PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

AUTOMATIC SURVEY-INVARIANT CLASSIFICATION OF VARIABLE STARS

PATRICIO BENAVENTE ESCANDÓN

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
KARIM PICHARA BAKSAI

Santiago de Chile, January 2018

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

AUTOMATIC SURVEY-INVARIANT CLASSIFICATION OF VARIABLE STARS

PATRICIO BENAVENTE ESCANDÓN

Members of the Committee:
KARIM PICHARA BAKSAI
ALVARO SOTO ARRIAZA
ANDREAS REISENEGGER VON OEPEN
PAVLOS PROTOPAPAS
LUIS RIZZI CAMPANELLA

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, January 2018

In loving memory of Mariana Cash

ACKNOWLEDGEMENTS

This thesis—and by extension all my personal, academic and professional work and achievements so far—is dedicated to the memory of my grandmother, Mariana Georgina Cash Molina. Her unconditional and unwavering love, support, containment, and sacrifice has strengthened and allowed me to learn and grow throughout my childhood and university years.

I want to thank my advisor Karim Pichara for his guidance, teaching, and support through my master's program. I also want to thank him for trusting me with this research project, and giving me amazing opportunities in collaboration with other researchers and universities.

I also want to thank my co-advisor Pavlos Protopapas for his great mentorship and encouragement to give the best of myself. His involvement and commitment—specially during my stay at Harvard—was instrumental to the realization of this work.

I would like to thank my mother, Verónica Escandón Martínez, for her abnegation and love raising my siblings and me. My father, Patricio Benavente Cash, for his love and for sparking my interest in mathematics and computer science as a child. I also want to thank my sister Francisca Benavente, and my brother Nicolás Benavente.

I want to express my profound gratitude to my uncle René Benavente and my aunt Cecilia Escandón for their selfless financial support during my university and high school years. Their help allowed me to concentrate better on my academic work, relieving the financial pressure.

Special thanks to my friend Isadora Nun for her constant support and encouragement finishing this work, for relentlessly pushing me forward when I wanted to give up, and for being my "link back home" abroad.

Special thanks as well to Ivania Donoso, Lucas Valenzuela, and Belén Saldías for their incredible support helping me prepare the thesis presentation in a constrained time-frame and under difficult personal circumstances—I could not have done it without them.
I also want to thank them, Nebil Kawas, and Rodrigo Saffie for their friendship and support through our university years, and for helping me navigate the incomprehensible bureaucracies of the program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Machine learning techniques have been successfully used to classify variable stars on widely-studied astronomical surveys. These datasets have been available to astronomers long enough, thus allowing them to perform deep analysis over several variable sources and generating useful catalogs with identified variable stars. The products of these studies are labeled data that enable supervised learning models to be trained successfully. However, when these models are blindly applied to data from new sky surveys their performance drops significantly. Furthermore, unlabeled data becomes available at a much higher rate than its labeled counterpart, since labeling is a manual and time-consuming effort. Domain adaptation techniques aim to learn from a domain where labeled data is available — the source domain — and through some adaptation perform well on a different domain – the target domain. We propose a full probabilistic model that represents the joint distribution of features from two surveys as well as a probabilistic transformation of the features between one survey to the other. This allows us to transfer labeled data to a study where it is not available and to effectively run a variable star classification model in a new survey. Our model represents the features of each domain as a Gaussian mixture and models the transformation as a translation, rotation and scaling of each separate component. We perform tests using three different variability catalogs: EROS, MACHO, and HiTS, presenting differences among them, such as the amount of observations per star, cadence, observational time and optical bands observed, among others.

Keywords: Astronomy, Variable Stars, Machine Learning, Transfer Learning, Domain Adaptation.

