hyperplane that separates both classes in a data set can be tricky and can result in sub-optimal solutions. For instance, when just a couple of data points are close to the hyperplane but others are relatively far away it would be a good decision to ignore them to maximize the margin. This would make the SVM classifier more general to test data or other unseen data. The hyperplane so obtained is called soft margin hyperplane. The number of data points or variables that can be misclassified/ignored to obtain a larger

margin hyperplane is called slack variables. Please refer to [22] to explore SVM in depth.

5.2 Sampling

Sampling, in the context of this thesis, refers to how data points are separated into test set and train set. We used two sampling methods: serial sampling and random sampling.

5.2.1 Serial Sampling

In serial sampling, the images are sorted according to their timestamps. At some timestamp, we split the data set into a test set and train set.

Serial sampling is crucial to test whether classifiers trained on the previous year's data set will work on the current year's data. It also tells us how similar is the current batch of the test set is compared to the previous batches of test sets or the train set.

MNIST data set cannot be serially sampled because the samples of the data set are not time-related.

5.2.2 Random Sampling

In random sampling, images are randomly sampled without replacement from the data set to form a train set and a test set. None of the attributes of the data point matter. However, we strive to maintain the ratio of each label in the test set and train set equal to that of the whole data set.

Random sampling is crucial to find out the best performance that can be obtained from a classifier given a data set with well-represented classes.

5.3 Over-fitting and under-fitting a classifier

All classification algorithms try to find a set of parameters that describe the given data set. Sometimes in the quest to achieve good results, the parameters of classifiers tend to describe the training data set so well that they do not generalize to the test data set. This is termed as over-fitting. The parameters describe the training data set too tightly for it to be applicable to the test data set. In other words, when the classification error for the

training set is lower than the testing set, the classifier is said to be over-fitted on the train set. Under-fitting of a classifier happens when the classification error of a classifier is lower on the test set than on the train set. Generally, machine learning practitioners try to attain the sweet spot where the training error is equal to the test error which is difficult.

5.4 Measuring Performance of Classifier

For a balanced data set, accuracy is enough to describe the performance of a classifier. However, when the data set is unbalanced, accuracy can be misleading. Since our data set has mostly unbalanced attributes, we use the following metrics based on a confusion matrix to measure and describe the performance of the machine learning algorithms that have been used in this thesis.

- Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

- Precision

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

- Recall

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

- F1 score

$$F1score = 2 * \frac{precision * recall}{precision + recall} \tag{5.4}$$

  where TP is True Positive, FP is False Positive, TN is True Negative, FN is False Negative

  True Positives and True Negatives occur when classes are properly assigned. False Positives and False Negatives imply wrong classes are assigned by the classifier.

Precision, recall, and f1 score can only be computed for binary attributes. For example, classifying image into digits will give you 10 classes from 0 to 9, so only accuracy can be computed to describe the classification performance. While classifying the weather as precipitating and non-precipitating you can have all the four metrics since the precipitation is a binary attribute.

None of the metrics alone describe all aspects of the classification performance that someone might be interested in. A combination of the above metrics helps with this.

5.5 Discussion and Results

Results of classification on two data set using different classification methods, and dimension reduction methods are discussed in this section.

5.5.1 Performance of classifiers on MNIST (Benchmark)

The accuracy of classifiers on MNIST was used as a general guideline to depict the improvement in the classifier performance as the number of components increase. PCA and RP were used as two different methods for dimension reduction. The classifiers used were SVM and kNN Classifier. The parameters of the classifiers used are listed below. The same parameters have been used for the classifiers throughout this thesis.

- kNN Classifier: k = 10, (distance) metric = euclidean, voting = majority.
- SVM Classifier: c = 10, gamma = scale.

Min_accuracy is the theoretical minimum accuracy obtained when an over-fitted classifier assigns every data point to the majority class. It is computed by dividing the number of samples in the majority class for an attribute by the total number of samples. Similarly, the label 'Accuracy_svm_pca' indicates the accuracy of the SVM classifier with PCA being used for dimension reduction. The other labels in the figures can be read similarly. Also, note that the last point in any curve is the performance of the classifier on the original data set.

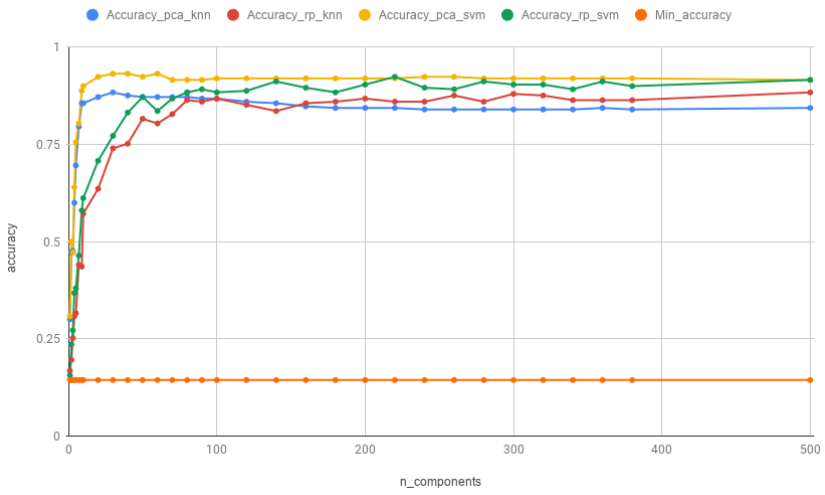

Fig. 5.2: Accuracy vs. n_component for SVM and kNN Classifier using RP and PCA as method of projection.

Of the two classifiers, SVM has higher accuracy for every n_component and both dimension reduction techniques. The random nature of Random Projection results in an erratic performance of classifiers. The following are the observations on using RP as the dimension reduction technique:

- Even though the overall performance of the classifiers increases on increasing n_component, this cannot be ensured for every consecutive n_component.
- The classifiers trained on the original data set outperform the classifiers on the reduced data set for every n_component. The trend line for accuracy or f1 score for classifiers using RP increases on increasing n_component, and the best performance was obtained on the original data set.

The usage of PCA as a method of projection is satisfying. At some n_component as shown in fig. 5.2 the projected data set outperforms the original data set.

5.5.2 Performance of classifiers on Spring Garden data set (Random Sampling)

The description of the data set used is as follows:

- Number of samples: 1600

- Train to test split: 75 to 25
- Image Preprocessing: None
- Color space: Grayscale
- Sampling: Random

The classifiers were trained separately on the following attributes:

- daynight: This attribute tells us whether the image was captured during nighttime or daytime. The time duration between sunset to sunrise is defined as nighttime and the rest is defined as daytime. The classifiers achieve pretty good performance even at very small n_component preserved. This attribute is nearly balanced as the f1 score and accuracy together suggest. Most of the classification error occurs during the transitions from day to night or night to day. On removing transition images - two each for day to night and night to day transition, the classification accuracy increased an average of 0.3% to 1.2%.

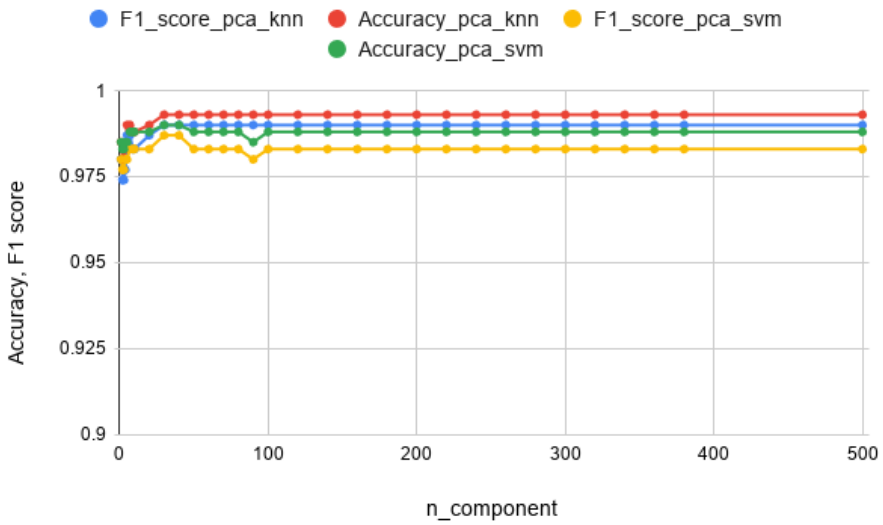

Fig. 5.3: Accuracy, F1_score vs. n_component for SVM and kNN Classifier using PCA as method of projection to classify daynight.

