A NEW APPROACH TO DOMAIN ADAPTATION APPLIED TO SUPERNOVA PHOTOMETRIC CLASSIFICATION

A Thesis Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Renuka Pampana

May 2016

A NEW APPROACH TO DOMAIN ADAPTATION APPLIED TO SUPERNOVA PHOTOMETRIC CLASSIFICATION

_____

Renuka Pampana

APPROVED:

_____

Dr. Ricardo Vilalta
Department of Computer Science
University of Houston

_____

Dr. Shishir Shah
Department of Computer Science
University of Houston

_____

Dr. Emile Ishida
Laboratoire de Physique Corpusculaire
Blaise Pascal University, France

_____

Dean, College of Natural Sciences and Mathematics

Acknowledgement

I would like to show my deepest gratitude for my advisor, Dr. Vilalta, for his continuous guidance, encouragement and support throughout this research project. I am truly grateful to him for his lectures in machine learning and artificial intelligence that led me to understand the problem effectively and to come up with creative solutions. It has been a dream of mine since childhood to be a part of the field of astronomy. With the help of Dr. Vilalta I now made my dream come true. He is the reason; I am able to contribute to the field of astronomy through computer science and machine learning. I am also grateful to him for giving me the opportunity to be a part of Pattern Analysis Lab and providing me with this project idea. I would also like to thank him for his unending patience and for his faith in me throughout my research work. Without his care and understanding it wouldn't have been possible.

I would also like to thank my colleagues from Pattern Analysis Lab for their immense support and encouragement especially my lab mate, Kinjal Dhar Gupta for his assistance and belief in my ability to accomplish this goal. Additionally, I would like to thank all my friends who have supported me in both happy and adverse conditions.

Last, but not the least, I would like to thank my parents and family to believe in my dreams and supporting my quest for higher education. Without their love and support, it wouldn't have been possible to experience the amount of success that I have.

A NEW APPROACH TO DOMAIN ADAPTATION APPLIED TO SUPERNOVA PHOTOMETRIC CLASSIFICATION

An Abstract of a Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Renuka Pampana

May 2016

Abstract

Supernova Type Ia plays a vital role in the measurement of the cosmological parameters. It is used as 'standard candles' for measuring extragalactic distances. There are other types of supernovae like Supernova Type Ib and Ic that closely resemble Supernova Type Ia (but are not as useful as Supernova Type Ia). Large telescopic surveys capture light curves of these supernovae events referred as photometric observations, which include all the three types. Thus, accurate classification of supernovae from these photometric observations is desirable for proper calculation of cosmological parameters.

The existing method for classification of supernova photometric observations is based on spectroscopic method, which is very cumbersome and expensive. In the future, with the increase in photometric surveys, myriad number of supernova photometric observations is expected. Thus, an efficient method for the classification of supernovae is required to replace existing methods. We also want to take advantage of existing dataset classified by spectroscopic method for the classification of upcoming photometric dataset. Since, these two datasets belong to different domains, an adaptive mechanism across the domains is required. Thus, we propose a method to generate a predictive model using domain adaptation with active learning that will classify supernovae (Ia, Ib, Ic) using spectroscopic data (aka source data) as a training set and photometric data (aka target data) as a testing set. Our method includes two concepts of machine learning: 1. Domain adaptation technique is used to transfer the source domain information to the target domain. 2. Active-learning technique is used to rely on only few target domain labels in a non-uniform distribution to build an effective model. The experiments and results show that our method outperforms various domain adaptation techniques with significant increase in classification accuracy.

Contents

Introduction **1**

1.1 Problem Statement **1**

1.2 Contribution **2**

1.3 Thesis Organization **3**

Background and Related Work **4**

2.1 Astronomy Background **4**

2.1.1 Supernova 4

2.1.2 Types of Supernova 5

2.1.3 Observation of Supernova 8

2.1.4 Challenges 10

2.2 Machine Learning Background **11**

2.2.1 Machine Learning Methods 11

2.2.2 Domain Adaptation 13

2.2.3 Active Learning 21

Methodology **24**

3.1 Overview **24**

3.2 Kernel Principal Component Analysis **25**

3.3 Landmark Selection **27**

3.4 Landmarks Clustering **30**

3.5 Domain Adaptation using Active learning **31**

Experiments and Results **36**

4.1 Experimental Setup and Background **36**

4.1.1 The Data 36

4.1.2 Weka 38
4.1.3 Matlab 38
4.1.3 Machine learning algorithms 39
4.2 Dimension reduction **39**
4.2.1 Principal Component Analysis 39
4.2.2 Kernel Principal Component Analysis 42
4.2.3 Discrete Wavelet Transform 43
4.2.4 Comparison of dimension reduction techniques 45
4.3 Domain Adaptation using Landmarks and Active Learning **46**
4.3.1 Landmark selection 46
4.3.2 Landmark clustering 47
4.3.3 Domain adaptation using Active learning 48
4.4 Kernel Mean Matching **51**
4.5 Unsupervised Subspace Alignment **53**
4.6 Landmarks-based Kernelized Subspace Alignment **54**
4.7 Comparison of results of domain adaptation methods **56**
Future Work and Conclusion **59**
5.1 Limitations and Future work **59**
5.2 Conclusion **60**
References **62**

Chapter 1

Introduction

1.1 Problem Statement

Although it has been a decade since the discovery of dark energy, astronomers still know very little about it. To improve their understanding of dark energy, they require cosmological parameters, which can be inferred from measurements of distances. It is easier to retrieve information from our Milky Way galaxy and nearby galaxies with the help of Cepheid stars. Cepheid stars are used as stellar candles for nearby galaxies to calculate distances with the help of the intensity of their light. But for farther galaxies, the brightness of Cepheid stars is not sufficient for the measurement. Astronomers require some brighter source of light to serve as standard candles to measure extragalactic distances. Supernovae are cataclysmic event, which happen at the last evolutionary stage of a star's life cycle releasing a tremendous amount of energy for a particular period. The light produced by some supernova can outshine their host galaxy for weeks or months. Thus, supernovae are ideal for serving as standard candles. There are different types of supernova events. But only Supernova Type Ia are homogenous enough to be used in the calculations of cosmological parameters. Large telescopic surveys are conducted to gather observations of supernovae. The observations include light curves captured from supernova event. These observations are known as photometric observations. They include all types of supernova like Supernova Type Ib, Supernova Type Ic, Supernova Type II, etc. Thus, correct identification of Supernova Type Ia is necessary among other forms for accurate calculations.

The current method for classification of supernova depends on the high-resolution spectroscopic method that is very cumbersome and expensive. In the future, the number of telescopic surveys is going to increase, which will lead to an unprecedented amount of photometric observations of supernova. Thus, astronomers require an automated method for the classification of supernova photometric observations to take advantage of future incoming data. Since, only a few of the supernovae samples are classified with the help of spectroscopy method, they also want to take advantage of existing supernova dataset classified with spectroscopy method to generate automated methods for the classification of supernova photometric data.

1.2 Contribution

In this thesis, the main goal is to find a methodology for the automatic classification of supernova photometric data. Employing machine-learning algorithms has solved many classification problems. With the use of machine learning techniques, it is possible to develop an automated method to solve the problem of supernova classification. We have proposed a method to generate a predictive model for the classification of supernova photometric data with the help of machine learning techniques. Since, both the supernova dataset classified with spectroscopy method and supernova photometric data belong to different domains and it is desired to use supernova spectroscopic data for the generation of automated method. We have developed a novel approach to domain adaptation technique to adapt the information across the domains and for the generation of an efficient model for classification. Many experiments have been performed on our method using different parameters. We have also performed various experiments on existing domain adaptation techniques for comparing the performance of our method.

The method proposed can be applied to other astronomy classification problem; it can be used outside the astronomy as well.

1.3 Thesis Organization

Chapter 1 contains the problem statement and our contribution to solve the problem. For the proper understanding of the problem and technical concepts related to methodology, some background knowledge will be provided in Chapter 2. Chapter 3 will include methodology explained in details. Followed by chapter 4, which will include experiments and results discussed in a precise manner. The last chapter of this thesis is chapter 5, which will conclude the thesis by providing limitations, future work and summary of the thesis as the conclusion.

Chapter 2

Background and Related Work

2.1 Astronomy Background

2.1.1 Supernova

Supernova is an astronomical phenomenon, which occurs at the last evolutionary stage of massive star's life cycle whose mass is greater than eight solar masses (1 solar mass = mass of the sun). This event includes the destruction of a star with a final gigantic explosion. The amount of energy released during this event is approximately equivalent to the amount of energy created during the entire lifetime of the solar-like star (star whose mass is equal to the sun) and can outshine the entire galaxy for few months or weeks. It is visible as a sudden appearance of a bright star whose magnitude degrades gradually and diminishes within a period. It may take weeks or months to disappear from the sight. The radiant of energy during supernova explosion can cause all its stellar material to expel at a speed of 30,000 km/hr. This speed is 10% of the speed of the light resulting in the fast-moving shock wave into the surrounding as an expanding shell of gas and dust, which mainly includes ejection of heavier elements results from nucleosynthesis. They distribute high mass elements in the galaxy and can trigger the formation of new star or planets.

The word supernova has been derived from the Latin word 'Nova' means new. Since supernova emerges as a new bright star at first which are brighter and energetic than a new star. Thus, Super word differentiates supernova from ordinary novae star. It is a rare event. So, far only three supernova explosions have been observed naked eye in

Milky Way galaxy. The most recent one has been noticed in the year 1604 named as SN1604. However, it is visible in other galaxies and can be recognizable by astronomical equipment.

2.1.2 Types of Supernova

To better understand the phenomenon astronomers have classified supernova in mainly two categories Type I and Type II. Astronomers' classify supernovae with Type codes that describe the type of light received from a supernova and the absorption lines of different elements in the spectra. The optical spectra provide the information of physical and chemical properties of the outermost layer of the star. The primary division of supernova is based on the presence of hydrogen lines in the spectrum. If the hydrogen lines are present in the range then supernova is classified as Type II or else Type I. There are further subdivisions in Type I supernova viz. Type Ia, Type Ib, and Type Ic depends on the further presence or absence of higher elements after hydrogen in the supernova spectrum.

2.1.2.1 Types Ia Supernova

Type Ia supernovae evidence the absence of hydrogen lines in the optical spectra. They were most popular in the last decade as they helped to detect the accelerated expansion of the Universe. Along with the absence of hydrogen lines, SNIa is characterized by the presence of elements like calcium, oxygen, silicon and sulfur in their peak luminosity phase. They are present in the outer layer of the exploded star. With the gradual appearance of iron in the spectra with the age contribution as it reaches to the core, which suggests the association of thermonuclear explosion of a white dwarf.

Type Ia supernova mainly occurs in a binary system where one of the stars is a carbon-oxygen white dwarf which is capable of nuclear fusion reaction resulting in the release of a high amount of energy, and another star companion can be any star between giant star or star whose mass is lighter than a white dwarf star. In the binary system, they orbit each other. The star companion swells massively as it ages during the process. Since, the white dwarf star is small and dense, due to which their gravity is intense enforces to pulls the material of the companion star resulting in the increase in the mass of the white dwarf star. When the mass of white dwarf star reaches a critical mass known as Chandrasekhar limit that is 1.4 solar masses it results in the explosion of Type Ia supernova causing the companion star eject away from the orbit.

Astronomers use Type Ia supernova to calculate extragalactic distance. Considering the intrinsic brightness of the Type Ia supernova homogenous throughout cosmic history they compare the brightness by how bright the supernova explosion should be and how bright it appears. It is based on the inverse square law, which states that, if two light source objects are in a line one after the other then the nearer object will appear brighter as compared to the farther object. Thus, it helps in estimating the distance based on the dimness of the light. Although to measure distance in our galaxy and a local group of the galaxy, they use Cepheid star as a standard candle but to measure distance in the farther galaxies, astronomers need an extremely brighter object. Thus, with the help of inverse square law, they compute the distance between us to the supernova ultimately giving the distance of supernova home galaxy.