RESUMEN

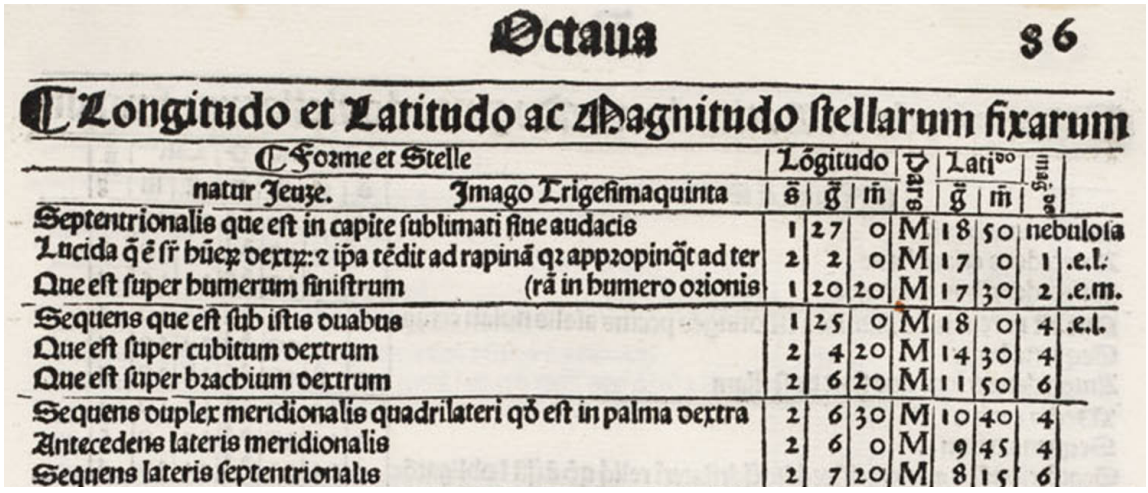
Las técnicas de aprendizaje de máquina han sido aplicadas con éxito en la clasificación de estrellas variables en sondeos astronómicos bien estudiados. Estos conjuntos de datos han estado disponibles el tiempo suficiente para que los astrónomos analicen en profundidad una serie de fuentes variables y generen catálogos prácticos con estrellas variables identificadas. El producto de estos estudios son datos etiquetados que permiten entrenar modelos supervisados con éxito. Sin embargo, cuando estos modelos son aplicados ciegamente a datos provenientes de nuevos sondeos celestes su desempeño disminuye de manera considerable. Más aún, los datos sin etiqueta son generados a una tasa muchísimo mayor que la de su contraparte etiquetada, ya que el etiquetado es un proceso manual que toma tiempo. Las técnicas de adaptación de dominio apuntan a aprender en un dominio donde hay etiquetas disponibles — el dominio fuente — y mediante alguna adaptación clasificar con éxito en otro dominio — el dominio objetivo. Proponemos un modelo probabilístico completo que representa la distribución conjunta de las características de dos conjuntos de datos distintos, así como una transformación probabilística desde las características de uno de los conjuntos de datos hacia el otro. Esto permite transferir datos etiquetados a un sondeo donde éstos no están disponibles y efectivamente aplicar un modelo de clasificación en un sondeo nuevo. Nuestro modelo representa las características de cada dominio como una mezcla de Gaussianas y modela la transformación como una translación, rotación y escalación de cada componente por separado. Realizamos pruebas usando tres catálogos de variabilidad diferentes: EROS, MACHO y HiTS. Presentamos las diferencias entre ellos, como la cantidad de observaciones por estrella, cadencia, tiempo de observación, y bandas ópticas observadas, entre otros.

Palabras Claves: Astronomía, Estrellas Variables, Aprendizaje de Máquina, Transferencia de Aprendizaje, Adaptación de Dominio.

1. INTRODUCTION

1.1. The Big Data Bang

Astronomy is perhaps one of the most notable examples of the "data explosion" phenomenon in science. Astronomical datasets have been growing in size at an accelerating pace since the appearance of the Babylonian star catalogs around 1370 BCE, some of the oldest known to date. These ancient catalogs contain precise measurements of over 70 stars (Rogers, 1998). More than 1,000 years later, around 126 BCE, the Greek astronomer Hipparchus of Nicaea compiled his star catalog, which included more than 653 stars. He introduced the apparent magnitude scale used to measure the brightness of celestial objects that is still in use today (Graßhoff, 2013). Ptolemy would then expand on Hipparchus work in the Almagest or Syntaxis Mathematica — one of the most important scientific texts of all time, which introduced the geocentric model — documenting Hipparchus' star tables and his work on trigonometry (Graßhoff, 2013). Figure 1.1 shows part of a star table from the Almagest.