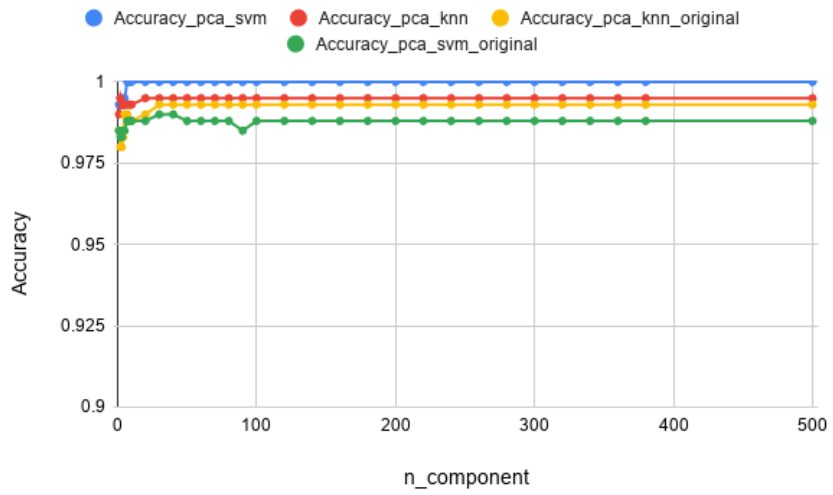

Fig. 5.4: Accuracy vs. n_component for SVM and kNN Classifier using PCA as method of projection to classify daynight after removing transition images. _original refers to classifiers trained on original data set and its absence refers to classifiers trained on data set with 4 less transitioning images for each day.

- precipitation: This attribute indicates whether there was precipitation or not when the image was captured. It is 1 if there was precipitation when the image was captured else it is 0. It is tricky to correctly classify precipitation in images because it is difficult to tell apart images where it is currently precipitating versus image where it precipitated half an hour ago even for a human participant. The classification performance on this attribute is shown in Figure 5.5

- precipitation1hr: This attribute indicates whether there was precipitation in the last one hour when the image was captured. It is 1 if there was precipitation in the last one hour when the image was captured else it is 0. It is more balanced than surfaceStatus and precipitation. The classification performance on this attribute is shown in Figure 5.6

- surfaceStatus: This attribute gives us information about the status of a road surface like whether the road is dry or wet or has a presence of moisture, etc. It is a multi-valued attribute with 14 classes from 1 to 14. The names of the classes are other, error, dry, trace moisture, wet, chemically wet, ice warning, ice watch, snow

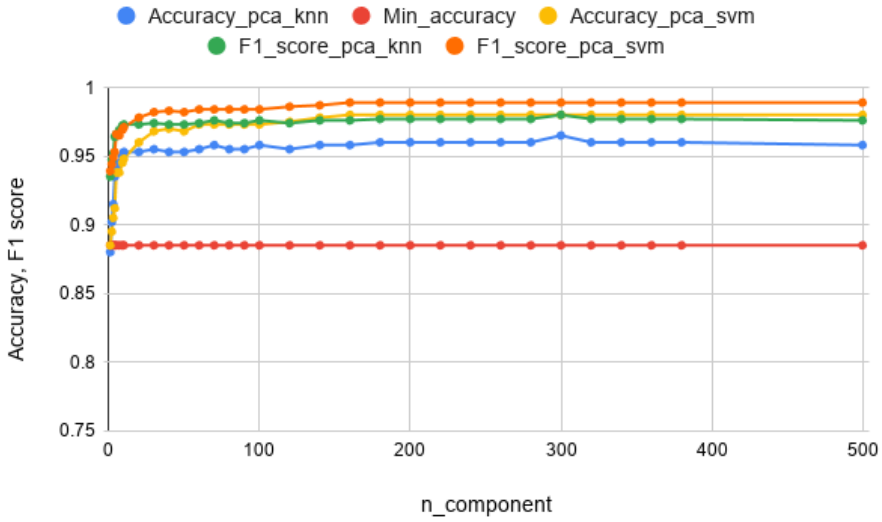

Fig. 5.5: Accuracy, F1_score vs. n_component for SVM and kNN Classifier using RP and PCA as method of projection to classify precipitation.

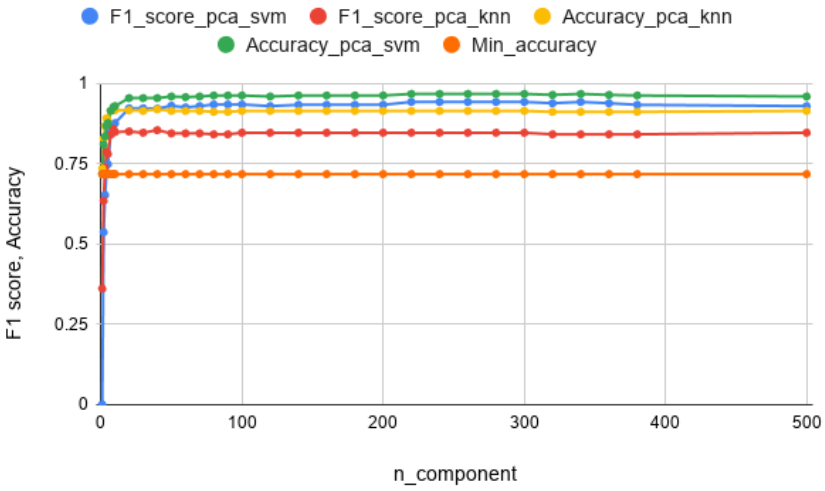

Fig. 5.6: Accuracy, F1_score vs. n_component for SVM and kNN Classifier using RP and PCA as method of projection to classify precipitation1hr.

warning, snow watch, absorption, dew, frost, and absorption at dew point. Since it is multi-valued, a confusion matrix cannot be computed for the attribute, and thus precision, recall, and f1 score cannot be computed. The classification performance on this attribute is shown in Figure 5.7.

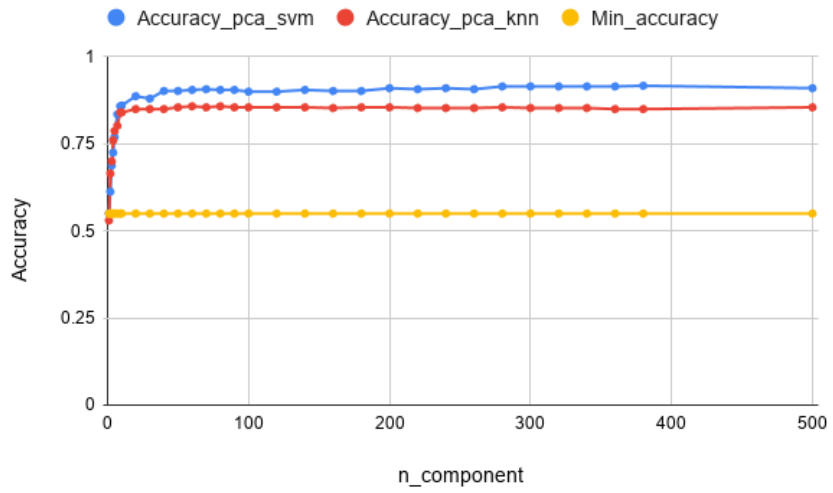

Fig. 5.7: Accuracy vs. n_component for SVM and kNN Classifier using RP and PCA as method of projection to classify surfaceStatus.

5.5.3 Do the classifiers generalize well to other data sets/locations?

All the data sets including MNIST and Spring Garden whose results have been reported and Black Butte which has been relegated to the appendix A.1 (for being largely similar to other results) were run using the same classification algorithm with the same parameters. They performed as well as those reported in this chapter. This shows that the algorithms and metrics generalize to other locations and data set well.

5.6 Performance of classifiers on serially sampled Spring Garden data set

The essence of measuring classifier performance on serially sampled train and test sets

is to evaluate if the classifiers trained on the existing data set will be useful on images that will be captured in the future.