2.1.2.2 Type Ib and Ic Supernova

Type Ib and Ic supernova both originate from the core collapse of the massive star. They mostly found in the spiral galaxies or HII regions of spiral galaxies. The energy source of

stars is nuclear fusion reaction, which fuses hydrogen elements to helium. A nuclear fusion reaction generates enough energy to oppose the gravitation force of the stars moving inward. Thus, maintaining the equilibrium prevents the star from collapsing. But if nuclear fusion reaction starts fusing into elements have an atomic mass heavier than helium like carbon, oxygen, iron and cobalt, etc., then it forms onion-like layer structure. Each layer represents an element starting with hydrogen followed by helium, carbon, neon, oxygen, silicon and iron. Formation of iron and nickel in the core of the star doesn't contribute towards energy production. Thus, no further nuclear reaction happens which results in the lack of energy. Due to insufficient energy the equilibrium breaks and the force of gravity starts pushing inward compressing the core with the overlaying mass of the star. When the mass of the core due to compaction reaches the critical mass, which result in a rapid collapse and explosion of the core within seconds. The outer layer of the star that includes hydrogen and helium is washed off due to the wind or either companion star, which are 3-4 solar masses. So, Type Ib lost its hydrogen layer due to collapse, and Type Ic lost both hydrogen and helium layer which differentiate Type Ib with Type Ic supernova.

Since it belongs to Type I supernova, there is an absence of hydrogen lines in the optical spectra. However, Type Ia can be differentiating with Type Ib & Ic by the presence of strong silicon absorption line at the maximum light. The absence of silicon lines and the presence of helium absorption lines represent Type Ib supernova, and the lack of both silicon and helium absorption lines characterizes Type Ic supernova. The difference between SNIa and SNIb becomes more evident in the later stages of the supernova when the outer layer starts to fade away as it ages resulting in the strong helium and oxygen absorption lines in the spectra for SNIb. SNIa shows strong iron and cobalt absorption lines in their later times.

2.1.2.3 Type II Supernova

Type II supernova is similar to Type Ib/Ic supernova originated from the core collapse of the massive star. But the only difference between Type II and Type Ib/Ic supernova is the presence of hydrogen absorption line in the optical spectra that is not present in the Type I supernova. Type II supernova can be further subcategorized into Type IIL, Type IIP, Type IIn and Hypernova based on their behavior in the optical spectrum. Type Ib/Ic and Type II supernova are mostly referred as the core-collapse supernova. These types of supernova are very prone to become a black hole depending on the mass of the core and the initial size of the star.

2.1.3 Observation of Supernova

The studies of supernova are carried out by the observations of light curves, i.e., graph between luminosity versus time following the explosion and the optical spectra, which are known as photometry and spectroscopy observation respectively.

2.1.3.1 Supernova Photometric Observation

Photometry is related to the study of the brightness of the object in a particular color. This type of study is mainly carried out with astronomical objects and events where the information source is the amount of energy, in the form of electromagnetic radiations known as flux. It helps to reveal the physical properties of the astronomical objects for example size of the object, object temperature, the distance between the object and other physical properties. The study of supernova also includes the photometric observation, which measures the variation of apparent brightness of the event over a particular period, which is mainly known as light curves. It is the measurement of flux over a broad wavelength of the radiations.

2.1.3.2 Supernova Spectroscopic Observation

Spectroscopy is related to the study of the spectrum of the electromagnetic radiations received from the objects. It is mostly used to examine the celestial objects and events by gathering the visible lights or infrared rays through a telescope. The electromagnetic radiations are spread into different wavelength with the help of spectrometer, which is a prism or grating material and projects into a screen, which is known as spectra. Spectra reveals about the chemical composition of the object based on the emission and absorption lines of the electromagnetic radiations in different wavelength that is also known as fingerprints of atoms and molecules. An object can emit three kinds of spectra: continuous spectra, emission spectra and absorption spectra. Continuous spectra are the smooth gradient of light like passing the sunlight through the prism will spread the white light into continuous colors of longer and shorter wavelength. If there will be some missing lines in the spectrum typically the dark lines which happen due to the absorption of wavelength by colder medium, this type of spectrum is known as an absorption spectrum. The stars produce this kind of spectrum when the colder outer region of stars absorbs the radiations of the inner hotter part. This type of spectra helps to know the elements present in the celestial object. Emission spectra are the spectrum of wavelengths of electromagnetic radiations of different atoms and molecules of the object. As some atoms and molecules emits extra light in the hot regions giving a sharp line in the spectrum. Thus, spectroscopic observations of supernova help to gather the information regarding chemical composition, mass, temperature, diameter, and distance, etc.

2.1.4 Challenges

Astronomers are more prone towards Type Ia supernova as it is used as 'standard candle to measure the distance to farther galaxies. In some cases, Type Ib/Ic supernova light curves appear identical to the light curves of Type Ia. Thus, they act as a contamination in the supernova surveys introducing erroneous in the calculation of the distances. For the correct estimation of the distance, Type Ib/Ic should be carefully removed from Type Ia samples. As the astronomical surveys are increasing day by day with advanced techniques, a significant amount of data is expected to gather. So, introduction of an automated system to classify the observations as different types of supernova is a necessity. With the help of machine learning techniques it is possible to classify the observation with the respective supernova type.

The observations of astronomical data mostly include high dimensions. The curse of dimensionality also possess a significant challenge, as a system will require a considerable amount of time and space to process the high dimension data efficiently and to produce the results. An efficient dimensionality reduction technique is needed to reduce the number of the observation without much loss of information.

Another challenge for supernova classification is that there can be two methods of observation for supernova as explained above. The present system for supernova classification is based on high-resolution spectroscopic observations. But in future with the upcoming surveys like Dark Energy Survey, Large Synoptic Survey Telescope (LSST) etc., it is possible to gather a large number of supernova observations based on photometric method. Since, we already have information about the classification of supernova based on spectroscopic method, an automated and adaptive method is

required to take advantage of information of existing system to classify the future incoming data.

2.2 Machine Learning Background

2.2.1 Machine Learning Methods

Machine learning is a field of computer science mainly a subarea of artificial intelligence that provides machines capability to learn without explicit programming. Machine learning has been applied to various real world problems successfully. Astronomy is also not remaining untouched with the application of machine learning techniques. Many astronomy problems have been solved with the development of automated systems based on machine learning methods, but it is still in a naïve stage. Machine learning tasks can be categorized into supervised learning, unsupervised learning, and reinforcement learning. Most of the problems can be dealt with supervised learning in which the goal is to build a model and determine the classification of an object based on their specific features. To learn a model, a training set is used which are the features of objects with their known classification. After the model is built with the help of training set, it is capable of classifying the object of unknown classification. To validate the model testing data is used. Testing data is a part of training data, which has been split before training. The dataset of known classification is divided into 2/3 training data and 1/3 testing data. But this might introduce bias in the model or the model can prone to overfitting. There are various methods to divide the dataset into training and testing which also includes ten-fold cross validation technique. In this procedure dataset is split into ten partitions randomly, and ten iterations are performed. In each iteration, nine partitions are treated as training set and one partition as testing data. This procedure results in the reduction of the bias of the model and tends to prevent overfitting.

In a variety of astronomical classification problem like star classification, galaxy morphology classification, supernova photometric classification, etc. included machine-learning techniques. With the discovery of dark energy, astronomers are conducting extensive surveys to garner more information regarding its nature and physical structure. During these surveys, they expect a significant number of supernova samples, which can be used as standard candles to measure the distance of the galaxies and other cosmological parameters as a function of redshift. It is possible to classify supernova with the help of spectroscopy method, but it is a very cumbersome task. Spectroscopy method is not an automated way to classify the supernova. Whatever the samples of supernova has been provided by the various surveys, only a fraction of it has been succeeded in labeling with spectroscopy method mainly by Supernova Legacy Survey [Asteir et al., 2006] and Sloan Digital Sky Survey [York et al., 2000].

Various efforts have been made in the past to provide an efficient statistical tool for the automated classification of supernova [Poznanski et al., 2002; Johnson & Crotts, 2006; Sullivan et al., 2006; Poznanski et al., 2007; Kuznetsova & Connolly, 2007; Kunz et al., 2007; Sako et al., 2008; Rodney & Tonry , 2009; Gong et al., 2010; Falck et al., 2010].

Most of them focused on the notion of template matching technique, in which they generated templates using supernova samples with known classification, and all the samples with unknown classification are classified by set of templates. The problem with this approach is that final classification is highly sensitive to the characteristic of template sample. Another approach has been made using posterior probabilities of each classification output [Newling et al., 2011; Sako et al., 2011]. In this method, the posterior probability of each data point has been calculated and assigned to templates according to their weight.

Following an entirely new approach [Richard et al., 2012] proposed a method of diffusion map to represent the data in low-dimensional space and in this random forest algorithm has been used to assign labels to supernova photometric data.

In the most recent work by [Karpenka et al., 2012], he proposed two-step method to classify the data. In the first step, spectroscopic confirmed supernova has been fitted to a parametric function and the parameters obtained are used to train a neural network classifier. In the next phase, the model obtained from the neural network is used in supernova photometric samples, which in turn provides the probability of a sample being an Ia supernova.

Another interesting approach has been made recently by [Ishida and de Souza, 2012], in which they used Kernel Principal Component Analysis to represent spectroscopically confirmed supernova data in low dimension and project photometric supernova samples to the same dimension. After projecting the supernova data in the lower dimension, k nearest neighbor algorithm is used for classification. Since the distributions of the both dataset are different, but their classes are same, there is a possibility for the application of domain adaptation techniques to improve the classification accuracy.

2.2.2 Domain Adaptation

One of the most common assumptions of machine learning problems is that the underlying distribution of the training dataset, which is used to train a model, is similar to the distribution of the testing set. However, in real world problems, this assumption is not valid. In many cases, training and testing dataset distributions are different, i.e., their marginal distributions are different which leads to the deviation of the classifier from the optimal model resulting in misclassification. In such cases, we use domain adaptation algorithms.

For the better explanation of domain adaptation we are assuming some terminology. Let X_s and X_t represent the training set also called source dataset and testing set also called target dataset and Y_s and Y_t denotes the class labels of the source data and target data respectively. Prior probabilities of the source and target data are represented as $P_s(X_s)$, $P_t(X_t)$,$P_s(Y)$,$P_t(Y)$ respectively and the posterior probabilities of the source and target are represented as $P_s(Y/X)$ and $P_t(Y/X)$. Since we are assuming a data point x belongs to X dataset, i.e., $\{x_s \rightarrow X_s\}$ and $\{x_t \rightarrow X_t\}$, the joint probabilities of the both source and target data can be represent as $P_s(x_s,y_s)$ and $P_t(x_t,y_t)$ where $\{y_s \rightarrow Y_s\}$ are the class labels belongs to the source class and $\{y_t \rightarrow Y_t\}$ are the class labels belongs to the target class. Thus, source domain can be defined as $D_s = \{X_s, P_s(X)\}$ and target domain as $D_t = \{X_t, P_t(X)\}$. The Learning task is to determine a predictive function that will predict class for the given features. For source domain, task is represented as $T_s = \{Y_s, f_s(.)\}$. Similarly for the target domain, task is defined as $T_t = \{Y_t, f_t(.)\}$. Thus, if we say domains are different it means their prior probabilities are different, i.e.,

$$P_s(X_s) \neq P_t(X_t) \tag{2.1}$$

But their feature space and tasks remain the same. If the model is built on only source dataset, it will not be able to classify target dataset correctly. The reason behind the difference of domain is due to data shift. There can be many reasons for the data shift in which one of the most prominent one is covariate shift. In covariate shift, we assume that there exists an optimal model that can classify both source and target dataset correctly. This leads us to assume that given certain model parameters the source and target posterior probabilities would be the same i.e.

$$P_s(Y_s/X_s) = P_t(Y_t/X_t) \tag{2.2}$$

But their marginal probabilities will be different. In most of the cases, source domain labels are available in abundance from which we can learn a source model that can

efficiently classify the source dataset but there is no or little information is available regarding target labels of the target domain. Thus, building a model on target domain with no prior information is difficult. This type of domain adaptation problem is known as unsupervised domain adaptation. Since, the source and the target domains are related we can use source information that is similar to target domain to learn a model in unlabeled target domain. This is called adapting a domain from source to target to get a better classification.