Octaua 86

Longitudo et Latitudo ac Magnitudo stellarum fixarum

Forme et Stelle natur Ieuze. Imago Trigesimaquinta	Lōgitudo s	g	m	Partes	Latio g	m	Magnitudo	
Septentrionalis que est in capite sublimati siue audacis	1	27	0	M	18	50	nebulosa	
Lucida q̄ ē sr̄ būex dextr: ꝛ ipa tēdit ad rapinā qꝛ appꝛopinq̄t ad ter	2	2	0	M	17	0	1	.e.l.
Que est super humerum sinistrum (rā in humero oꝛionis	1	20	20	M	17	30	2	.e.m.
Sequens que est sub istis duabus	1	25	0	M	18	0	4	.e.l.
Que est super cubitum dextrum	2	4	20	M	14	30	4	
Que est super bꝛachium dextrum	2	6	20	M	11	50	6	
Sequens duplex meridionalis quadrilateri qō est in palma dextra	2	6	30	M	10	40	4	
Antecedens lateris meridionalis	2	6	0	M	9	45	4	
Sequens lateris septentrionalis	2	7	20	M	8	15	6	

Figure 1.1. Star table from a 1515 print of the Almagest (Ptolemei, 1515). The table shows star designations, positions in ecliptic coordinates and magnitudes. Source: Linda Hall Library of Science, Engineering & Technology (http://lhldigital.lindahall.org/cdm/ref/collection/astro_images/id/1700).

The greatest increases in the size of astronomical datasets have been tied to technological advancement, which has allowed the observation of the skies beyond the capabilities of the naked eye. Galileo Galilei's telescope, constructed in 1609, started the era of optical astronomy. Following this invention, ever larger and more precise telescopes have been constructed. Figure 1.2 shows various optical telescope aperture diameters by the year they were built.

The XVII century also saw the first well-organized sky survey when the English astronomer John Flamsteed cataloged around 3,000 stars. In the late XIX century, Henry Draper funded the first spectroscopic star survey, which cataloged around 300,000 stars.

Table 1.1. Data volume of different astronomical surveys as presented by Y. Zhang and Zhao (2015).

Astronomical Survey	**Data Volume**
DPOSS (The Palomar Digital Sky Survey)	3 TB
2MASS (The Two Micron All-Sky Survey)	10 TB
GBT (Green Bank Telescope)	20 PB
GALEX (The Galaxy Evolution Explorer)	30 TB
SDSS (The Sloan Digital Sky Survey)	40 TB
SkyMapper Southern Sky Survey	500 TB
PanSTARRS (The Panoramic Survey Telescope and Rapid Response System)	40 PB expected
LSST (The Large Synoptic Survey Telescope)	200 PB expected
SKA (The Square Kilometer Array)	4.6 EB expected

The advent of digital surveys, automated telescopes and online catalogs brought astronomy to the big data era. The Sloan Digital Sky Survey (SDSS), designed in 1990, surveyed on the visible spectrum one-third of the sky obtaining positions and brightness of a billion stars, galaxies, and quasars, as well as the spectra of a million objects. Still active today, it generates around 200 GB of data every night, accumulating more than 40 TB of data to date (Feigelson & Babu, 2012). The Large Synoptic Survey Telescope (LSST), currently under construction in Chile, is expected to generate an average of 15 Terabytes of data per night upon entering operations in 2022 (Jurić et al., 2015). The Square Kilometer Array (SKA), a multi radio telescope project, would generate an estimated 4.6 EB