As the figures 5.8, 5.9 and 5.10 show, the classifier performance on serially sampled data set is not promising enough. The obtained performance is barely above minimum accuracy for classification of precipitation1hr and surfaceStatus and lower than minimum accuracy for classification of the precipitation attribute. The minimum accuracy is computed as the ratio of the number of samples in the majority class to the total number of samples in the data set. Also, the f1 score is relatively low compared to results obtained using random sampling which hints that the trained classifier model is probably over-fitted. It simply tells us that the train set was not representative enough of all the classes of the attributes that are to be classified. The two ways to make the data set representative enough are to obtain more data or reduce the variability between test set with the training set. The disappointing performance of image processing techniques to reduce variability between images implies that only more data is the answer to this problem. In Figure A.7 we demonstrated that adding more data to the train set reduces the variability between images/data points.

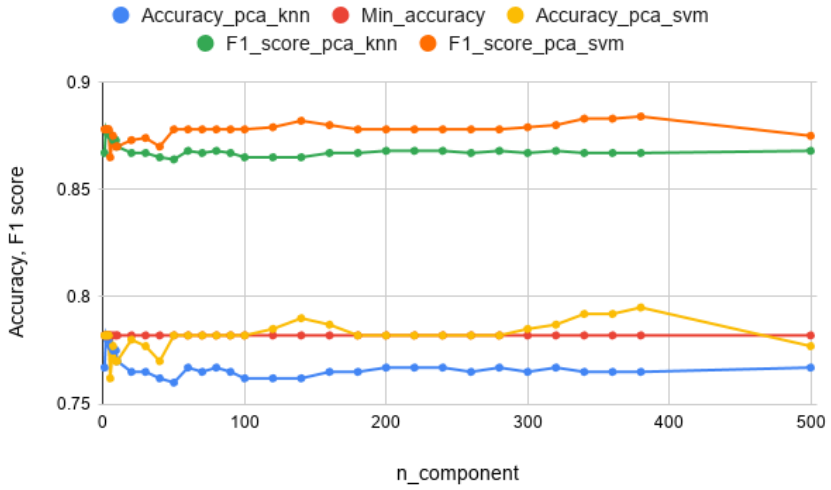

Fig. 5.8: Accuracy, F1_score vs. n_component for SVM and kNN Classifier using PCA as method of projection to classify precipitation.

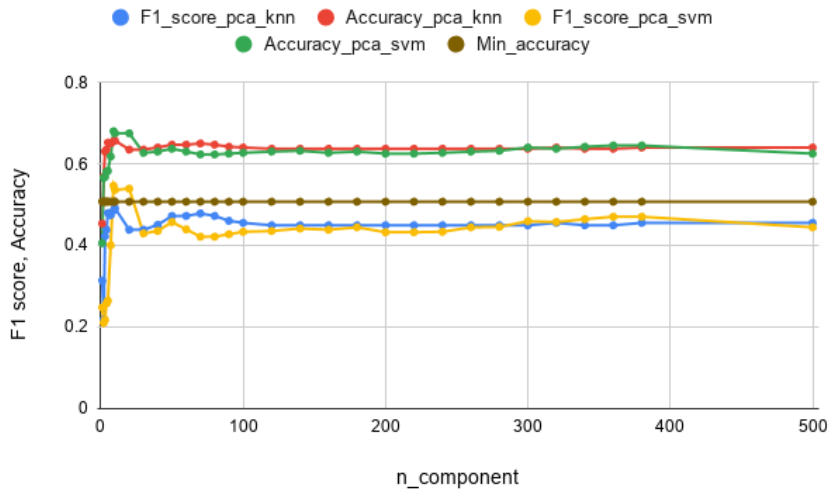

Fig. 5.9: Accuracy, F1_score vs. n_component for SVM and kNN Classifier using PCA as method of projection to classify precipitation1hr.

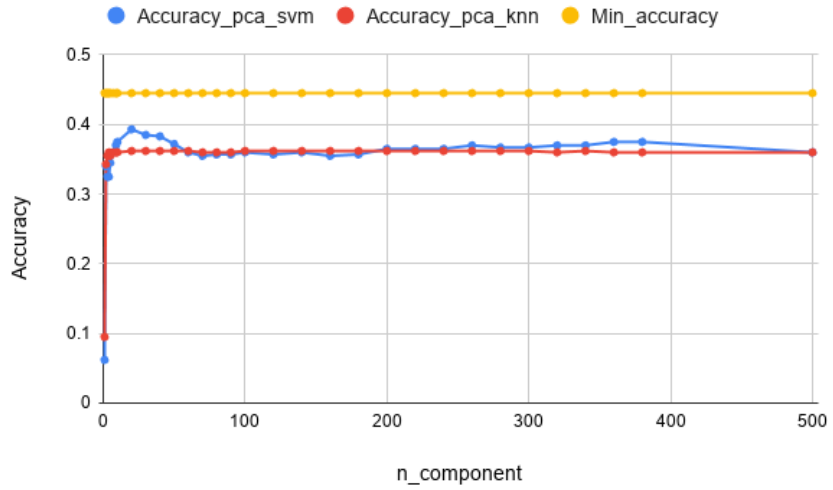

Fig. 5.10: Accuracy, F1_score vs. n_component for SVM and kNN Classifier using PCA as method of projection to classify surfaceStatus.

5.7 Performance of classifiers using same projection parameters for subsequent batches of data

Consider the scenario where we obtain transformation parameters for a dimension reduction method like PCA, RP, or Image Scaling from one batch of data. It would be highly desirable to apply the same linear/nonlinear transformation on the consecutive batch of data before training a classifier or classifying the data points using existing trained classifier models. This allows us to not have to store batches of the data set. All that is required to be stored is the set of projection parameters extracted from the first batch of data set and the transformed values of subsequent batches. This assumes that the consecutive batch of the data set is similar to the train set (first batch of data). Otherwise, the trained classifier would perform poorly on the projected consecutive data set for two reasons:

- Projection parameters obtained from the original/first batch of data set do not describe well the subsequent batch of data set.
- The classification model is not trained or has not seen the similar instances of data points it is having difficulty to classify.

It works for methods like Random Projection which does not depend on the data set at all except for the number of features, and Image Scaling which treats each image individually. PCA, on the other hand, tries to extract features from the whole of the data set. The argument here is that even methods like PCA would work if the data on which projection parameters were extracted are fairly representative.

The generalization of classifier performance over batches of a data set can be measured by plotting classification performance on several batches of a data set. If the performance decreases, the first batch of the data set was not representative enough of the subsequent batches of the data set. Metrics for quality of projection can be plotted sequentially for each batch of data set. Again, if the quality of projection decreases for subsequent batches of the data set, the first batch of data set from which the projection parameters were obtained were not representative enough.

To obtain the results, we first train a classifier on a train set and also obtain projection parameters. Each of the consecutive batches of data is projected using the projection parameters obtained from the train set and is then classified using the classifier trained on the train set. The results on daynight attribute of the Spring Garden data set are shown in Figure 5.11. We varied the batch size for the first batch of data and the rest of the batches and found that the larger the first train set the better the performance on the consecutive batch of data sets.

In the results obtained in Figure 5.11, the batch size of the data sets (second through last) is fixed to 400 samples. We vary the batch size of the first batch (or train set) from 400 to 2000 in the increment of 400. As the batch size of the first batch increases, the performance increment is stark and significant. Also, PCA was used as the method of projection and SVM was used as the classifier.