Thus, the goal of domain adaptation is to build optimal classification model on an unlabeled dataset (target dataset) by using relevant information from a related labeled dataset (source dataset).

Apart from covariate shift, there are many reasons for the data shift, which includes prior probability shift in which we assume that the likelihoods of the source and the target data have different priors of the class i.e.

$$P_s(Y_s) \neq P_t(Y_t) \text{ and } P_s(X_s|Y_s) \neq P_t(X_t|Y_t) \tag{2.3}$$

Another reason can be sample selection bias in which we assume that source domain is a sample of target domain but not completely represent the target domain. Imbalance data can also be a reason in which we assume that class priors are different but likelihoods of the source and the target are the same.

$$P_s(Y_s) \neq P_t(Y_t) \text{ and } P_s(X_s|Y_s) = P_t(X_t|Y_t) \tag{2.4}$$

But due to difference in the measurement of the dataset there can be a shift in the data this is known as domain shift.

2.2.2.1 Techniques of Domain Adaptation

2.2.2.1.1 Instance-based methods

Instance reweighting is one of the standard approaches for the domain adaptation problem. The assumption for the data shift here is the sample selection bias. We consider the source domain is a part of a target domain but not completely represent the target dataset. Thus, the source data that is closer to the target distribution can be given more importance by assigning weights to them, which will help to reduce discrepancy among the distributions. The goal of domain adaptation is learning an optimal model for the target domain that will minimize the expected loss over the target domain. Thus, it can be represented as

$$\theta_t^* = \arg min_{\theta \in \Theta} \sum_{(x,y) \in X \, x \, Y} P_t(x,y) l(x,y,\theta) \tag{2.5}$$

Which can be rewritten as,

$$\theta_t^* = \arg min_{\theta \in \Theta} \sum_{(x,y) \in X \, x \, Y} \frac{P_t(x,y)}{P_s(x,y)} P_s(x,y) l(x,y,\theta) \tag{2.6}$$

Since, P (X, Y) is unknown we can use empirical distribution to approximate P (X, Y) which can be represented as ~P (X, Y),

$$\theta_t^* = \arg min_{\theta \in \Theta} \sum_{(x,y) \in X \, x \, Y} \frac{P_t(x,y)}{P_s(x,y)} \sim P_s(x,y) l(x,y,\theta) \tag{2.7}$$

If we randomly select N_s number of training data points (x_i^s, y_i^s) from the distribution P_s (X, Y), we can minimize the empirical risk to get best model by following way,

$$\theta_t^* = \arg min_{\theta \in \Theta} \sum_{(x,y) \in X \, x \, Y} \sim P_s(x_i^s, y_i^s) \, l(x,y,\theta) \tag{2.8}$$

$$\theta_t^* = \arg min_{\theta \in \Theta} \sum_{(x,y) \in X \, x \, Y} l(x_i^s, y_i^s, \theta) \tag{2.9}$$

So, the above equation can be rewritten as

$$\theta_t^* = \arg min_{\theta \in \Theta} \sum_{i=1}^{N_s} \frac{P_t(x_i^s, y_i^s)}{P_s(x_i^s, y_i^s)} l(x_i^s, y_i^s, \theta) \tag{2.10}$$

Thus, we can get optimal target model by weighting the loss function over that data points. However, we don't have enough labels for the target domain to calculate the above ratio as weight. Various approach being made to calculate the ratio or replace the ratio with some other equations to calculate optimal model. Covariate shift by [Shimodaira,'00] assumed that prior probabilities of the source and the target domains are different but their posterior probabilities are the same. Thus, the weight ratio can be computed as follows:

$$\frac{P_t\ (x,y)}{P_s(x,y)} = \frac{P_t(x)\ P_t\ (y/x)}{P_s(x)\ P_s\ (y/x)} \tag{2.11}$$

$$\text{Since, } P_s\,(Y_s/X_s) = P_t\,(Y_t/X_t) \tag{2.12}$$

$$\frac{P_t\ (x,y)}{P_s(x,y)} = \frac{P_t(x)}{P_s(x)} \tag{2.13}$$

Thus, weights can be calculated by the ratio of priors of the source and the target domain.

In the class imbalance approach by [Japkowicz et al., 2002] they assumed that likelihoods of the source and the target domains are equal but their class priors are different.

$$\frac{P_t\ (x,y)}{P_s(x,y)} = \frac{P_t(y)\ P_t\ (x/y)}{P_s(y)\ P_s\ (x/y)} \tag{2.14}$$

$$\text{Since, } P_s\,(X_s/Y_s) = P_t\,(X_t/Y_t) \tag{2.15}$$

$$\frac{P_t\ (x,y)}{P_s(x,y)} = \frac{P_t(y)}{P_s(y)} \tag{2.16}$$

According to this approach if there is a class imbalance across the domains weights, it can be calculated as the ratio of the class priors of the target and the source data. In another kind of approaches, [Sugiyama et al., 2007] built density estimator for both source and target and calculated the weight by their ratio. One of the most interesting methods to solve domain adaptation problem was proposed by [Huang et al., 2006]. The method is known as Kernel Mean Matching in which they tried to minimize the maximum mean discrepancy between two distributions by projecting the distribution in reproducing Hilbert kernel space (RKHS). They estimated the beta values depends upon the source data points which are closer to the target points and multiplied the beta values to the loss function of the classifier to provide importance to the source data points which can help to build an efficient model for the target domain.

Another interesting approach has been done by [Bickel et al., 2007] in which they learned weights by initially assigning all the class of source data points as 1 and the target data points as 0 and determine whether the source data points lies in the target domain or not. One of the major assumptions in instance-based approaches for the data shift in the distribution is due to sample selection bias, and we assume that there learning tasks are similar, but it is not true in all cases. Thus, this led to another type of approach to address domain adaptation problem.

2.2.2.1.2 Feature-based methods

It is not necessary that the source domain is a part of the target domain, and the data shift reason is sample selection bias. Sometimes source and target distribution doesn't fall in the same region. In such cases, we can use some feature representation techniques to project source and target distribution in shared space such that they align. The goal of feature-based methods is to change the representation of features of both source and

target domain and explicitly project them in the new feature space for the better representation of shared characteristics between them.

One of the most common feature-based methods is subspace alignment given by [Fernando et al., 2013] in which they learned subspace for both source and target domain using principal component analysis and then aligned the source subspace according to the target subspace using a transformation matrix. When the source is aligned in the new space, it will represent the target domain. Thus, a model is built on aligned source dataset for the better classification of target dataset.

Another good approach in feature-based methods is Distance metric learning by [Kulis et al., 2011; Saenko et al., 2010] in which they calculated mahalanobis distance among the feature vectors and found the similarity and dissimilarity across the domains. Feature augmentation by [Daume III et al., 2007; 2010] is also a unique approach in which initially they train a model on only source domain and classify the target domain on that model. The predictions obtained by the model are augmented in the feature vector as additional features for the target data, and the target data is trained to build a model in the new feature vector. This approach requires target labels thus; this applies to fully supervised domain adaptation problem. Another method was proposed by [Blitzer et al., 2007] known as structural correspondence learning, which has been mostly used in sentiment analysis. They exploit both the source and the target domain to determine pivot features. Pivot features are the features, which behave similarly to both the source and the target domain. These pivot features are then integrated with the feature vector of the source domain to learn a model.

One of the traditional approaches in the feature-based methods is manifold learning by [Gopalan et al., 2011] in which they project both the source and the target domain in a new dimension, and they learn a geodesic path on the Grassman manifold between both

the projections. The most recent approach has been proposed by [Rafah et al., 2015] which is an improvised version of subspace Alignment technique [Fernando et al., 2013] known as Landmark based kernelized subspace alignment in which they choose landmarks across the source and the target domain. Landmarks are the regions where the source and the target data points overlap each other. These data points are known as candidates. After choosing landmarks, subspace alignment technique is applied to the candidates to get the better model in the target domain. Because this algorithm is the basis of this thesis, it is explained in more detail in Chapter 3.

2.2.2.1.3 Iterative-based method

The main principle behind this approach is to gather information regarding the target using pseudo labels and builds the model iteratively. If necessary add or remove the data points and continue till the convergence has been reached or no more data points remain. One of the most traditional approaches, which belong to this category, is DASVM (Domain adaptation support vector machine) by [Bruzzone et al., 2010]. In this method, initially model is built on support vector machine using the source data and all the target data is classified using the initial model, which gives the pseudo labels for the target data. In the next step, select the target data points that fall above and below the margin of the support vector machine both for the class +1 and -1. Include that target points in the training set for the next iteration. Meanwhile, remove the source data points from the training set which are far away from the margin. Build the model again using new training set in which target points have been integrated and again classify the remaining target data points. Repeat the procedure of selecting the target points, including in the training set, removing the source data point, building the model on the training set and classifying the target data points till the convergence has been reached

or no source data points remain in the training set. One of the disadvantages of this approach is that it is not applicable for multiclass problem.

2.2.3 Active Learning

Active learning is a field of computer science and subfield of machine learning and artificial intelligence, which involves interactive learning process continuously. In traditional machine learning algorithms, hypothesis is built on the available training data at that particular time. But in active learning, every time new hypothesis is built actively by previous experience.

In contrast to passive learning, active learner chooses the data points from the dataset to label. This act helps to reduce the number of data point labels required while learning and thus reduces the labeling cost. This type of learning is applied to the problems where labels are available in an accessible manner for example spam filtering problem. If the user interacts with the system to label an email as spam or non-spam to the query asked by the system, it will be very helpful for the system in learning process and will lead to better classification. In active learning, there used to exist an oracle or expert[1], which provides labels for the data points that are being asked. There can be three types of approach to ask labels to the oracle to learn a model:

Query synthesizing: In this scenario of active learning, learner can ask query for any unlabeled data during its learning process. The data points for which labels has been asked can be any arbitrary instances. Thus, query synthesizing is not a popular approach as sometimes, learner can query the data points which has no information and

[1] In this work we are using simulated SN data and as a consequence all labels for the target sample are available. Thus enabling the method as a proof of concept.

insignificant towards the learning process. But it works well for the uniform data distribution.

Stream-based selective sampling: In stream-based selective sampling, learner can sample unlabeled data from the distribution and can decide whether to query label for the sample data points or not. The decision can be based on the information content of the data. If the learner is very uncertain about the particular data point, it can query the label or else discard the data point. This setting works well when there is non-uniform distribution of data.

Pool-based sampling: In pool-based sampling, the learner can choose sample data points to query from the pool of unlabeled data. In this scenario also, the learner decides whether to query the label for the selected data point depends on the information content. The difference between stream-based selective sampling and pool-based sampling is that latter has pool of data which are ranked before selecting best data point to be queried whereas, in stream based sampling, data points are queried sequentially by the learner.

In most of the active learning techniques, uncertainty sampling is being used to determine the best query in the data distribution. Uncertainty sampling refers to the selecting of the data points among the distribution, which is very uncertain and contains high information for querying. There can be three strategies for uncertainty sampling, which are as follows:

Least confident sample: In this strategy, data points that are least confident are selected for the query. The least confident factor is calculated by the lowest posterior probability of the data point for the given model. Usually, highest posterior probability is considered for the prediction of the class for the data point. Thus, lowest posterior probability represents uncertainty in the prediction. The only disadvantage of this

strategy is that it doesn't consider other posterior probability as information. It only considers the best prediction.