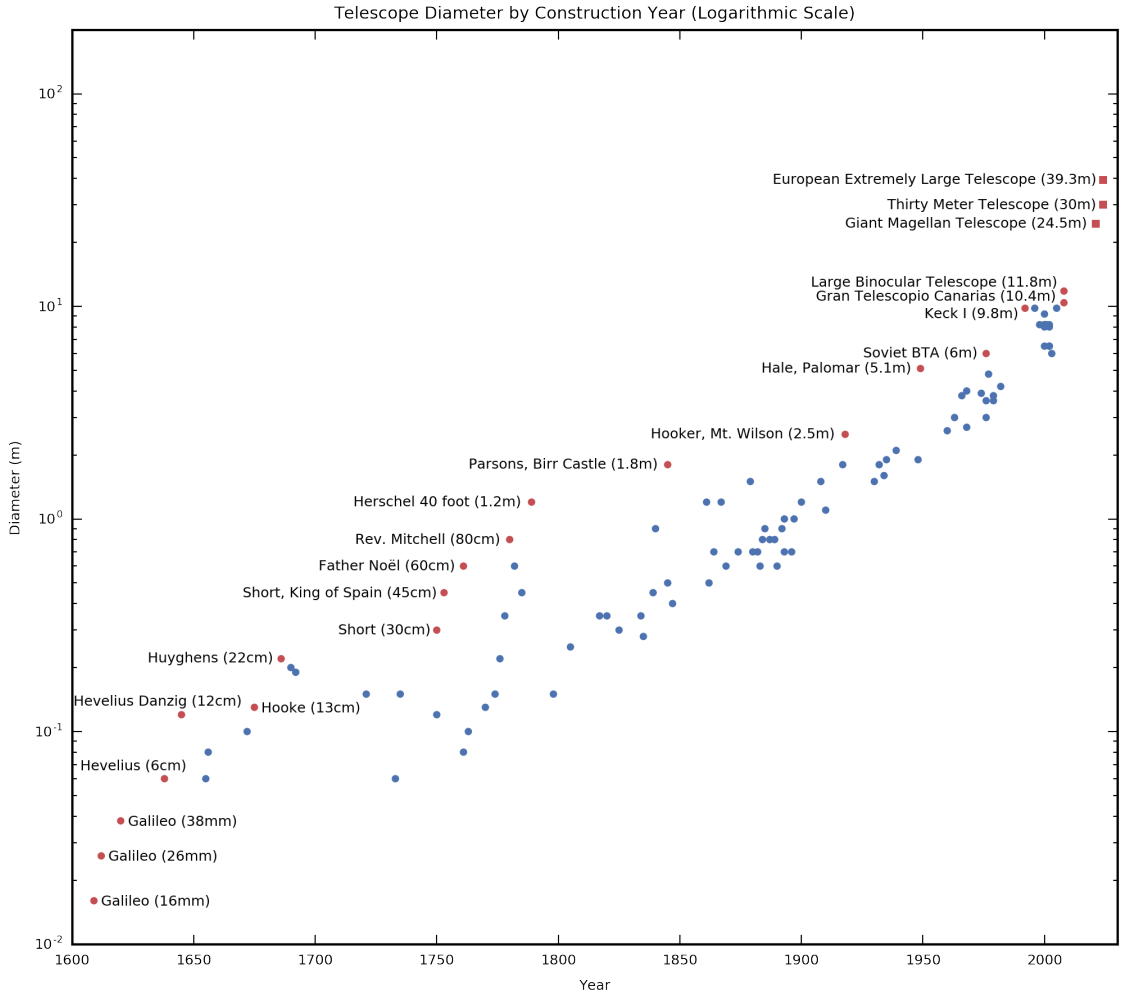


Figure 1.2. Optical telescope aperture diameter by construction year. The aperture of a telescope is the diameter of its main light-gathering lens or mirror. The diameter axis is in logarithmic scale. Telescopes under construction are shown with a square marker. Based on data by Racine (2004), and updated with data from GRANTECAN S.A. (2013), European Southern Observatory (2014), TMT Observatory Corporation (2007), LBTO (2016) and GMTO Corporation (2016).

of data in total (Y. Zhang & Zhao, 2015). Table 1.1 shows the estimated volume of data generated by past and planned stellar surveys.

Traditional analysis techniques do not scale with this myriad amount of data. As a solution to this, machine learning methods have been applied with success to astronomy

problems such as classification of galaxy morphology (Freed & Lee, 2013), spectral classification (Christlieb, Wisotzki, & Grasshoff, 2002; Bromley, Press, Lin, & Kirshner, 1998), photometric classification (Cavuoti, Brescia, D'Abrusco, Longo, & Paolillo, 2014; Brescia, Cavuoti, D'Abrusco, Longo, & Mercurio, 2013), solar activity prediction (Colak & Qahwaji, 2009), photometric redshift regression (Benitez, 2000; Collister & Lahav, 2004), anomaly detection (Nun, Pichara, Protopapas, & Kim, 2014) and variable star classification (Pichara & Protopapas, 2013; Pichara, Protopapas, & León, 2016; Blomme et al., 2011).

1.2. Astronomical Domain Shift

Astronomical datasets have different characteristics depending on the survey they were derived from. Indeed, filters and atmospheric conditions vary, and observations differ due to detector sensitivity and the depth observed, among other factors. For example, a deep survey may be more biased towards active galactic nuclei (AGNs) – the central, most bright, section of a galaxy – than a shallower survey, because it is able to detect objects farther away (such as galaxies). This means that models trained in one survey cannot be readily used in data generated from other surveys and must be retrained from scratch. Moreover, labeled data – needed for supervised classification – is unavailable for new surveys, since labeling must be done manually by trained astronomers in a time consuming effort (Sterken & Jaschek, 2005). Since applying a previously trained model to a new survey results in considerable losses in performance, this renders supervised learning on new datasets unfeasible.