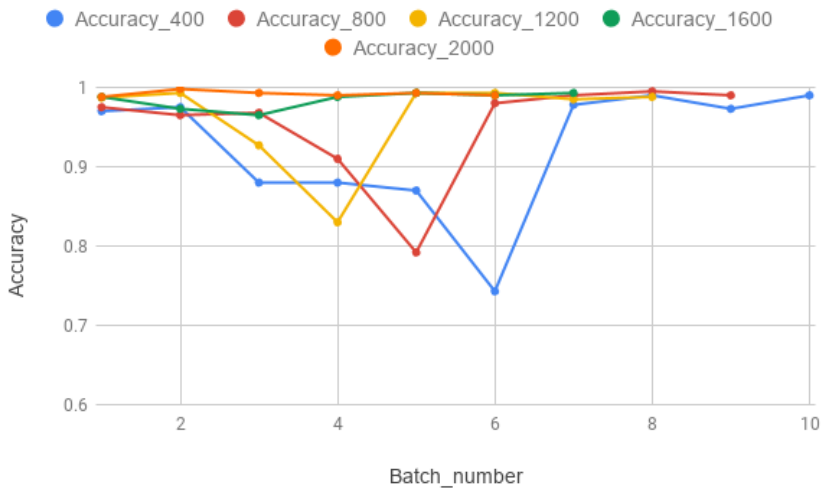

Fig. 5.11: Performance of SVM trained on first batch of data and using same parameters of projection for subsequent batch of test data

5.7.1 Why small n_component values give us good performance?

While it might seem a that small number of components preserved (n_component) gives us a good performance, we have to see the bigger picture to choose the amount of dimension

reduction. It just might the case of overfitting or unbalanced attribute. Spring Garden data set has mostly unbalanced attributes or attributes with binary classes where at least an accuracy of 50% can be expected for even a random classifier. We computed minimum accuracy as the ratio of the number of samples in the majority class to the total number of samples in the data set for all attributes. The minimum accuracy for the attributes 'precipitation', 'precipitation1hr', 'surfaceStatus', and 'daynight' are 0.885, 0.718, 0.55 and 0.62 respectively. Also, looking at the f1 score might suggest that at small n_component, classifiers are good at classifying the majority class while classifying other classes poorly.

It also might be that the problem statement is really easy in which it is easy to observe good classification performance even for small n_component preserved. For instance, the classification of daynight attribute is a relatively easy problem statement. It would not be surprising for us to have good performance at small n_component.

5.7.2 Comparison of the Dimension Reduction methods on the MNIST data set

We plotted metrics measuring quality of projection for PCA, RP, and Image Scaling against the accuracy of classifiers (SVM and kNN Classifier) to explain the impact of dimension reduction on the performance of classifiers.

The metrics for quality of projection do not measure the classifier performance, rather they provide clues as to what we can expect from the classifier, and when we can expect peak performance from the classifier. They help to estimate relative performance of the classifier as Maaten et. al [1] argue that for successful classification of data its structure needs to be retained and these metrics measure how well the structure of the data is preserved.

A high value of any metric measuring quality of projection does not indicate that the classifier will a have good performance on the reduced data set. Rather, it indicates that the classifier will have relatively good performance.

Both the classification performance and quality of projection metrics were computed on the same MNIST data set of 1000 randomly selected images. The dimensions of the image are 28*28. The number of neighbors considered (k) is 10 for computing quality of

projection metrics as well as for the kNN classifier. The minimum accuracy defined is the ratio of the number of samples in the majority class to the total number of samples is 12.4%.

The accuracy of the classifier trained on a data set reduced using PCA can be noted to increase steadily until it peaks and then starts declining. It can be noted that n_component increases from left to right in the graphs below.

Trustworthiness and Continuity are the best predictors of classification performance among all the quality of projection metrics discussed in this thesis. It is evident in the figures below that when the quality of projection measured by Trustworthiness and Continuity increase so does the accuracy of the classifiers (SVM and kNN Classifier). For the same value of projection quality, Image Scaling has the lowest accuracy. As evident in the prior section, Image Scaling requires the largest number of components to achieve similar projection quality. Thus for a specific n_component, Image Scaling has the lowest accuracy and projection quality.

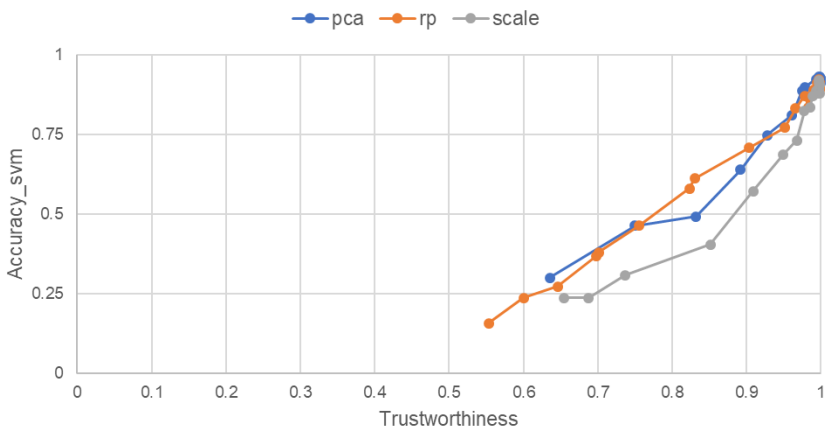

Fig. 5.12: Accuracy versus Trustworthiness for SVM classifier

Distortion1 and Distortion2 are pretty good estimators of accuracy. For Distortion2, increase in quality of projection corresponds to increase in accuracy until they both peak or plateau. As the n_component increases, the decrease in distortion and increase in accuracy is the fastest for PCA and the slowest for Image Scaling (n_component increases from right to left in this case).

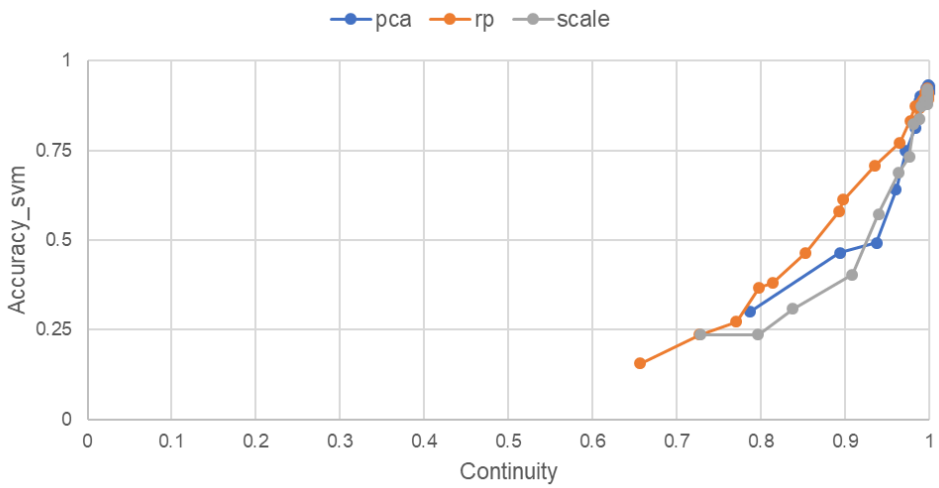

Fig. 5.13: Accuracy versus Continuity for SVM classifier

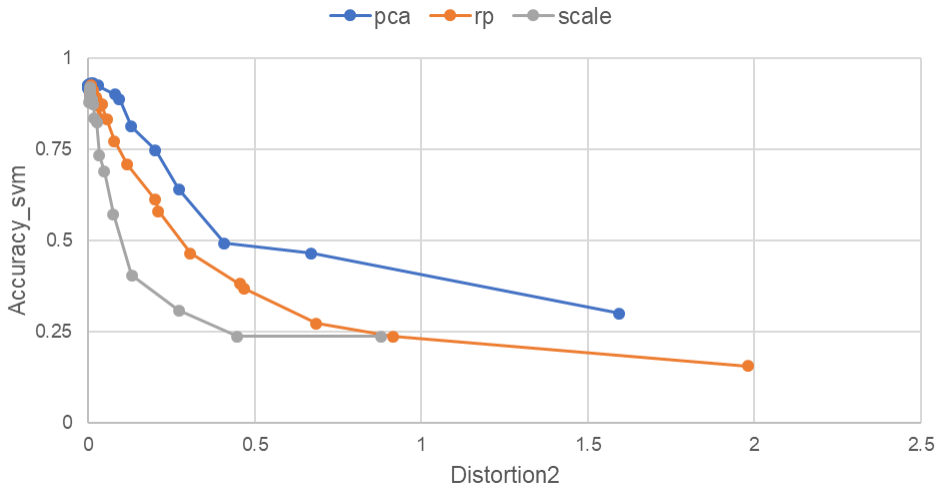

Fig. 5.14: Accuracy versus Distortion2

The accuracy of the classifiers increases as the Ratio Preserved increases. For PCA, once it achieves the peak accuracy its accuracy does not increase even on significant increment in Ratio Preserved while the accuracy of RP and Image Scaling increases slowly for smaller gains in quality of projection until both accuracy and quality of projection plateaus. PCA achieves a higher quality of projection and accuracy than RP and Image Scaling.