Margin sampling: In this strategy, data points with least margin are selected for the query. Margin is calculated by the difference of first and second highest posterior probability of the data point for the given model. One of the major advantages of this strategy over least confident sampling is that it considers two best posterior probabilities to determine the uncertainty.

Entropy Sampling: The measure of randomness is known as entropy. However, in machine learning, entropy is used to measure the average information content of a variable. Thus, highest entropy value represents the most informative data point, which can be selected to query.

Chapter 3

Methodology

3.1 Overview

The method for the classification of supernova photometric observations has been generated by the integration of two algorithms. Since it is a domain adaptation technique thus, we will refer supernova spectroscopic data as source data and supernova photometric data as target data. The experiments have been performed on Matlab R2015b code. Matlab has been chosen as it has rich machine learning and data analysis toolbox, easy to implement and best for faster statistical calculations.

The first step of the methodology is accompanied with a feature reduction procedure with the help of kernel principal component analysis. The attributes are selected by the highest classification accuracy in the source data, and the target data has been projected in the same dimension using Eigenvectors of the source data.

The next phase is the selection of landmarks in source and target distribution on the new reduced feature dataset. Landmarks are the set of points in source and target data where their distributions are similar. The points in the landmarks are referred as candidates. The data points in landmarks belong to the source distribution are referred as source candidates and those data points belong to the target distribution are referred as target candidates. There is no use of the source and the target labels for the selection of landmarks. This idea is based on the method proposed by [Rahaf et al., 2015].

After selection of landmarks, Clustering technique is applied on the landmarks to obtain different clusters of the source and the target candidates. For clustering, EM algorithm with Kmeans has been used. With, the help of domain adaptation with active learning on

each cluster, an individual model is generated. For testing purpose, target data points are assigned to the different clusters based on the minimum distance from Kmeans centroids obtain from the clusters and classified with the model of the cluster in which the target data point belongs. The predicted labels are compared with the actual labels of the target dataset, and classification accuracy is calculated in percentage. The methodology is explained in details in the further section of this chapter.

3.2 Kernel Principal Component Analysis

Kernel Principal Component analysis is a generalization of Principal Component analysis (PCA) in a non-linear way proposed by [Schölkopf et al., 1997]. Unlike PCA, Kernel PCA finds principal components in feature space rather than in input domain. In Kernel PCA, at first, data is mapped non-linearly into high dimensional dot product feature space F.

$$\Phi : R^n \rightarrow F \tag{3.1}$$

$$x \rightarrow \Phi(x) \tag{3.2}$$

Where, Φ represents a non-linear function and F represents large dimensional space. The covariance matrix of this high dimensional space F is given by,

$$C_F = \frac{1}{N} \sum_{i=1}^{N} \Phi(x_i)\, \Phi(x_i)^{\mathrm{T}} \tag{3.3}$$

With the help of kernel trick, we can determine the dot product of $\Phi(x_i)\ \Phi\ (x_i)^{\mathrm{T}}$ by kernel matrix of N X N as

$$K_F(x_i, x_j) = (\Phi(x_i)\ \Phi\ (x_i)^{\mathrm{T}}) \tag{3.4}$$

Kernel matrix also known as a gram matrix allow to compute the dot product in the feature space F without actually mapping the data in high dimension with a non-linear operator.

To find principal components, we need to diagonalize the covariance matrix, which can be done by finding Eigen values and Eigen vectors by solving this equation,

$$\lambda v = C_F v \tag{3.5}$$

Where, λ represents Eigen values ≥ 0 and v represents Eigen vectors.

Now, there exist coefficients α_i $(i = 1 \dots m)$ such that

$$v = \sum_{i=1}^{m} \alpha_i \, \Phi \, (x_i \,) \tag{3.6}$$

For the gram matrix $K_F(x_i, x_j)$ of size n X n, we can rewrite above equation as,

$$n\lambda K_F \, \alpha = K_F^2 \alpha \tag{3.7}$$

Thus, we can solve this equation by finding Eigen values and Eigen vectors of the gram matrix and $\lambda_1 > \lambda_2 > \dots\dots > \lambda_n$ represents non –zero Eigen values and $\alpha^1, \alpha^2 \dots, \alpha^m$ represents corresponding Eigen vectors.

The resulting kernel principal components can be calculated as,

$$y_k(x) = \Phi(x) \, v_k = \textstyle\sum_{i=1}^{m} \alpha_{ki}^n \; k \, (x, x_i \,) \tag{3.8}$$

For any test data point x, we can compute projection by using Eigen vectors by this equation,

$$\left(v^n, \Phi(x)\right) = \sum_{i=1}^{m} \alpha_i^n \; k(x_i \,, x) \tag{3.9}$$

All these equations are based on the assumption that the data is centered and mean is zero in the feature space. However, after projecting the data in the new feature space by using kernel matrix doesn't ensure that the data is centered. Thus, for centralization of kernel matrix can be computed by following equation,

$$K'_{ij} = (\, K_F \, - 1_m K_F - K_F \, 1_m + \; 1_m \, K_F 1_m \,) \text{ Where, } (1_m \,)_{\text{ij}} = 1/\text{m for all i,j} \tag{3.10}$$

Thus to summarize this algorithm, Kernel PCA is performed by following step:

1. Find the gram matrix from the source data: $K_F(x_i, x_j)$.
2. Centralize the kernel matrix using equation (3.10).
3. Solve for the Eigen decomposition problem and compute Eigen values and Eigen vectors $\alpha^1, \alpha^2 \ldots, \alpha^m$ using K'.
4. Compute kernel principal components using equation (3.8).
5. Project the target data with Eigen vectors of kernel matrix using equation (3.9).

The experiment has been performed using various kernel functions, which included polynomial functions with different degrees, Gaussian with different sigma value etc. The type of kernel function and the features were selected by the performance of transformed source data in different classifiers.

3.3 Landmark Selection

The first step of our methodology is the selection of landmarks. The idea of landmarks has been taken from the method proposed by [Rahaf et al., 2015]. Landmark selection is a segment of the method proposed by them. Landmarks are the set of data points in source and target distribution; if we project these points in a common space their distributions will be similar. Landmarks represent the source and target data points that overlap each other. To represent the source data, we will use S and to represent target data we will use T and landmarks will be designated as A. Thus, the set of landmarks will be the subset of S∪T. Landmarks are selected by finding the similarity between each point c belongs to S∪T with all the points p in S∪T. Each point c is considered as a landmark candidate. If the similarity measure of the candidate is above a threshold, it is considered as a landmark.

The similarity between candidate c and each point p ∈ S∪T is measured by Gaussian kernel with standard deviation s:

$$K(c,p) = \exp\left(\frac{-\left|\left|c-p\right|\right|^2}{2s^2}\right) \tag{3.11}$$

Here, s is the kernel radius. It defines the size of the neighborhood with which candidate landmarks will measure the similarity. The parameter s helps us to capture the local properties in the feature space. It also helps us for the better alignment of source and target data. The value of parameter s is very crucial while similarity measurement as extreme values of s lead to perfect match between the distributions. A very small value of s close to zero will give all the values of K(c, .) as 0 and a very high value of s will give all the values of K(c, .) as 1. To avoid this situation, we perform multi-scale analysis by computing Euclidian distance among all the pairs of points in the distributions and try every percentile of these distances. We try out a range of s based on percentiles and compute the overlap between the source and target distribution.

Thus, to summarize this step we compute Gaussian kernel using candidate c with source distribution and target distribution separately i.e.

$$K_S = \exp\left(\frac{-||c-p||^2}{2s^2}\right) \text{ Where p} \in \text{S} \tag{3.12}$$

$$K_T = \exp\left(\frac{-||c-p||^2}{2s^2}\right) \text{ Where p} \in \text{T} \tag{3.13}$$

Now we compute degree of overlap between the two sets of K(c,.) using an overlap function. To be able to determine the overlap between two distributions, they are approximated as normal distributions. Computation of overlap is based on the similar idea of Bhattacharyya coefficient. If we want to compute the overlap between two distributions, we can find integration of their product. Higher the value of the product, higher will be the overlapping between the distributions. The normal distributions can be summarized by their means and standard deviations. Let μ_s, σ_s represents the mean

and standard deviation of normal distribution of source data and μ_T, σ_T represents the mean and standard deviation of normal distribution of target data.

Integration of two normal distributions is given by:

$$\int \mathcal{N}(x \mid \mu_s\ , \sigma_s^2)\ \mathcal{N}(\ x\big|\ \mu_T, \sigma_T^2)dx = \mathcal{N}(\mu_s - \mu_T\ , \sigma_{sum}^2) \tag{3.14}$$

Where, $\sigma_{sum}^2\ =\ \sigma_s^2 + \sigma_T^2$.

Thus, the overlap function can be given by:

$$overlap(\mu_s\ , \sigma_s\ ;\ \mu_T\ , \sigma_T) = \frac{\mathcal{N}(\mu_s - \mu_T\ , \sigma_{sum}^2)}{\mathcal{N}(0|0\ , \sigma_{sum}^2)} \tag{3.15}$$

The denominator of the overlap function represents the maximum value of the numerator as, if the means of both the distribution will be the same, $\mu_s - \mu_T\ =$ 0. Thus, it acts as a normalization factor. The maximum value of overlap function will be one if two distributions will perfectly match. The range of overlap function value helps us to determine the threshold th to consider candidate as a landmark or not.

Thus, algorithm *select_landmarks* can be summarized as follow:

Input: Source data S, Target data T, threshold th

Output: Landmarks A contains set of points from source and target data

A → {}
Distances ← {||a-b||, (a, b) ∈ $(S\ \cup T)^2$}
for c in $(S\ \cup T)$ do
for s in percentiles (distances) do
$K_s = \exp\left(\frac{-||c-p||^2}{2s^2}\right)$ Where p ∈ S
$K_T = \exp\left(\frac{-||c-p||^2}{2s^2}\right)$ Where p ∈ T
If overlap (KV_s, KV_T) > th then
A = A ∪ {c}
End if
End for
End for

3.4 Landmarks Clustering

The next step after selecting the landmarks from the source and the target data is clustering. Many clustering techniques can be applied to obtain different clusters. Since landmarks are set of source and target data points that overlap each other, with the help of clustering we can determine which source and target data points are overlapping. Among the various clustering technique, we have applied EM algorithm on the landmarks for the clustering. EM algorithm assigns the probability to each data point, which determines the probability of it belonging to each of the clusters. Since the number of clusters to be formed is not known to the prior, we have chosen Weka to perform EM algorithm on the landmarks. It doesn't require number of clusters to be formed as a parameter before executing this function. EM algorithm uses cross-validation procedure to determine number of clusters.

In Weka, EM algorithm performs the following procedures to determine the number of cluster and cluster means:

1. Initially the number of clusters is set to 1.
2. The data provided for clustering is split into ten folds randomly.
3. EM algorithm is performed ten times in each ten folds similar to cross-validation procedure.
4. The log likelihood determined by EM algorithm is averaged across ten folds.
5. If the log likelihood has increased then the number of cluster is increased by 1.
6. Execution continues to step 2.

Thus, EM algorithm provides the number of clusters and cluster means as output. Since, Kmeans clustering algorithm is used in the framework of EM algorithm, we assigned landmarks to the clusters based on the minimum Euclidian distance from the cluster means.

3.5 Domain Adaptation using Active learning

The goal of domain adaptation is to transfer the source information to the target domain to build the target model for better classification. Active learning helps to exploit the source and the target domain locally. After clustering of landmarks, the next step is to apply domain adaptation using active learning. The first step of this method is to build the initial training set, which will be used for building initial model for active learning. In basic active learning algorithms, the initial model is learnt using random sampling. In random sampling, few data points are randomly selected and queried for the labels. We are trying to use source information for building the target model. Thus, in the first step, we are incorporating source information in building initial model. Here, we have assumed that source candidates represent target candidates in an overlapping region.