The latter problem arises from the assumption, often taken in traditional learning techniques, that the distribution of the data used to train the model is the same as the distribution of the data to which the model is applied to. However, this assumption does not generally hold in practical applications.

Therefore, it is desirable to transfer the information learned by a classifier in a domain where labels abound — the source domain — to a domain where few or no labels are available — the target domain. This problem is known as domain adaptation (Jiang, 2008) and is part of the more general problem of transfer learning (Pan & Yang, 2010; Raina, Battle, Lee, Packer, & Ng, 2007) – applying knowledge generated solving one problem into another one.

1.3. Thesis Contribution

This thesis addresses the problem of domain adaptation in the context of variable star classification. Stars are classified as variable when their apparent brightness, as seen from Earth, measurably fluctuates over human timescales. Furthermore, variable stars are classified according to the nature of the brightness change, the time-scale and magnitude of the fluctuation, among other properties. Variable stars and their classification are introduced in more depth in Section 4.1.2.

In our problem, the source domain is a well-known astronomical survey — in which a relatively large amount of labels exist and a trained classifier can perform accurately — and the target domain is a newer or relatively less studied survey where no or very few labels exist.

To solve this, one may use the target dataset instances to induce a change in the source domain classifier that allows it to perform better in the target domain. This approach relies on the creation of an adaptation objective that effectively reduces the classification error on the target domain using no label information. On the other hand, one might find a transformation between the feature spaces of the source and target domains, which allows passing the instances in the target dataset to a representation suitable for training a new classifier. Moreover, this transformation can also be applied in the opposite direction; i.e. to transfer the labeled instances from the source domain to the feature space of the target domain and then train a classifier on this data. We favor the latter approach, as it is model

independent. A classifier modification would generally depend on a model's particularities — such as the way a discriminative classifier models class boundaries — while a feature space transformation has the advantage of being model agnostic, decoupling the adaptation and classification problems and thus allowing for the use of the best suited model in a given situation and applying new models as they become available.

We propose a new probabilistic model, based on the Gaussian mixture model (GMM). We use two GMMs to represent the feature distributions of the source and target domains. We then infer linear transformations of the GMM components. We assume that the statistical descriptor shift between the surveys can be corrected by translating, rotating and scaling the GMM components. Our method is unsupervised, as we only require unlabeled data in both domains. We estimate and apply a transformation to the labeled instances in the source domain for each of the mixture components, weighted by how much importance each component has on each data point. In this way, we build a training set suitable for classifying in the target domain. In doing this, we assume that the transformation that corrects the shift in the unlabeled dataset will also correct it in the training set.

Our approach offers some advantages compared to previous research:

(i) It finds a transformation from the feature space of one domain to the other, meaning that any data from one domain can be used as if belonging to the other. Other methods perform adaptation at the model level and leave data intact.

(ii) Since it makes no assumptions about the classifier, our approach is classifier agnostic. Transformed training sets can be used with any model of choice, effectively decoupling domain adaptation from model learning.

1.4. Thesis Overview

This document is based on the paper *Automatic Survey-invariant Classification of Variable Stars* by Patricio Benavente, Pavlos Protopapas and Karim Pichara. The paper was

published on the 21st of August, 2017 in The Astrophysical Journal, Volume 845, Number 2, Page 147 (Benavente, Protopapas, & Pichara, 2017).

The remainder of this thesis is organized as follows: Chapter 2 formalizes the problem and introduces relevant notation. Chapter 3 presents an overview of related previous work on the subject, and Chapter 4 briefly introduces the relevant background theory. Our method is described in depth in Chapter 5. Experimental results are presented and analyzed in Chapter 6. Section 6.1 outlines our experimental methodology. Section 6.2 illustrates the model's operation and results in simulated datasets. Results in real datasets are presented in Section 6.3. Finally, we conclude this work in Chapter 7.

2. PROBLEM DESCRIPTION AND NOTATION

We follow a notation similar to Jiang (2008). Let $\mathcal{X}$ be the feature space and $\mathcal{Y}$ the label space in our problem. The term feature refers to a property or characteristic that describes an object or phenomenon being observed (Bishop, 2006). Features should be informative and discriminative, in order to allow distinguishing between labels. In variable star classification, features may include statistical descriptors about the star's magnitude over time (such as the mean, standard deviation and skewness), or brightness periodicity characteristics derived from autoregressive models, among others (Nun et al., 2015).

Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be random variables representing the observed features and the observed class labels, respectively. We denote their true underlying joint distribution as $P(X, Y)$. We distinguish two domains: a source domain where a large amount of labeled data is available and a target domain where labeled data is unavailable or scarce. We denote the true joint distributions of X and Y in the source and target domains as $P_s(X, Y)$ and $P_t(X, Y)$, respectively. Consequently, we denote the true marginal distributions of X and Y for each domain as $P_s(X)$, $P_s(Y)$, $P_t(X)$ and $P_t(Y)$, and the true conditional distributions as $P_s(X|Y)$, $P_s(Y|X)$, $P_t(X|Y)$ and $P_t(Y|X)$, as one would expect.

Let $D_s^l = \left\{(x_i^{sl}, y_i^{sl})\right\}_{i=1}^{N_s^l} \subseteq \mathcal{X} \times \mathcal{Y}$ be the labeled data available in the source domain and $D_s^u = \{x_i^{su}\}_{i=1}^{N_s^u} \subseteq \mathcal{X}$ the available unlabeled data in the source domain. Similarly, let $D_t^u = \{x_i^{tu}\}_{i=1}^{N_t^u} \subseteq \mathcal{X}$ be the unlabeled data available in the target domain and $D_t^l = \left\{(x_i^{tl}, y_i^{tl})\right\}_{i=1}^{N_t^l} \subseteq \mathcal{X} \times \mathcal{Y}$ the labeled data in the target domain. We call a value $x \in \mathcal{X}$ an unlabeled instance and a tuple $(x, y) \in \mathcal{X} \times \mathcal{Y}$ a labeled instance.

Three types of domain adaptation problems are distinguished based on the kind of data available (Pan & Yang, 2010): (a) *Supervised domain adaptation* exploits labeled data both in the source and in the target domain, (b) *unsupervised domain adaptation* uses only unlabeled data, and (c) *semi-supervised domain adaptation* employs only a small amount of labeled data from the target domain.

We focus on the unsupervised domain adaptation problem, therefore we will generally ignore the labeled data in the target domain, D_t^l, and use it for testing purposes only.

In our problem, the true joint distributions differ between the two domains: $P_s(X, Y) \neq P_t(X, Y)$. Additionally, several different scenarios can be considered under the domain adaptation problem depending on the assumptions made about the cause of the joint distribution difference between the domains.

2.1. Covariate Shift

If we assume that the causal relationships between X and Y remain the same and that the only difference in the joint distributions arises from the marginal distribution of the covariates – that is $P_s(Y|X) = P_t(Y|X)$ and $P_s(X) \neq P_t(X)$ – then the problem is known as *covariate shift* or *sample selection bias* (Shimodaira, 2000; Huang, Gretton, Borgwardt, Schölkopf, & Smola, 2006). This scenario applies whenever there is a bias in the data selection procedure. For example, consider two different telescopes of which one is equipped with a detector with higher sensitivity than the other. The data captured by the more sensitive telescope will be more biased towards dimmer objects than the one captured by the less sensitive telescope. However, the characteristics of the celestial objects do not change, that is, $P(Y|X)$ is the same. Figure 2.1 shows covariate shift between a sample of the EROS and HiTS datasets when looking at the mean apparent magnitude and the Psi_CS feature from the FATS package (Nun et al., 2015). These features are described in table 6.2.