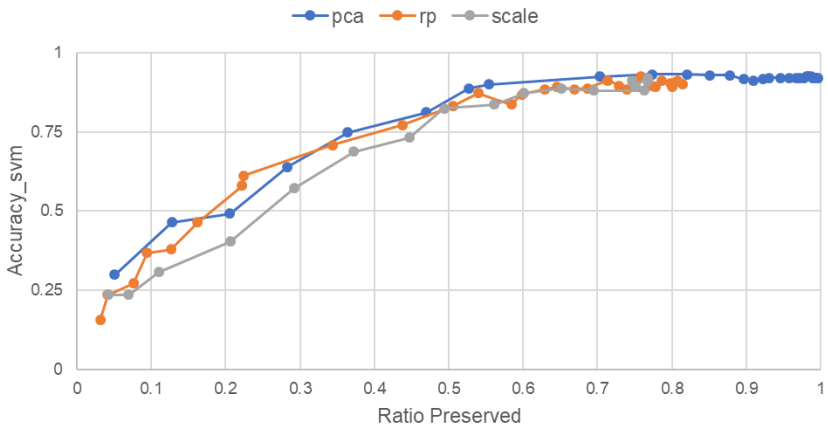

Fig. 5.15: Accuracy versus Ratio Preserved

Spearman Variant behaves similar to Ratio Preserved while also accounting for the relative order of neighbors preserved.

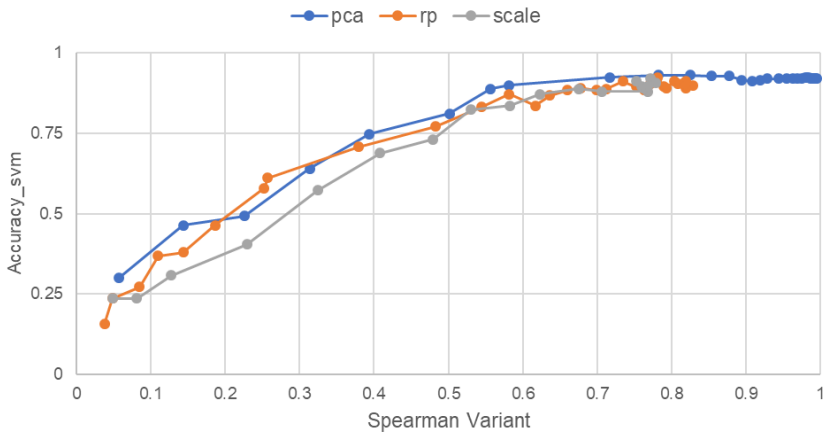

Fig. 5.16: Accuracy versus Spearman Variant

Spearman is the strictest quality of projection measure among all the metrics that have been used in this thesis. It estimates the degree to which the relative ordering of neighbors of each data point is preserved. The graph below suggests that preserving the relative ordering of the nearest neighbors is a challenging task for a smaller size of reduced data set and is not necessary to achieve a good classification performance. It can also be noted that even as Ratio Preserved increases, Spearman may not increase for smaller n_component. After the first few n_components, even as the projection quality increases, the accuracy does not increase and stays relatively unchanged. PCA achieves both the best quality of projection and accuracy.

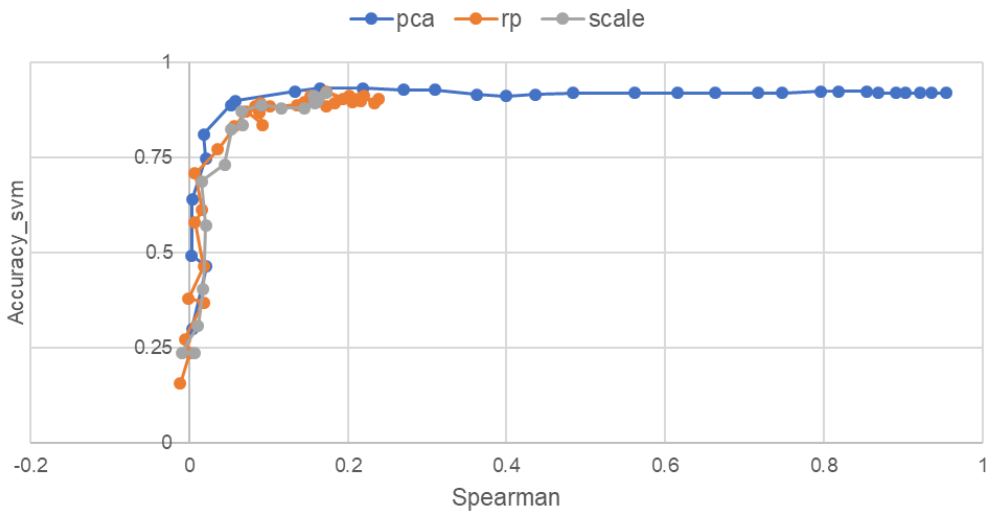

Fig. 5.17: Accuracy versus Spearman

Like Trustworthiness and Continuity, Top k Normalized Distance also assumes the worst for the missing neighbors. For the first missing neighbor, it assumes that it ranks the last when all of the neighbors are considered and the second missing neighbor is second to the last when all of the neighbors are considered and so on. For very small n_component, i.e. n_component << num_features, the change in projection quality does not correspond to change in accuracy. PCA achieves both the best quality of projection and accuracy.

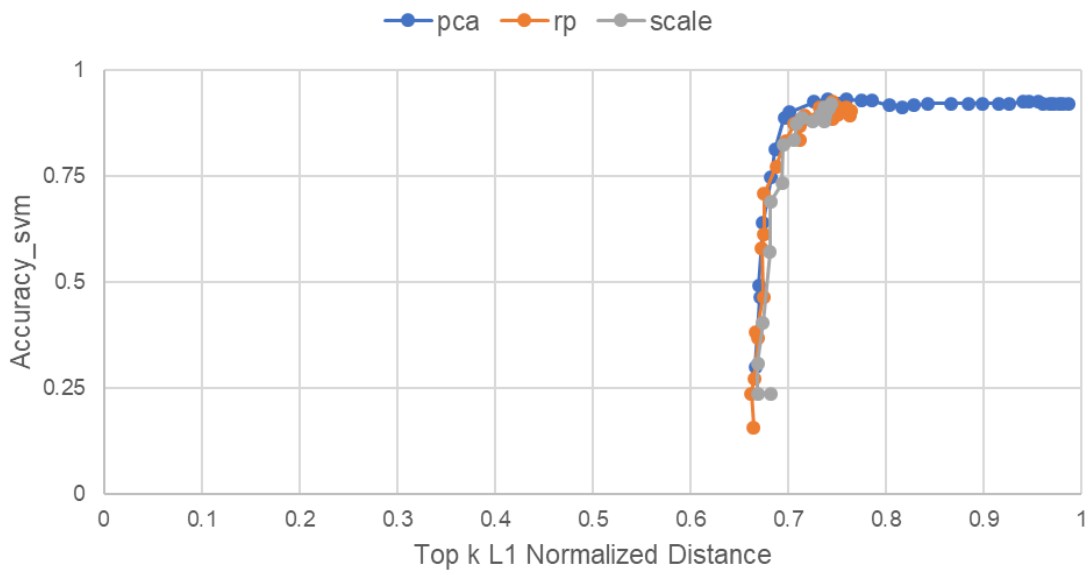

Fig. 5.18: Accuracy versus Top k L1 Normalized Distance

5.7.3 Comparison of the Dimension Reduction methods on Spring Garden data set

Both the classification performance and quality of projection metrics were computed on the Spring Garden data set. The classification used 1600 images while quality of projection used only 1000 randomly selected images due to the computational cost of computing values of multiple metrics. The number of neighbors considered (k) is 10 for computing quality of projection metrics as well as for the kNN classifier. Unlike MNIST in which the 'digit' attribute has ten classes and the minimum accuracy was 12.4%, the 'precipitation1hr' attribute on which these tests are carried out is an unbalanced binary attribute with a minimum accuracy of 71.8% because of which even at n_components < 10, the accuracy is really high. The rest of the discussion adheres to the MNIST data set's discussion.

Please note that n_component increases from left to right in the graphs below.