In each cluster, the number of source candidates, NS_c is calculated. Let *minPoints* is the minimum number of data points require to learn a model, S_c represent the source candidates present in one cluster and NT_c' represents the number of target candidates randomly selected from the cluster. If the number of source candidates, NS_c is less than *minPoints* then rest of the points, NT_c' = (*minPoints* - NS_c) is selected randomly from the target candidates in the cluster. For example: In cluster 1, if the number of source candidates is 4 and the minimum data points required to build the initial model is 10, then 10-4 =6 target candidates are randomly selected from the cluster 1 and included in the training data. Let the randomly selected target candidates is defined by T_c'. Target candidates, T_c' is queried for the labels from the expert or oracle and included in the initial training set along with the source candidates. Thus, in this case, initial training set = $S_c \cup T_c'$. If the number of source candidates, NS_c is more than *minPoints;* the entire source candidate is included in the training set. As we want to take advantage of source

information as many as possible. So, in this case initial training set = S_c. But if cluster will not contain any of the source candidates, i.e., NS_c = 0, then target candidates are randomly selected from the cluster equal to the minimum number of data points required to learn initial model i.e. NT_c' = *minPoints*. In this case, initial training set = T_c'. Thus the algorithm *initial_training_model* is as follow:

Input: Clusters, *minPoints*

Output: *train_set, train_labels* for each cluster

For each Cluster C = i... N where N = number of clusters

Compute number of source candidate in the cluster, NS_c.

If NS_c = 0 then

Choose T_c' from the cluster where NT_c' = *minPoints*.

L_T' = Query the labels of T_c' with expert.

train_set= T_c', *train_labels* = L_T'.

Else if NS_c < *minPoints* then

Choose T_c' from the cluster where NT_c' = *minPoints* - NS_c.

L_T' = Query the labels of T_c' with expert.

train_set = $S_C \cup T_c'$, *train_labels* = $L_s \cup L_T'$.

Else

Choose S_C from the cluster.

train_set =S_C, *train_labels* =L_s.

End If

End For

Where,

S_c = Source candidates in Cluster, NS_c = Number of source candidates in the cluster,

NT_c' = Number of target candidates selected randomly from the cluster,

minPoints = Minimum data points required to learn a model,

T_c' = Randomly chosen target candidates from the cluster, L_s = Label set for the source candidate, L_T' = Label set for randomly chosen target candidate, *train_set* = initial training set, *train_labels* = initial training labels.

The above algorithm represents domain adaptation technique as we are transferring source information for the next step of methodology. The next phase is to apply active learning. With active learning, we rely on only few target labels for better classification of supernova photometric data. Active learning is applied in each cluster individually. The first step in active learning is to learn initial model in each cluster based on the training set and training labels obtained from algorithm *initial_training_model*. Each cluster is assigned with maximum number of target candidate labels that can be queried during active learning. We refer this parameter as maxCost. At first in each cluster, all the target candidates, T_c are classified with the initial model. With the posterior probabilities obtained from classification is used for the calculation of margin. The margin is calculated as the difference of highest posterior probability Pr_1 with second highest posterior probability Pr_2.

$$Margin = Pr_1 - Pr_2 \tag{3.16}$$

Since, the candidate with lowest margin represents the least confident data point.

Thus, the target candidate with the minimum margin, T_M is selected and asked for the label to the expert. Include this target candidate, T_M in the initial training set and target candidate label, L_M in the initial training labels obtained from the algorithm *initial_training_model*. Again learn model on the basis of new training set and classify all the target candidates in the cluster. Compute the margin for all the target candidates. Pick the target candidate with lowest margin and query for the label from expert. Include

this target candidate and label in training set and training label. Repeat this procedure till the maxCost is reached.

Thus, algorithm *active_learning* is as follows:

Input: Cluster C, *train_set, train_labels*, maxCost

Output: *final_model* for each cluster

For each C = i… N where N = number of cluster

While cost ≤ maxCost

model = classifier (*train_set, train_labels*)

[Pr] = classify (*model*, T_c).

Calculate the margin for each target candidate, T_c as

$$Margin = Pr_1 - Pr_2$$

Pick the target candidate, T_M with minimum margin.

Query the label, L_M from expert for target candidate T_M.

train_set = train_set ∪ T_M, train_labels = train_labels ∪ L_M.

cost = cost +1

End while

final_model = *model*

End For

Where,

C= Cluster, Pr = posterior probability, Pr_1 = highest posterior probability,

Pr_2 = Second highest posterior probability, T_M = target candidate with minimum margin

L_M = Target candidate label with minimum margin, maxCost = maximum number of labels can be queried from expert for each cluster.

Final model obtained from above algorithm for each cluster is the required predictive model, which can be used to predict labels for test data.

Thus, to summarize the methodology:

Step 1: Selecting landmarks in source and target data.

Input: Source data S, Target data T, threshold th

Output: Landmarks A

A = *select_landmarks* (S, T, th)

Step 2: Apply EM algorithm for clustering landmarks.

Input: Landmarks A

Output: Clusters C

Step 3: Apply domain adaptation and active learning on each cluster C.

This step will be carried off in two sub steps:

a. For each cluster, determine initial training set and training labels. This step can be referred as domain adaptation technique.

   Input: Input: Clusters C, *minPoints*

   Output: *train_set, train_labels* for each cluster

   [*train_set, train_labels*] = *initial_training_model*(C, minPoints)

b. Apply active learning technique in each cluster

   Input: Cluster C, *train_set, train_labels*, maxCost

   Output: *final_model* for each cluster

   final_model = active_learning(C, *train_set, train_labels*, maxCost)

Testing of the target data is done by final model obtained from each cluster. Each target data point is assigned to the different clusters based on the minimum Euclidean distance from the cluster means. After assigning the target data points to the cluster, they are classified with the final model of their respective cluster to get predicted labels.

Chapter 4

Experiments and Results

4.1 Experimental Setup and Background

4.1.1 The Data

The Supernova Photometric Classification Challenge had released a dataset of supernova, which consisted of simulated supernovae with different types selected in ratio to their expected rate. It consisted of nearly 20,000 supernova light curves. Light curves were simulated according to Dark Energy Survey specifications. It has been simulated using SNANA light curve simulator. The dataset has been divided into training set and test set. The training set consisted of confirmed light curves using spectroscopic method, which are lesser in number as compared to test set and test set consisted of photometric samples.

Preprocessing of light curves was required before implementing any machine learning techniques for photometric classification. Thus, the observation from each supernova passed through different filters. Let the number of filters were b. The filters can be denoted as F = $\{F_1, F_{2,} \dots F_b\}$. Each filter consisted of series of three parameters F_i = $\{\{t_{i1}, f_{i1}, \sigma_{Fi1}\} \dots \{t_{ie}, f_{ie}, \sigma_{Fie}\}\}$ where t_{ij} = jth observation epoch, f_{i1} = flux measured at time t_{ij}, σ_{Fij} = error introduced in the measurement and e = number of observations epochs in F_i. All the observations were taken in MJD format. MJD is abbreviated for Modified Julian Day is a dating method used by astronomers. Thus, for each observation, time in MJD was translated to the time since the maximum brightness in each filter. Let t_{max} is the ideal time of peak brightness for an observation of supernova.

For each filter F_i, time of maximum brightness in each epoch was determined by

$$(t_{max})_{ij} = t_{ij} - t_{max} \quad (4.1)$$

The data points in each filter i represented as $F_i = \{\{(t_{max})_{i1}, f_{i1}, \sigma_{Fi1}\} \dots \{(t_{max})_{ie}, f_{ie}, \sigma_{Fie}\}\}$. It was possible to have non-uniform sampling of the light curve in different filters for each supernova observation. The information translated to grid equally spaced in time to obtain a smooth curved function for each supernova observation. This translation was done by Gaussian process. Normalization was done by the maximum flux measured in all filters for a particular supernova observation to ensure the light curve in a reasonable range. Let S_{Ni} denotes the normalized fitted curve for the observations of supernova in filter i. Each observation of supernova represented as $S_N = \{S_{N1}, .., S_{Nb}\}$ where b = number of filters. Each row of supernova observation represents all the information regarding single supernova and the column contains the flux measurement in a particular observation epoch and filter.

We applied our methodology to the dataset, which consists of two parts. The first part represents the training data, which are spectroscopically confirmed light curves. The number of observations for the training data is 718[2]. The second part represents test data, which are photometric samples. The number of photometric samples in test data is 11946[2]. The number of features in the dataset is 108[3] and the last column of each observation represents the class of the supernova. The supernova Type Ia, Ib and Ic is denoted as 120,111 and 113 respectively in the class column of the supernova observations. Thus, the data matrix of supernova training data composed of 718 rows and 109 columns. Similarly, the data matrix of supernova test data consisting of 11946

[2] After the selection cut

[3] Corresponding to observations between -3 and +23 days since maximum brightness

rows and 109 columns where 1 to 108 columns represent the feature set and the last column represents the class of the supernova to which it belongs.

4.1.2 Weka

For analyzing data and experimenting with different machine learning methods, we used Weka, an open source Java-based data mining application. It has a graphical user interface and many machine-learning algorithms available with the ability to change parameters as per our experiments. We can visualize data with the help of visualizing tool available in Weka. Also, It can be used for application of clustering algorithms and selection of attributes. Data should be in ARFF format to perform experiments in Weka. ARFF abbreviated for Attribute-Relation File Format. In .arff files, all the features are declared by @relation feature name and variables are separated by comma. We used Matlab code to generate ARFF format data files.

4.1.3 Matlab

Matlab is a platform used for solving scientific problems. It has rich toolboxes available for different scenarios like machine learning, image processing, computer vision, signal processing, robotics, etc. Matlab is also a fourth-generation programming language. It is intended for mathematical calculations. We have used Matlab for the code generation of various experiments. Matlab is also available with different machine-learning algorithms like Multilayer Neural Networks, Support Vector Machines, Naïve Bias, Decision Trees, etc. Apart from Weka, we have used Matlab for the experiments with machine learning algorithms.

4.1.3 Machine learning algorithms

Various machine learning algorithms were used for the experiments using Weka and Matlab. For example Multilayer Neural Network, Random Forest, Support Vector Machine with different kernels, J48 Decision tree. We have used default settings of Weka. For the clustering purpose, we have used EM algorithm from Weka with the default setting. In Matlab, we have used Multilayer Neural Network to perform experiments.

4.2 Dimension reduction

The first step of our methodology was to reduce the dimension of supernova data. The goal of every dimension reduction technique is to project high dimensional data onto lower dimension without affecting the high dimensional structure. It should be able to restore the information as much as possible. We have applied three dimension reduction techniques: Principal Component Analysis, Kernel Principal Component Analysis and Discrete Wavelet Transform. Various experiments have been performed on these algorithms to determine the right parameters. The selection of attributes has been made by their performance on classification of training data. Experiments and results of these three algorithms and comparison among them are provided in details in further sections of this Chapter.

4.2.1 Principal Component Analysis

Principal Component Analysis (PCA) is the most popular dimension reduction technique proposed by [Wold et al., 1987]. It is used to determine a small number of uncorrelated variables from a large set of data. These are called principal components. They are the direction where there is more variance in the data. The variables with more variance

contain more information and vice versa. It always tries to recognize strong patterns in the data to reduce dimension with minimum or no loss of information.