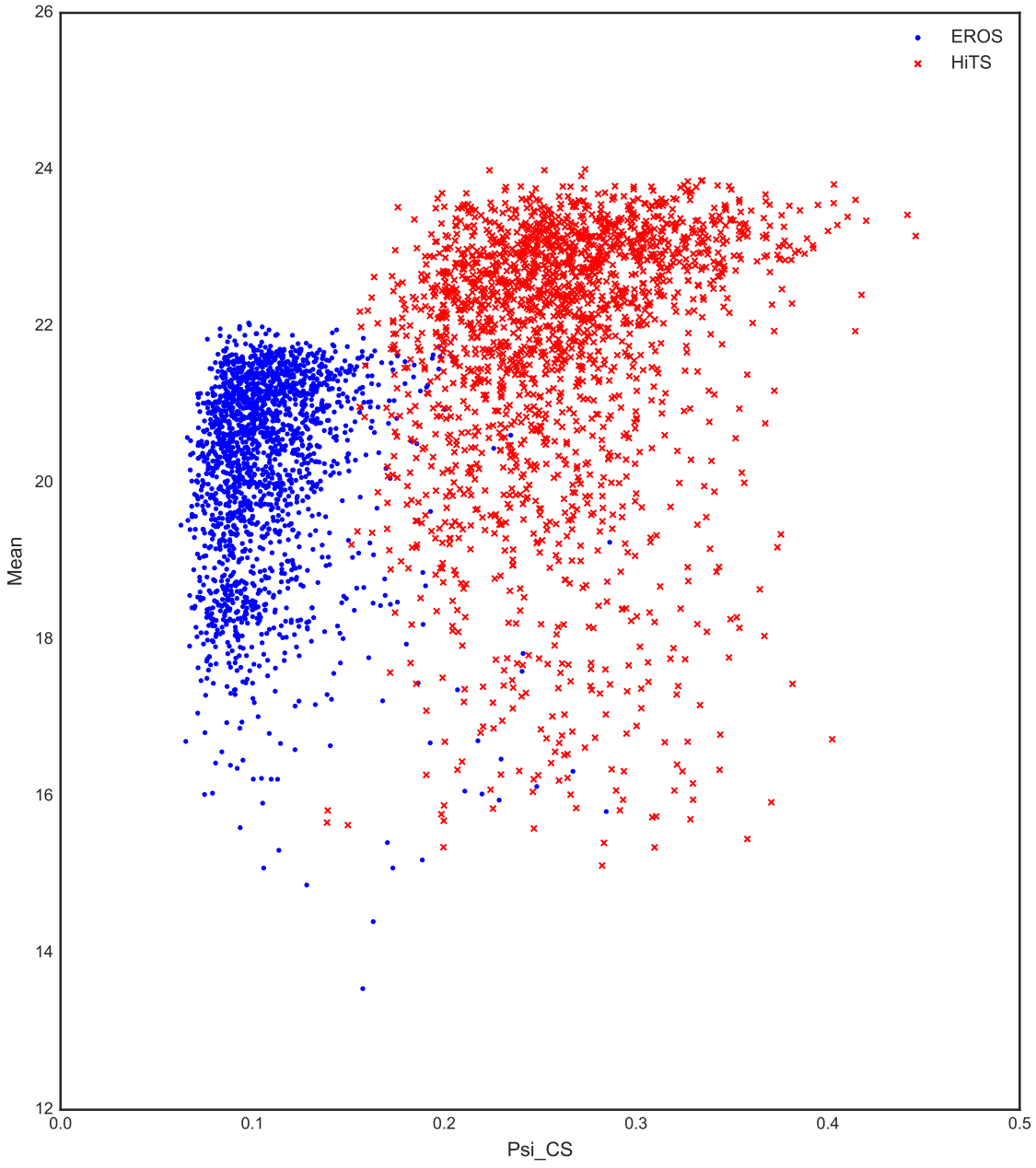


Figure 2.1. Covariate shift between EROS and HiTS datasets. The HiTS survey is more biased toward dimmer objects than EROS. See table 6.2 for a description of the axes.

2.2. Target, Conditional and Generalized Target Shift

K. Zhang, Muandet, and Wang (2013) identify three distinct scenarios that arise if we assume that $P_s(Y|X) \neq P_t(Y|X)$. By virtue of Bayes' theorem, this difference is produced by the marginals $P(Y)$ or the conditionals $P(X|Y)$ being different, or both.

If the marginal distributions of the classes change and the conditional distributions of the features given the classes stay the same – that is $P_s(Y) \neq P_t(Y)$ while $P_s(X|Y) = P_t(X|Y)$ – then the problem is known as *target shift* (TarS) (K. Zhang et al., 2013), *class imbalance* (Patel, Gopalan, Li, & Chellappa, 2015) or *prior probability shift* (Quionero-Candela, Sugiyama, Schwaighofer, & Lawrence, 2009). This scenario arises whenever a class is more present in one domain than in the other. For example, in the problem of medical diagnosis prediction, one is interested in predicting diseases given symptoms. Disease prevalence varies across geographical locations, as such some diseases that are common in tropical regions will be rare in areas close to the poles and class distributions will be different (but the probability of the symptoms given the disease remains constant). In astronomy, if the sensitivity of the telescope changes significantly, then we see objects such as distant galaxies that are not in the other dataset. Note that covariate shift would also be present in this scenario.