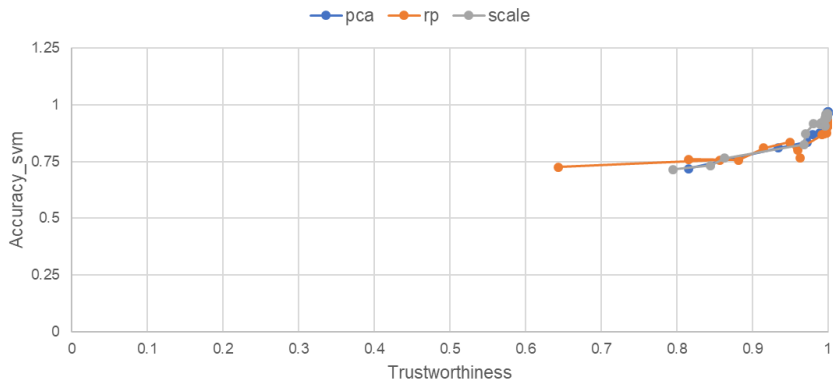

Fig. 5.19: Accuracy versus Trustworthiness for SVM classifier

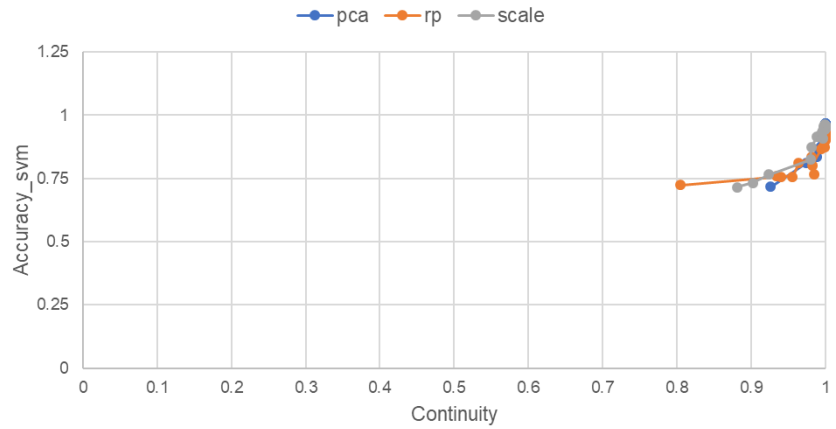

Fig. 5.20: Accuracy versus Continuity for SVM classifier

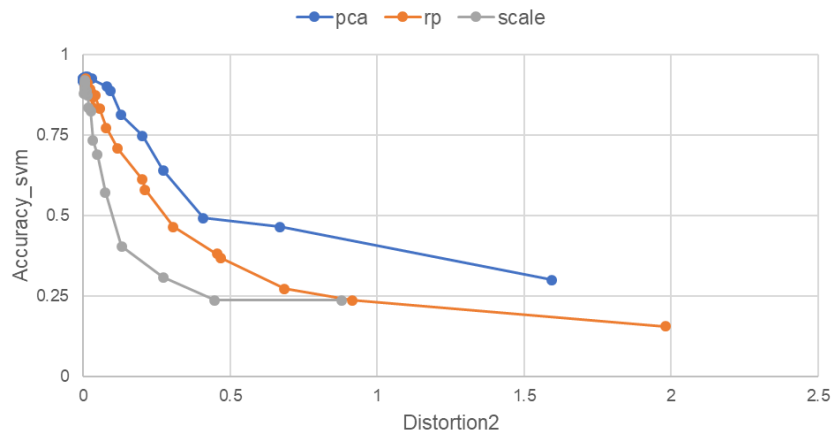

Fig. 5.21: Accuracy versus Distortion2 (n_component increases from right to left in this graph)

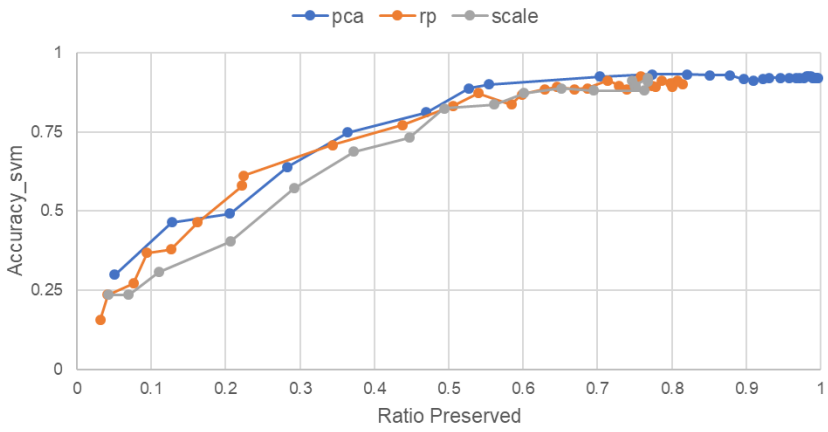

Fig. 5.22: Accuracy versus Ratio Preserved

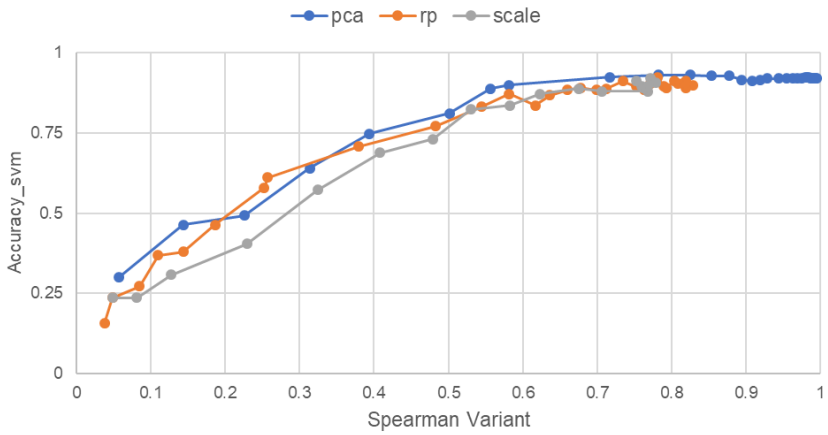

Fig. 5.23: Accuracy versus Spearman Variant

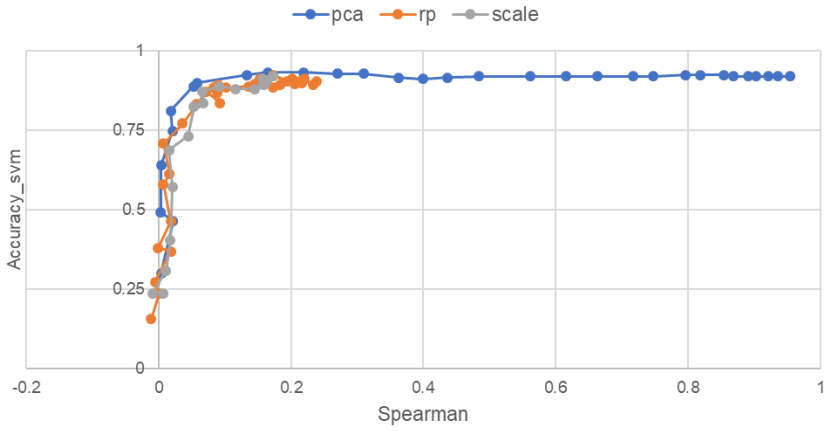

Fig. 5.24: Accuracy versus Spearman

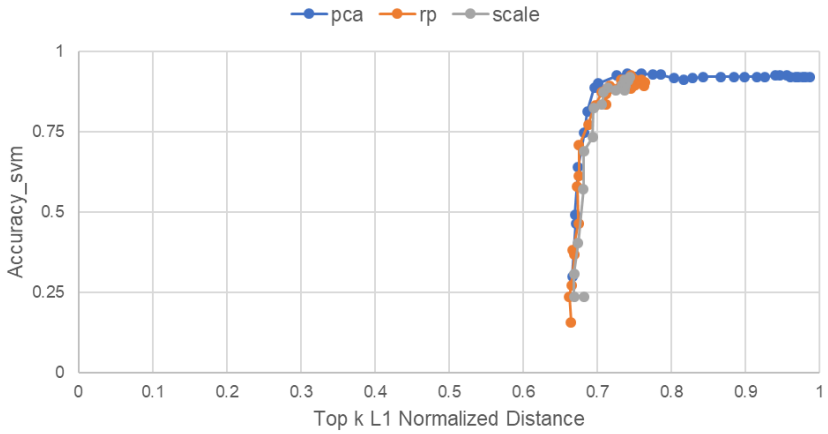

Fig. 5.25: Accuracy versus Top k L1 Normalized Distance

5.8 Conclusion

Here are some of the notable conclusions from this chapter.