Let D is the dataset with d dimension and N number of samples. Thus, Data matrix D can be represented as

$$D = \begin{matrix} a_1 & \dots & a_d \\ \vdots & \vdots & \vdots \\ a_{1N} & \dots & a_{dN} \end{matrix} \tag{4.2}$$

The first step to determine principal component analysis is to center the data by their means that is done by determining the mean of each dimension and subtracting each variable by their mean. Let a_{mean} is the vector of mean values for D. Thus, centered observation X $\in$ R^d can be given as,

$$x_i = a_i - a_{mean} \tag{4.3}$$

Next step is to compute the covariance matrix C,

$$C = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T \tag{4.4}$$

To find principal components, we need to diagonalize covariance matrix, which can be done by solving eigenvalue equation,

$$\lambda_i v_i = C v_i \tag{4.5}$$

Where, λ_i = eigenvalues > 0 and v_i = eigenvectors.

After determining eigenvectors and eigenvalues, sorting of eigenvalues is done in decreasing order. The highest eigenvalue represent the dimension with highest information. Thus, the dimension reduction can be achieved by choosing k eigenvectors with largest eigenvalues. Let W represent the eigenvector matrix of N X K dimension. Thus, the transformed dataset, y is obtained by following equation,

$$y = W^T D \tag{4.6}$$

Application of principal component analysis on supernova data was done with the help of Weka. Weka is integrated with select attribute function from where we can select principal component method to apply on the provided data. The first step for this experiment was to convert the supernova training data into ARFF file format. We did this by using Matlab code. The training data in .arff format was loaded in Weka and select attribute tab was selected. In select attribute interface, principal component option was chosen and setting of parameters was done by changing the center data to true and providing the value of variance. Variance determines the amount of information we required in the reduced feature set. The value of variance varies from 0-1. We had performed experiments by changing the value of variance from 0.95 to 0.999 with an interval of 0.1. Results obtained by principal component analysis in Weka comprised of a covariance matrix, eigenvalues, eigenvectors and the number of attributes (reduced one). It also has a provision to save transformed data obtained after applying PCA. The transformed data used to be in .arff format compatible with Weka. Thus, in each experiment, value of variance was changed and PCA was executed. After execution, transformed data was saved and then again loaded in the Weka for ten-fold cross-validation classification. Different classifiers were used like Multilayer Neural Network, Random Forest, Support Vector Machine Kernel 1, Support Vector Machine Kernel 2, Support Vector Machine Kernel 3 and J48 Decision Tree for the classification purpose. The number of features was selected on the basis of highest accuracy obtained by classification of transformed data. Thus, for the variance value 0.999 we had obtained 16 features with highest classification accuracy of 93.03 % with Support Vector Machine Kernel 3 as the classifier.

4.2.2 Kernel Principal Component Analysis

Kernel Principal Component Analysis (Kernel PCA) is nonlinear principal component analysis used as dimension reduction technique. The methodology to perform Kernel PCA has already been explained in Chapter 3. We had implemented Kernel PCA in Matlab. Various experiments were performed using different kernels for Kernel PCA. We had run experiments for following Kernels: Polynomial degree 1, Polynomial degree 2, Polynomial degree 3, Gaussian kernel with sigma 0.5,0.6,0.7, and 1. The number of transformed data had been produced using Kernel PCA for each Kernel. The number of features in transformed data varied from 6-40. For each Kernel, we had generated 36 transformed datasets. Each dataset had number of features starting from 6 to 40 with an interval of 1. Each dataset then converted into .arff format to make it compatible to Weka using Matlab. The datasets were loaded in Weka and classified with ten-fold cross-validation procedure using different classifiers: Multilayer Neural Network, Support Vector Machine Kernel 1, Support Vector Machine Kernel 2, Support Vector Machine Kernel 3, Random Forest and J48 Decision Tree. The accuracy graph had been plotted for each kernel with x-axis represented the number of features ranges from 10-50 and y-axis represented the classification accuracy corresponding to that feature. The number of features was selected on the basis of highest classification accuracy on the transformed data. After analyzing accuracy graph of different kernels, the maximum classification accuracy with minimum number of features was obtained with polynomial degree 1 kernel, 23 features with classification accuracy of 93.44 % using classifier Support Vector Machine Kernel 3.

4.2.3 Discrete Wavelet Transform

Discrete wavelet transform is another popular dimension reduction technique. The technique of dimension reduction technique from discrete wavelet transform is taken from the method proposed by [Yinsheng Qu et al., 2003]. Data can be treated as signal, and the transformation of signal can be treated as another representation of the signal. It doesn't affect the information content of the signal. The wavelet transform is the representation of the signal in time and frequency. The wavelet transform can get required sampling by translation and dilation of mother wavelets. Discrete wavelet transforms is a type of wavelet transform in which wavelets are discretely sampled. If the data is transformed into wavelets, dimension reduction can be achieved by removal of very low amplitudes of the signal. As high amplitude signals carry most significant information rather than a very low amplitude signals, which carries less important information. The first step for discrete wavelet transform is to transform the dataset into wavelet coefficients. Wavelet coefficients are calculated by using cascade algorithm by [Mallat et al., 1989]. This algorithm includes recursive decomposition of original signal. Let S_0 is the original signal, at first step S_0 is decomposed into two signals: smoothed signal, S_1 also known as average signal and fluctuation signal, d_1 also named as detailed signal. The length of S_0 is equal to the length of $S_1 + d_1$. Again, average signal S_1 is further decomposed into S_2 and d_2 such that $S_1 \rightarrow S_2 + d_2$. Now, the original signal can be represented as $S_0 = S_2 + d_2 + d_1$. Thus, by decomposing the average signal recursively, we can represent the original signal. The wavelet coefficients obtained by the transformation of original data are mathematically equivalent to the orthogonal transformation of data, which we carried out in principal component analysis. The next step after getting wavelet coefficients is selecting a threshold, λ. Those coefficients whose absolute values are less than λ are set to zero. The threshold, λ is obtained by calculating

99^{th} percentile of the absolute values of wavelet coefficients. Thus, for each column in the dataset, a threshold λ is computed where λ_j represents the threshold value for column j (j =1… n) where n = 718. A voting method is implemented to select a common subset of wavelet coefficients. The voting method is based on the idea that a wavelet coefficient will be selected if its absolute value is greater than threshold λ in m out of n samples. After selecting the subset of wavelet coefficients, it is reconstructed to the original signal by using the same mother wavelet used during construction of wavelet. The reconstructed signal represents the reduced dimension data.

We have implemented discrete wavelet transform technique in Matlab and performed various experiments with different mother wavelets and different levels of decomposition. We had generated the wavelet coefficients using Haar, DB1, DB2, DB3, DB4, DB5, DB6, DB7, coif1, coilf2, coif3, coif4, and coif5 mother wavelets. The level of decomposition varied from level 1-11. For the selection of subset of wavelet coefficient we had chosen m = 10. The reconstructed dataset obtained by the application of discrete wavelet transform was converted into .arff format to make it compatible with Weka. The data was loaded in Weka and then classified with ten- fold cross-validation technique using different classifiers: Multilayer Neural Network, J48 Decision Tree, Random Forest, Support Vector Machine Kernel 1, Support Vector Machine Kernel 2, Support Vector Machine Kernel 3. All the datasets obtained by using different mother wavelets and different levels of decomposition were classified in Weka to get classification accuracy.

The features with highest classification accuracy were selected as reduced feature set for supernova data. Thus, the number of features of supernova data was reduced to 9 features with the application of discrete wavelet transform using DB6 mother wavelet at

the fourth level of decomposition. The classification accuracy for 9 features was 91.77 % with Multilayer Neural Network as the classifier.

4.2.4 Comparison of dimension reduction techniques

Different dimension reduction techniques have been applied to the supernova spectroscopic data (aka training data). The results obtained by dimension reduction methods are as follows:

Dimension reduction method	***Number of features***	***Classification Accuracy (%)***
Principal Component Analysis	*16*	*93.03*
Kernel Principal Component Analysis	*23*	*93.44*
Discrete Wavelet Transform	*9*	*91.77*

Table 4.1: Number of features and classification accuracy on reduced supernova training data using different dimension reduction techniques.

The above table represents the results of dimension reduction methods applied on supernova training data. The first column lists the dimension reduction methods applied. The second column determines the number of features obtained after application of dimension reduction method and the third column refers to the classification accuracy obtained on the training data after reduced features. As it is evident from the Table 4.1, the classification accuracy of training data obtained by Kernel PCA is more as compared to other methods. Although the training accuracies of Principal component analysis and Kernel PCA are very close but still we want the reduced feature to represent the original data as much as possible. Thus, we have chosen

Kernel PCA as the dimension reduction technique for the supernova training data and reduced the feature set from 108 features to 23 features. Supernova photometric data (aka test data) was also reduced to 23 features by projecting it to the same dimension using eigenvectors of reduced supernova training data.

4.3 Domain Adaptation using Landmarks and Active Learning

Our devised method has been already explained in Chapter 3. After a feature reduction procedure, next step is the application of domain adaptation. The experiments were performed on the 23-feature set both training and testing data obtained by the application of Kernel PCA. Since, we are discussing about domain adaptation, we will refer supernova reduced-feature training data as source data and supernova reduced-feature testing data as target data. The application of our method on the source and target data was followed in three steps: first step was the selection of landmarks from source and target data, second step was the landmarks clustering and the last step was the application of domain adaptation and active learning in each clusters to generate a predictive model. Experiments performed in each step are explained in details in further sections.

4.3.1 Landmark selection

The selection of landmarks from the source and the target data has already been explained in Chapter 3. For more details please refer Chapter 3 section 3.3. The method to select landmarks from source and target data was implemented using Matlab. For the calculations of Euclidian distances between the data points in source and target data, we had used Matlab inbuilt function. Experiments were performed by monitoring two

parameters for the selection of landmarks: the percentile of distance s and the overlap threshold th. The parameter s determines the size of neighborhood, with which the candidate points will measure the similarity. Any candidate point will be considered as a landmark if its overlapping function value will be greater than threshold th. It has already been mentioned that the value of s should not be extremely small and not be extremely large. Thus, we started experiment initially with s = 10 and th =0.9. The threshold th is fixed to 0.9 as the overlapping function value for each candidate were in the range of 0.86 to 0.94. So, we have chosen the average value 0.9 as threshold. We experimented with s= 5, 1, 0.5 and threshold th =0.9 for the selection of landmarks. The landmarks obtained after each experiment was saved for further procedure.

4.3.2 Landmark clustering

After the selection of landmarks, the next step was the clustering of landmarks. For most of the clustering algorithms, we should initially know the number of clusters to be formed. But, in our case it was unknown. Thus, we had chosen EM algorithm in Weka for clustering purpose. The landmark dataset was converted into .arff file format to be compatible with Weka. EM algorithm was executed on landmark dataset with Weka default parameters. The result obtained by clustering contains cluster means. The cluster means were saved in .csv format. Assigning each data point to the closest cluster mean formed clusters of landmarks. The number of clusters formed for the different landmark datasets are as follows:

Landmark dataset for s	***Number of clusters***
0.5	*33*
1	*37*
5	*35*
10	*34*

Table 4.2: Number of clusters formed for each landmark dataset

The number of target candidates was more in each cluster as compared to the source candidate. Few clusters were devoid of source candidates. The next step after clustering of landmarks was the application of domain adaptation and active learning.

4.3.3 Domain adaptation using Active learning

This part of experiment was conducted in two steps: the first step was the generation of initial model for each cluster and second step was the application of active learning on the basis of initial model. The methodology has already been explained in the Chapter 3 section 3.5. Various experiments were performed by regulating two parameters: *minPoints* in the first part and *maxCost* in the second part. *minPoints* refers to the minimum number of data points required to build initial model for each cluster. *maxCost* refers to the maximum number of labels can be queried with an expert for active learning for each cluster. For learning the initial model, we performed experiments by providing *minPoints* in the range of 5 - 20, with the interval of 5. After generating the initial model, it was passed to the function of active learning along with the *maxCost* for each cluster. For the purpose of classification, we have used Multilayer Neural Network with 12 hidden layers as the classifier. Since, we wanted to rely on only few target labels for the classification purpose. Thus, we limited the maxCost value from 5-20, with the interval of 5 for each cluster. Target data was tested on the model

generated by the proposed method. Classification accuracy was calculated to evaluate the model using Multilayer Neural Network with 12 hidden nodes. Since learning of initial model was based on the randomly selected target candidates. Thus, each experiment was performed ten times and the average of the classification accuracy was taken into consideration.