- SVM outperformed kNN Classifier except for the 'daynight' attribute. The best classification performances were obtained using PCA for dimension reduction.
- While classifiers trained on the data set reduced using PCA performs better than the data set reduced using RP or Scaling, its performance plateaus at around 200 number of components preserved. At its peak, a classifier trained on data set reduced using PCA outperforms the original data set indicating the presence of noise in the original data set.
- The performance of classifiers trained on serially sampled data set is really poor indicating that the serially sampled data set is not representative of the test set or the whole data set. Figure 5.11 suggests that adding more samples to the data set might help to improve the performance of the classifier on serially a sampled data set.
- Classifiers trained on data sets reduced by Image Scaling have the lowest value of accuracy for any value of quality of projection measured by any of the metrics estimating quality of projection. Classifiers trained on data sets reduced by PCA have the highest value of accuracy for any values of quality of projection.

- Qualitatively, Trustworthiness, Continuity, Distortion1, and Distortion2 are good estimators of the accuracy of classifier while the rest of the metrics namely Spearman, Spearman Variant, Ratio Preserved, and Top k Normalized Distance are decent predictors of the accuracy of classifiers.

- Even though the aim of this thesis is not to build classifiers, we were able to train pretty good classifiers on all of the attributes that we explored.

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

In Chapter 3, 'Dimension Reduction', we introduced several parameters to measure quality of projection and used some existing metrics as well. We also used some statistical tools to measure quality of projection like the Spearman Rank Correlation Coefficient. Spearman estimates the preservation of the relative order of the neighbors of each data point. We also addressed its shortcomings as a metric for quality of projection and introduced Spearman Variant. We rigorously tested Image Scaling which is not as established as PCA and Random Projection for dimension reduction. Image Scaling required the highest dimension of reduced dataset for equivalent quality of projection.

In Chapter 4, 'Image Preprocessing', it was found that single-channel images performed as well or better than full channel images which is quite surprising given that full channel images are three times the size of the single-channel images. Of all the single channels, we preferred grayscale because it performs at par or better than other single-channel images including histogram equalized images and channels of other color spaces like HSV and is more interpretable than all of them.

In Chapter 5, 'Image Classification', we found that of the SVM and kNN Classifier, SVM almost always outperformed the kNN Classifier without tuning for individual attributes, locations, or colorspace. Even though it is not the aim of this thesis, the classifiers trained on randomly sampled dataset performed pretty well for all of the attributes. It is commendable since the problem of classifying weather or road status from images is difficult and the dataset is highly unbalanced.

One of the interesting results from this chapter is that if a classifier is trained on dimension reduced train set which is fairly representative of the dataset, then the consecutive

batch of test sets can be reduced using the same transformation parameters that were obtained from the first batch of a dataset without little compromise in classifier performance. This also implies that the consecutive batches suffer a small loss in quality of projection even if the transformation parameters were not computed from them.

We also found out that quality of projection metrics like Trustworthiness, Continuity, Distortion1, Distortion2, Ratio Preserved, and Spearman Variant are decent predictors of the accuracy of classifiers qualitatively.

6.2 Future Work

The essence of research is that it leaves you with more questions than you started with. This thesis is no exception as it leaves me with multiple questions, some more exciting than others. For instance, can we adjust the transformation parameters obtained after fitting PCA (or other dimension reduction method) to a dataset so that it fits the next batch of the dataset better?

Another more interesting topic of research is to quantify the number of components that need to be preserved for preserving distance between data points. In this thesis, only qualitative assessments were performed. Also, it would be useful to be able to compute the equivalent number of components required by different dimension reduction methods for a given projection quality metric.

Some more obvious things to do would be to extend the results of this thesis to other projection methods like Non-Negative Matrix Factorization (NMF), Singular Value Decomposition (SVD), Image patch sampling, etc. We only explored SVM and kNN Classifier in this thesis. It would not be difficult to extend it to other classifiers like Decision Trees, Random Forest, Logistic Regression, etc.

REFERENCES

[1] L. van der Maaten, E. Postma, and J. van den Herik. (2009) Dimensionality reduction: A comparative review. [Online]. Available: https://lvdmaaten.github.io/publications/papers/TR_Dimensionality_Reduction_Review_2009.pdf

[2] A. Gracia, S. González, V. Robles, and E. Menasalvas. (2014) A methodology to compare dimensionality reduction algorithms in terms of loss of quality. [Online]. Available: https://core.ac.uk/download/pdf/148668147.pdf

[3] W. Wang and M. A. Carreira-Perpinan, "The role of dimensionality reduction in classication," 2014.

[4] S. Keller, A. Braun, S. Hinz, and M. Weinmann, "Investigation of the impact of dimensionality reduction and feature selection on the classification of hyperspectral enmap data," 2016, pp. 1–5.

[5] Z. Jelena, O. Kurasova, and M. Liutvinavičius, "Dimensionality reduction methods: The comparison of speed and accuracy," *Information Technology And Control*, vol. 47, 03 2018.

[6] J. Nikkilä, P. Törönen, S. Kaski, J. Venna, E. Castrén, and G. Wong. (2014) Analysis and visualization of gene expression data using self-organizing maps. [Online]. Available: https://core.ac.uk/download/pdf/148668147.pdf

[7] H. S. Obaid, S. A. Dheyab, and S. S. Sabry, "The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning," Oct.

[8] Rwis field short description. [Online]. Available: http://cwwp2.dot.ca.gov/documentation/rwis/rwis-field-description.htm

[9] J. P. Allebach. (2005) Nearest neighbor interpolation. [Online]. Available: https://www.sciencedirect.com/topics/engineering/nearest-neighbor-interpolation

[10] K. Pearson, "On lines and planes of closest fit to systems of points in space," in *Philosophical Magazine*, 1901, pp. 559–572.

[11] J. Shlens. (2014, 127003) A tutorial on principal component analysis. [Online]. Available: https://arxiv.org/pdf/1404.1100.pdf

[12] A. N. Bhagoji, C. Sitawarin, and P. Mittal, "Enhancing robustness of machine learning systems via data transformations," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, 2018, pp. 1–5.

[13] W. B. Johnson, J. Lindenstrauss, and G. Schechtman, "Extensions of lipschitz maps into banach spaces," 1986.

[14] Sparse random projection. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.random_projection.SparseRandomProjection.html

[15] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," 2006.

[16] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Prentice Hall, 2012, ch. 2.

[17] J. Venna and S. Kaski, "Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity," 2005, pp. 55–61.

[18] W. W. Daniel, "Spearman rank correlation coefficient," 1990, pp. 358–365.

[19] Y. LeCun, C. Cortes, and C. J. Burges. (1998) The mnist database of handwritten digits. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[20] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 1992, ch. 3, pp. 88–91.

[21] J. C. V. Guerra, Z. Khanam, S. Ehsan, R. Stolkin, and McDonald-Maier, "Weather classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of convolutional neural networks," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 2018, pp. 305–310.

[22] C. Cortes and V. N. Vapnik, "Support-vector networks," in *Machine Learn 20*, Oct. 1995, pp. 237–297.