Following are the results obtained by the experiments:

Landmark Cluster for s	***Min Points***	***Classification Accuracy (%)***			
		maxCost= 5	***maxCost= 10***	***maxCost= 15***	***maxCost= 20***
s=0.5	*5*	*71.79 (1.55)*	*72.83 (1.36)*	*75.34 (0.65)*	*75.82 (0.39)*
	10	*72.72 (0.23)*	*74.27 (0.72)*	*76.09 (0.76)*	*77.68 (0.27)*
	15	*74.03 (0.56)*	*76.10 (0.60)*	*77.49 (0.43)*	*77.85 (0.73)*
	20	*77.45 (0.46)*	*77.63 (0.58)*	*78.18 (1.52)*	*78.53 (0.50)*
s=1	*5*	*72 (0.85)*	*75.60 (0.49)*	*76.88 (0.42)*	*77.40 (0.79)*
	10	*74.45 (1.13)*	*75.25 (0.72)*	*74.63 (0.82)*	*76.76 (0.26)*
	15	*75.24 (1.72)*	*77.88 (0.88)*	*77.44 (0.30)*	*79.12 (0.34)*
	20	*78.54 (0.99)*	*76.55 (1.24)*	*77.19 (0.43)*	***79.40(0.44)***
s=5	*5*	*72.42 (0.98)*	*72.08 (2.07)*	*73.91 (1.57)*	*75.88 (0.45)*
	10	*74.49 (1.70)*	*75.65 (0.83)*	*76.47 (0.85)*	*77.43 (0.27)*
	15	*72.98 (1.09)*	*74.42 (0.92)*	*76.41 (0.76)*	*77.66 (0.26)*
	20	*77.31 (0.60)*	*76.65 (1.12)*	*77.95 (0.49)*	*78.07 (1.04)*
s=10	*5*	*70.20 (0.72)*	*73.44 (0.65)*	*73.66 (0.92)*	*74.34 (1.02)*
	10	*71.65 (2.16)*	*74.08 (1.57)*	*73.02 (1.57)*	*73.74 (0.66)*
	15	*72.33 (1.46)*	*73.53 (1.09)*	*71.39 (2.40)*	*75.44 (1.07)*
	20	*75.07 (0.69)*	*74.57 (0.59)*	*75.51 (0.60)*	*76.56 (1.17)*

Table 4.3: Classification Accuracy on Supernova test data by the application of Domain Adaptation using Landmarks and Active Learning algorithm (our method)

The first column of the Table 4.3 refers landmark clusters based on the different value of s. The second column represents the minimum number of candidates required to learn initial model for active learning for each cluster. The last four columns represent classification accuracies on supernova photometric data based on different *maxCost* provided to each cluster. Along with accuracies standard deviations is also provided.

4.3.4 Analysis of the result

From the table 4.3, it can be seen that with the increase in *maxCost*, classification accuracy was increasing. It is because we were incorporating more target candidates in learning the model with the increase in maxCost. If we were making maxCost as constant, classification accuracy was increasing with the increase in *minPoints*. The reason behind the increment is that the probability of target candidates is more in initial training set with the increase in *minPoints*. Another point, it is worth noting that, the maximum accuracy was achieved for landmark dataset with s=1 and second best accuracies with s=0.5 and it didn't work well for s=10. It was already mentioned in the method proposed by [Rahaf et al., CVPR'15] that landmark selection is based on the value of the parameter s. Here, s refers to the neighborhood we are considering for finding the similarity between two distributions. According to the method of landmark selection, extreme values of s should be avoided. Thus, in our method maximum accuracy is achieved in the average value of s, neither too high nor too low.

To evaluate the performance of our proposed model, we have compared our method with other domain adaptation techniques like Kernel Mean Matching [Huang et al., 2006], Subspace Alignment [Fernando et al., 2013] and Landmark based Kernelized Subspace Alignment [Rahaf et al., 2015].

We have applied all these algorithms on supernova data and many experiments were performed using different parameters to generate the best model. The models generated

were tested with supernova photometric data and classification accuracy was calculated to compare with our proposed method.

4.4 Kernel Mean Matching

Kernel Mean Matching domain adaptation method belongs to instance reweighting methods of domain adaptation. In this type of method, the source instances, which are closer to the target instances, are given more importance by providing weights to those instances. In instance-based methods of domain adaptation, weights are calculated using distribution estimation. However, it is different in the case of Kernel Mean Matching. First, we determine the kernel matrix for both source and target data. With the help of kernel matrix few parameters are calculated by solving the simple quadratic programming for Beta, β values. Beta values are the weights provided to the source data, and the model is built on weighted source data. The main idea behind Kernel mean matching is to minimize the maximum mean discrepancy between source and target data by projecting it into kernel space.

The implementation code for the Kernel mean matching is already available on the author's page. We had modified the code to perform various experiments. Experiments were performed using different kernel types for instance: Polynomial degree 1, Polynomial degree 2, Gaussian kernel 0.5 and Gaussian kernel 0.6. The predictive model generated was evaluated by testing on target data. We had used different classifiers for the testing purpose, and the classification accuracy was calculated using predicted target labels. The result of the experiments performed on this algorithm is as follows:

Kernel	***Classifier***	***Classification Accuracy (%)***
Polynomial degree 1	*MNN*	*72.04*
	RF	***75.95***
	SVM1	*63.46*
	SVM2	*68.47*
Polynomial degree 2	*MNN*	*60.98*
	RF	*69.3*
	SVM1	*57.92*
	SVM2	*61.45*
Gaussian 0.5	*MNN*	*72.86*
	RF	*73.85*
	SVM1	*59.92*
	SVM2	*67.38*
Gaussian 0.6	*MNN*	*75.07*
	RF	*72.12*
	SVM1	*64.31*
	SVM2	*65.71*

Table 4.4: Classification Accuracy on Supernova test data Kernel Mean Matching DA algorithm

The classification was performed using Weka with default parameters provided for all the classifiers. , MNN = Multilayer Neural Network, RF = Random Forest, SVM1 = Support Vector Machine Kernel 1, SVM2 = Support Vector Machine Kernel 2. It is clear from the above table that Random Forest performed best overall with the kernel Polynomial degree 1. However, Multilayer Neural Network was very close to Random Forest and it performed better than Random Forest for Gaussian Kernel with sigma 0.6.

4.5 Unsupervised Subspace Alignment

Subspace alignment [Fernando et al., 2013] domain adaptation technique is one of the popular domain adaptation methods falls under the category of feature-based methods in domain adaptation. The idea behind this approach is to decrease the discrepancy between two domains by moving the source and target subspace closer. In this method, they learn a transformation matrix that transforms the source subspace coordinate system to the target subspace coordinate system. Alignment between the two coordinate systems is achieved by aligning the source basis vectors with target basis vectors. Let S represents the source data, T represents the target data and d represents the subspace of dimension d. Thus, the algorithm can be given as follows:

Input: Source data S, Target data T, Source labels L_S, Subspace dimension d

Output: Predicted target labels L_T

$X_s \leftarrow$ PCA (S, d)
$X_T \leftarrow$ PCA (T, d)
$X_a \leftarrow X_a X_S' X_T$
$S_a = SX_a$
$T_T = TX_T$
$L_T \leftarrow$ Classifier (S_a, T_T, L_S)

We had implemented this algorithm using Matlab. The principal components for both source and target data were determined using Weka. Experiments were performed on principal components to determine optimal subspace dimension d. We had chosen d= 16 on the basis of maximum variance. Different models were generated using different classifiers. Classification accuracies were calculated by testing the models on target data using Weka. The result of the subspace alignment DA algorithm is as follows:

Classifier	***Classification Accuracy (%)***
Multilayer Neural Network	*67.2*
Random Forest	*61.25*
Support Vector Machine Kernel 1	*60.55*
Support Vector Machine Kernel 2	*61.04*
Support Vector Machine Kernel 3	*65.99*

Table 4.5: Classification Accuracy on Supernova test data using Subspace Alignment DA algorithm

It is evident from the above table that Multilayer Neural Network has better classification accuracy in comparison to other classifiers followed by Support Vector Machine Kernel 3 with only slight difference in the accuracy.

4.6 Landmarks-based Kernelized Subspace Alignment

This algorithm is the most recent one proposed by [Rahaf et al., 2015]. The algorithm is based on two steps: the first step is the selection of landmarks and the second step is the application of kernelized subspace alignment on the selected landmarks. Landmarks are the set of points, which can be used to project the source and target data in a shared space where their distributions are similar. The first part of our proposed method is based on this algorithm. Thus, the selection of landmarks has already been explained in detail in Chapter 3 section 3.3. After the selection of landmarks A with the function *select_landmarks,* all the points in source and target data is mapped non-linearly to the common space defined by the landmarks using Gaussian kernel. The value of standard deviation for the Gaussian kernel is selected by calculating the median distance between any pair of points randomly chosen from source and target data. Next step after non-

linear mapping is the application of subspace alignment to the kernels of source and target data. Let S denotes the source data, T indicates the target data, th is the threshold for overlapping function, A represents the landmarks, d is the subspace dimension and σ denotes the standard deviation of the Gaussian kernel. Algorithm for the Landmark based kernelized subspace alignment is as follows:

Input: Source data S, Target data T, Source labels L_S, Threshold th
Subspace dimension d
Output: L_T Predicted target labels
$A \leftarrow select\_landmarks\ (S, T, th)$
$\sigma \leftarrow median\_distance\ (S \cup T)$
$K_S \leftarrow project\_using\_kernel\ (S,A,\sigma)$
$K_T \leftarrow project\_using\_kernel\ (T,A,\sigma)$
$X_S \leftarrow PCA\ (K_S, d)$
$X_T \leftarrow PCA\ (K_T, d)$
$M \leftarrow X_S' X_T$
$P_S \leftarrow K_S X_S M$
$P_T \leftarrow K_T X_T$
$classifier \leftarrow learn\_classifier\ (P_S, L_S)$
$L_T \leftarrow classifier\ (P_T)$

We had implemented this algorithm using Matlab. There were two parameters s, neighborhood radius and th, threshold of overlapping function were required to determine for the selection of landmarks in source and target data. Thus, we selected the values of s similar to our proposed method for this algorithm i.e. s = 0.5,1,5,10 and th =0.9. After the selection of landmarks from source and target data, Subspace Alignment algorithm was applied on landmarks. The subspace dimension d was selected by maximum variance on the principal components of the kernels. The training data and testing data obtained after the application of Subspace Alignment was then fed to

different classifiers to generate models. Classification accuracies were calculated by testing the target data on different models. Following are the result of the experiments performed on this algorithm:

Landmarks for s=	***Classification Accuracy (%)***				
	MNN	***RF***	***SVM1***	***SVM2***	***SVM3***
0.5	*37.29*	*69.44*	*25.53*	*34.12*	*22.59*
1	*23.24*	*70.56*	*25.31*	*22.53*	*22.51*
5	*67.42*	***71.30***	*68.93*	*66.43*	*35.35*
10	*30.44*	*68.29*	*23.42*	*28.59*	*23.45*

Table 4.6: Classification Accuracy on Supernova test data using Landmark-based kernelized Subspace Alignment DA algorithm

In the above Table 4.6, MNN = Multilayer Neural Network, RF = Random Forest, SVM1 = Support Vector Machine Kernel 1, SVM2 = Support Vector Machine Kernel 2, SVM3 = Support Vector Machine Kernel 3. As it can be seen from the table that Random Forest had performed better as compared to other classifiers. Surprisingly, all other classifiers apart from Random Forest had failed miserably to classify supernova photometric data.