APPENDICES

APPENDIX A

Classification Appendix

A.1 Classifier performance on dataset of Black Butte

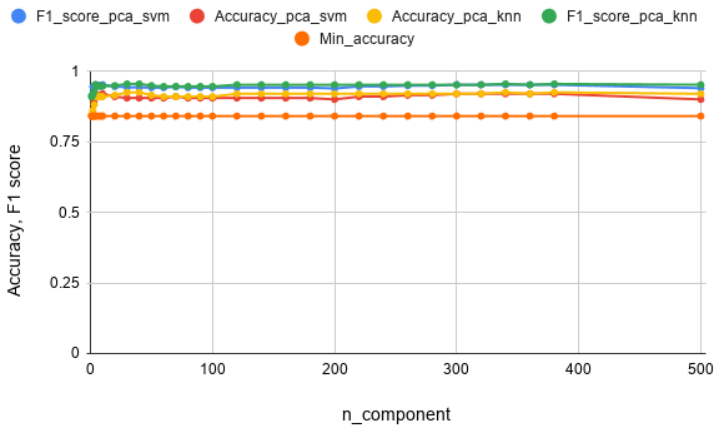

Fig. A.1: Accuracy vs. n_component for SVM and kNN Classifier classifying precipitation using PCA for dimension reduction

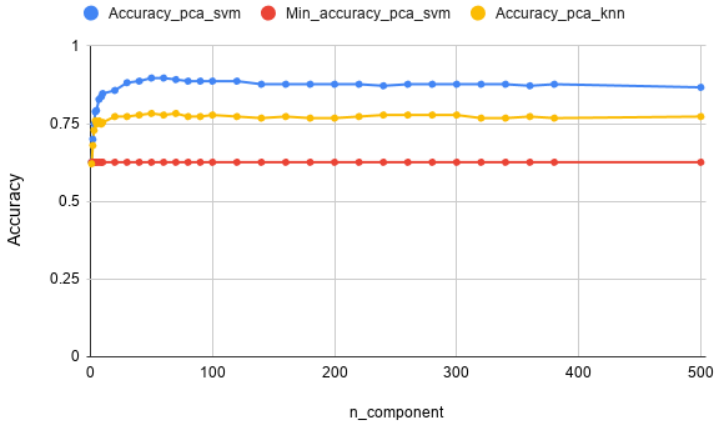

Fig. A.2: Accuracy vs. n_component for SVM and kNN Classifier classifying surfaceStatus using PCA for dimension reduction

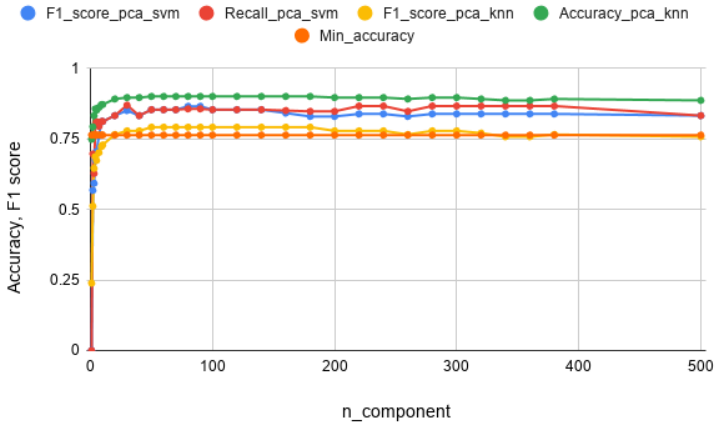

Fig. A.3: Accuracy vs. n_component for SVM and kNN Classifier classifying precipitation1hr using PCA for dimension reduction

A.2 Classifier performance on Spring Garden dataset reduced using Scaling

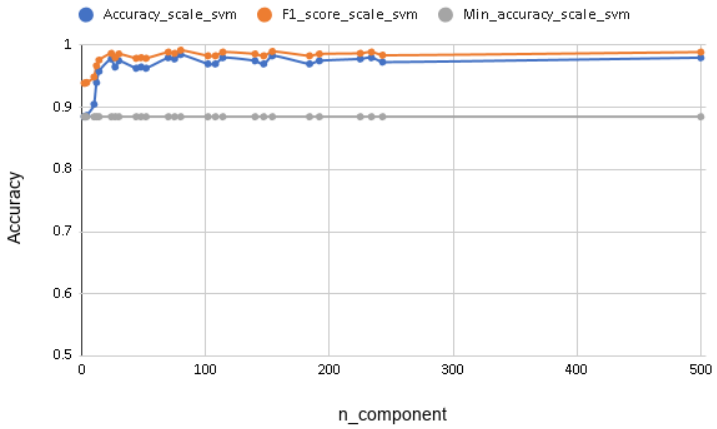

Fig. A.4: Accuracy, F1 score versus n_component for SVM using Scaling as method of projection for the classification of precipitation.

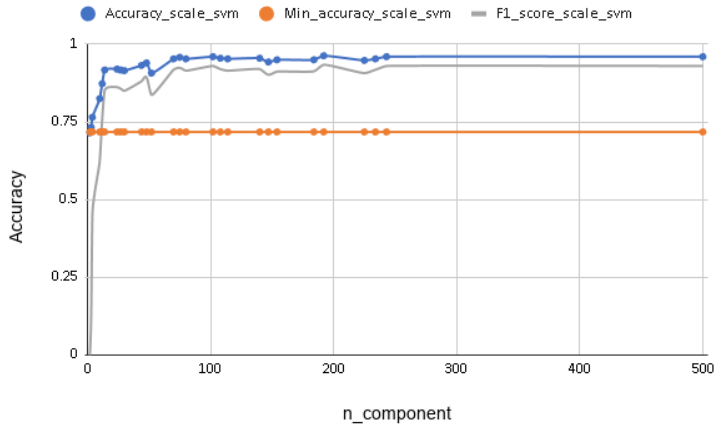

Fig. A.5: Accuracy, F1 score versus n_component for SVM using Scaling as method of projection for the classification of precipitation1hr.

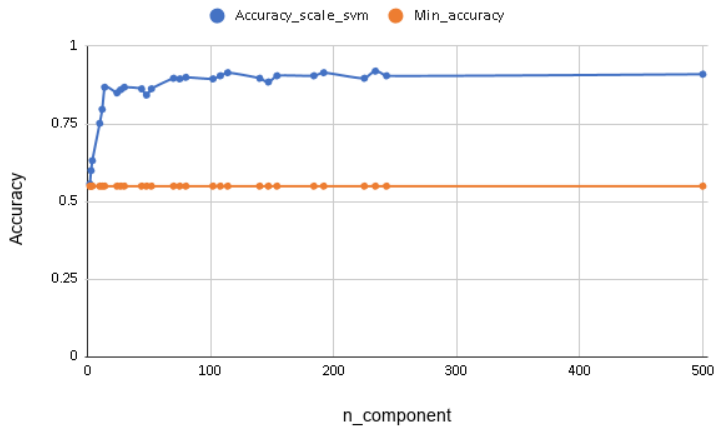

Fig. A.6: Accuracy score versus n_component for SVM using Scaling as method of projection for the classification of surfaceStatus.

A.3 Performance of classifiers using same projection parameters for subsequent batches of data

In the results obtained in Figure A.7, the batch size of the data sets (second through last) is fixed to 5000 samples. We vary the batch size of the first batch (or train set) from 5000 to 2000. As the batch size of the first batch increases, the performance increment in stark and significant. Also, PCA was used as the method of projection and SVM was used as the classifier.

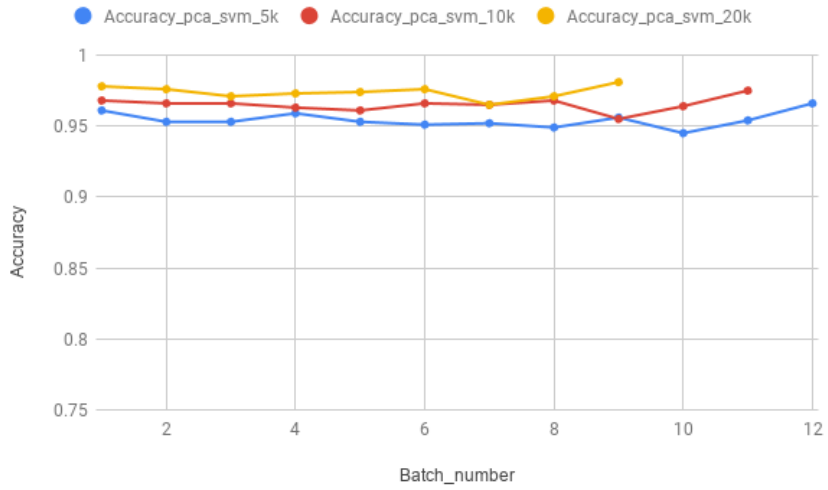

Fig. A.7: Performance of SVM trained on first batch of data and using same parameters of projection for subsequent batch of test data