4.7 Comparison of results of domain adaptation methods

Results from different domain adaptation algorithms on supernova data are compared with our proposed method. We have compared with best classification accuracy obtained from all the algorithms on supernova photometric data. However, we have also determined the classification accuracy on raw supernova photometric data with 108 features. Raw supernova photometric data refers to the data without pre-processing and domain adaptation algorithm. In addition to it, we have calculated the classification accuracy on supernova photometric data after the application of Principal Component

Analysis and Kernel Principal Component Analysis. We have listed out these classification accuracies for comparing with our method.

The comparison table of classification accuracies for different algorithms on supernova photometric data is as follows:

Different algorithms on Supernova data	***Classification accuracy on Supernova Photometric Data (%)***
Raw Supernova data (108 features)	*68.44*
Supernova with PCA (16 features)	*72.19*
Supernova with KPCA (23 features)	*74.35*
Kernel Mean Matching DA method	*75.95*
Subspace Alignment DA method	*67.2*
Landmarks-based Kernelized Subspace Alignment DA method	*71.30*
Domain Adaptation using Landmarks and Active Learning (our method)	***79.40***

Table 4.7: Comparison of different algorithms on Supernova Photometric Data

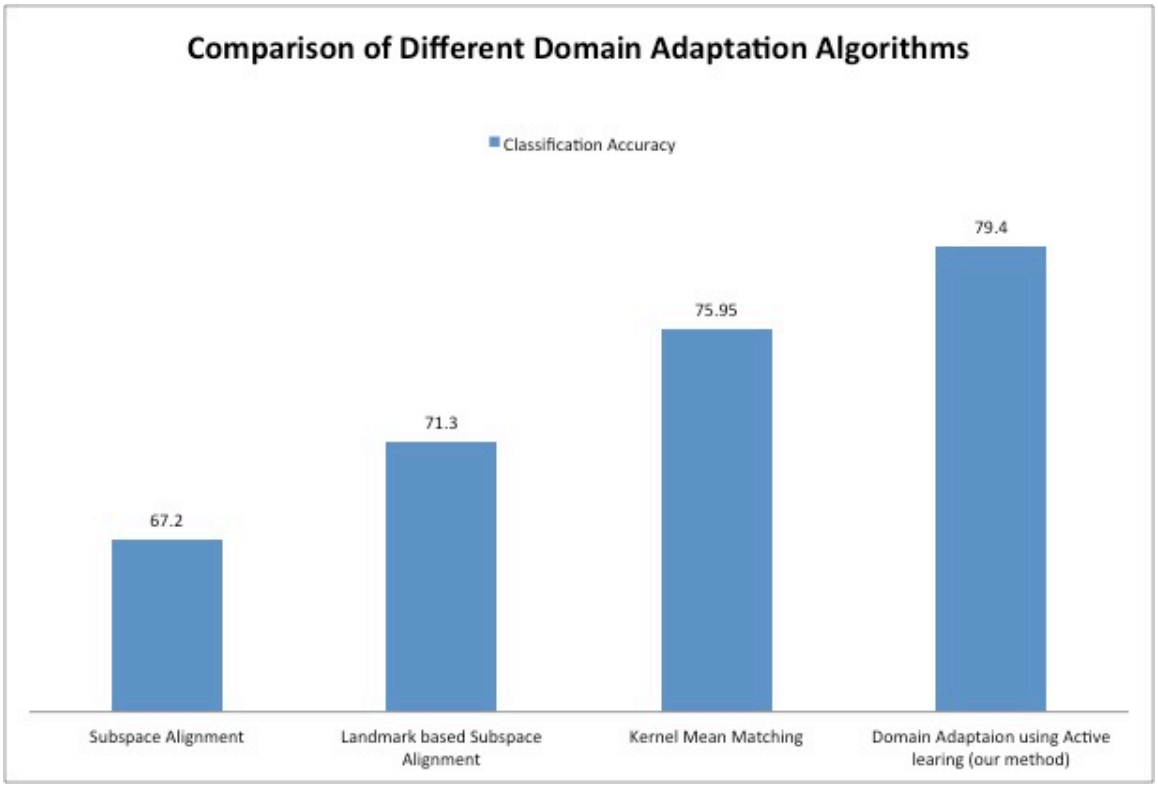

Figure 4.2: Comparison of Different Domain Adaptation Algorithms

It is apparent from the Figure 4.2 that our method performed considerably well as compared to other domain adaptation algorithms. If we compare classification accuracy of our method with raw supernova data, it has been increased by nearly 11%. However, the classification accuracy of Kernel Mean Matching is close to our method as compared to other domain adaptation algorithms. Subspace Alignment DA algorithm performed worst overall with least classification accuracy among others. The reason behind the better working our method over other domain adaptations algorithms is that our method works more locally in the data. Since, the supernova datasets are highly overlapping, it requires a more local approach rather than global performed by other DA algorithms.

Chapter 5

Future Work and Conclusion

5.1 Limitations and Future work

Even though our proposed method has improved the classification accuracy but still there is a space for improvements. One of the major limitations of the proposed method is that there is no limit in the *maxCost* (the number of target candidate labels that can be queried with an expert during active learning). The number of *maxCost* is increasing exponentially with the increase in the number of clusters. Thus, we could devise a strategy to use maxCost efficiently among clusters if it is limited in number.

Another major limitation is that there is a possibility to get pure clusters of target candidate. Building a model on the pure cluster is useless. Thus, identifying a pure cluster without label information could be a challenge.

Also, the number of samples in source data is minimal as compared to target data, i.e., 718:11946. Our method can work better if we include more source information in future.

There is a space for different sampling strategy of the target candidates for active learning apart from margin sampling that could have been an advantage not explored in this thesis. We have only used Multilayer Neural Network for learning model, for the calculation of margin using posterior probabilities and for the classification purpose. Instead of Multilayer Neural Network, we could use other machine learning algorithms that can provide posterior probabilities for the calculation of margins.

Also, we have used EM algorithm for the clustering purpose; comparison of different clustering algorithms can be done for the better performance.

Another direction that can be taken in future is that, from the classification of different classes, it is observed that Supernova Type Ic has classified most of the times correctly. By analyzing the histogram of the supernova data by class distribution, it is found that Supernova Type Ic is entirely separable from other two classes. But there is an ambiguity between Supernova Type Ia and Supernova Type Ib samples. They are not well separated. Experiments could be done for separating Supernova Type Ia and Supernova Type Ib after completing removing Supernova Type Ic by two-stage classification process. The exercise presented here in a simulated context might be used as a guide for survey stratifies once we identify more closely the properties of target data used in training as a receipt for follow up.

5.2 Conclusion

The primary focus of this research is to find an automated way for the classification of supernova. Automation is necessary to replace the existing method of classification, based on spectroscopy method that is very expensive and time-consuming. With the wide range of telescopic surveys in the future, significant number of supernova samples is expected. An automated way of classification will help the astronomers to conduct their research efficiently and can help in future discoveries. Astronomers also wanted to take advantage of existing dataset of supernova classified with the help of spectroscopic data for devising an automated method. With the help of domain adaptation and active learning it is possible to generate a predictive model that can classify supernova types automatically with good classification accuracy.

In this thesis, we have reduced the dimension of the supernova dataset by applying different dimension reduction techniques: Principal Component Analysis, Kernel Principal Component Analysis, and Discrete Wavelet Transform. Experiments are

performed to determine the best dimension reduction technique. Our method of domain adaptation using active learning is applied to the reduced feature set of supernova to generate the predictive model. To evaluate our method, we have compared our result with the existing domain adaptation algorithms: Kernel Mean Matching, Subspace Alignment and Landmark based Kernelized Subspace Alignment. Analysis of the result has shown that our methodology has performed well by giving better classification accuracy as compared to other existing algorithms.

References

[1] Arlot S., Celisse A., 2010, Statistics Surveys, 4, 40 Astier P., Guy J., Regnault N., Pain R., Aubourg E., Balam D., Basa S., Carlberg R. G., et al. 2006, A&A, 447

[2] Huang J, Gretton A, Borgwardt KM, Schölkopf B, Smola AJ. Correcting sample selection bias by unlabeled data. InAdvances in neural information processing systems 2006 (pp. 601-608).

[3] Shimodaira, Hidetoshi. "Improving predictive inference under covariate shift by weighting the log-likelihood function." *Journal of statistical planning and inference* 90, no. 2 (2000): 227-244.

[4] Japkowicz, Nathalie, and Shaju Stephen. "The class imbalance problem: A systematic study." *Intelligent data analysis* 6, no. 5 (2002): 429-449.

[5] Bickel, Steffen, Michael Brückner, and Tobias Scheffer. "Discriminative learning for differing training and test distributions." In *Proceedings of the 24th international conference on Machine learning*, pp. 81-88. ACM, 2007.

[6] Fernando, Basura, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. "Unsupervised visual domain adaptation using subspace alignment." In*Proceedings of the IEEE International Conference on Computer Vision*, pp. 2960-2967. 2013.

[7] Blitzer, John, Mark Dredze, and Fernando Pereira. "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification." In*ACL*, vol. 7, pp. 440-447. 2007.

[8] Gopalan, Raghuraman, Ruonan Li, and Rama Chellappa. "Domain adaptation for object recognition: An unsupervised approach." In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 999-1006. IEEE, 2011.

[9] Aljundi, Rahaf, Rémi Emonet, Damien Muselet, and Marc Sebban. "Landmarks-based kernelized subspace alignment for unsupervised domain adaptation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 56-63. 2015.

[10] Bruzzone, Lorenzo, and Mattia Marconcini. "Domain adaptation problems: A DASVM classification technique and a circular validation strategy." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32, no. 5 (2010): 770-787.

[11] Settles, Burr. "Active learning literature survey." *University of Wisconsin, Madison* 52, no. 55-66 (2010): 11.

[12] Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller. "Kernel principal component analysis." In *Artificial Neural Networks—ICANN'97*, pp. 583-588. Springer Berlin Heidelberg, 1997.

[13] Wold, Svante, Kim Esbensen, and Paul Geladi. "Principal component analysis." *Chemometrics and intelligent laboratory systems* 2, no. 1-3 (1987): 37-52.

[14] Qu, Yinsheng, Bao-ling Adam, Mark Thornquist, John D. Potter, Mary Lou Thompson, Yutaka Yasui, John Davis et al. "Data reduction using a discrete wavelet transform in discriminant analysis of very high dimensionality data."*Biometrics* 59, no. 1 (2003): 143-151.

[15] Mallat, Stephane G. "A theory for multiresolution signal decomposition: the wavelet representation." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 11, no. 7 (1989): 674-693.

[16] Kessler, Richard, Bruce Bassett, Pavel Belov, Vasudha Bhatnagar, Heather Campbell, Alex Conley, Joshua A. Frieman et al. "Results from the supernova photometric classification challenge." *Publications of the Astronomical Society of the Pacific* 122, no. 898 (2010): 1415.

[17] Ishida, Emille EO, and Rafael S. de Souza. "Kernel PCA for Type Ia supernovae photometric classification." *Monthly Notices of the Royal Astronomical Society* 430, no. 1 (2013): 509-532.

[18] Daumé III, H., Kumar, A. and Saha, A., 2010, July. Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing* (pp. 53-59). Association for Computational Linguistics.

[19] Gopalan, Raghuraman, Ruonan Li, and Rama Chellappa. "Domain adaptation for object recognition: An unsupervised approach." In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 999-1006. IEEE, 2011.

[20] Karpenka, Natalia V., F. Feroz, and M. P. Hobson. "A simple and robust method for automated photometric classification of supernovae using neural networks." *Monthly Notices of the Royal Astronomical Society* (2012): sts412.