Conversely, the problem is known as *conditional shift* (ConS) (K. Zhang et al., 2013) if the conditional distribution of the features given the classes changes and the marginal distribution of the classes stays the same – that is $P_s(Y) = P_t(Y)$ and $P_s(X|Y) \neq P_t(X|Y)$. ConS appears when the causal relationship of one or all the classes in relation to the features changes. For example, some diseases manifest symptoms differently depending on the patient's gender. The probability of having nausea given that the patient is suffering a heart disease will be higher if female patients are being diagnosed. In astronomy, variability may appear in some part of the electromagnetic spectrum. For example, a star may be variable in the ultraviolet, but not in the optical spectrum. If we are considering the mean

apparent magnitude in the band observed by the telescope, then the variability of certain objects would appear in some surveys and not in others.

Finally, if both distributions change, meaning that $P_s(Y) \neq P_t(Y)$ and $P_s(X|Y) \neq P_t(X|Y)$, the problem is known as *generalized target shift* (GeTarS) (K. Zhang et al., 2013). This situation arises when both TarS and ConS are present.

Our research focuses on the domain adaptation problem under covariate and generalized target shift. Therefore, we do not assume that any of the marginal probability distributions are the same. Our goal is to find a transformation from the source feature space to the feature space of the target domain, in order to adapt the source labeled instances to a representation suitable for training a classifier that performs well in the target domain. By doing this, we assume that the transformation that corrects the shift in the unlabeled dataset will also correct it in the training set.

3. RELATED WORK

Domain adaptation has been studied extensively in the contexts of natural language processing (NLP) (Foster, Goutte, & Kuhn, 2010; Blitzer, McDonald, & Pereira, 2006) and computer vision (Gong, Shi, Sha, & Grauman, 2012; Gopalan, Li, & Chellappa, 2011; Patel et al., 2015).

A popular approach for domain adaptation is known as *instance weighting* or *importance reweighting* (Shimodaira, 2000; Foster et al., 2010). In instance weighting, the terms of the loss function corresponding to each sample are weighted by the relative density $\frac{P_t(x,y)}{P_s(x,y)}$, which effectively minimizes the expected loss of the model over the target distribution (Jiang, 2008). However, it is generally not possible to calculate this value and the support of the source distribution must be contained in that of the target distribution for this to work in practice. In the covariate shift scenario, this weight can be simplified to $\frac{P_t(x)}{P_s(x)}$ (Shimodaira, 2000). Under target shift, on the other hand, the weighting term is $\frac{P_t(y)}{P_s(y)}$. See Patel et al. (2015) for a more thorough explanation.

Daumé III (2009) proposes a feature augmentation framework in which features from both source and target domains are mapped into a feature space triple size of the original, which captures the feature similarities and particularities between source and target domains. In Daumé III's approach, given an input vector $x \in \mathbb{R}^n$, two mapping functions $\Phi_s(x) = [x,\, x,\, \mathbf{0}]$ and $\Phi_t = [x,\, \mathbf{0},\, x]$ are defined for the source and target datasets, respectively. Here $\mathbf{0} = [0,\, 0,\, ...,\, 0] \in \mathbb{R}^n$ is the zero vector. In this way the enhanced feature space contains a general version of the data (the first third of the vector where data from both domains appear) and a version of the data specific to one of the domains (the second third of the vector for the source domain and the last third of the vector for the target domain). The classifier is then expected to learn the adaptation by, for example, assigning weights to each version of the data depending on how well it generalizes between the domains.

Other approaches are based on the idea of learning new feature representations that are domain invariant. Gopalan et al. (2011) developed a method motivated by incremental learning in which the adaptation is performed by gradually transitioning from one domain to the other. This is done by treating each domain as a point in the Grassmann manifold and sampling points along the geodesic path between them to obtain a description of the underlying domain shift. Gong et al. (2012) go further and integrate over an infinite number of subspaces using a kernel-based method.

Chan and Ng (2005) use expectation-maximization (EM) to estimate the class densities under the TarS setting by assuming the that the distribution of the features given the labels stay constant and applying the iterative procedure of the EM algorithm.

Kulis, Saenko, and Darrell (2011) introduce a method for finding non-linear transformations between domains by learning in the kernel space.

Our method is similar to the *location-scale generalized target shift* (LS-GeTarS) transformation proposed by K. Zhang et al. (2013). In LS-GeTarS, a transformation from $P_s(X|Y)$ to $P_t(X|Y)$ is modeled as a translation and a scaling of the data given by $x^{\text{new}} = x \cdot W + B$, where W is the scaling matrix and B the translation. Instead of working directly with the marginal and conditional distributions they use their kernel mean embedding. A kernel mean embedding is a representation of a probability distribution as a point in a Reproducing Kernel Hilbert Space. In this manner, they do not need to assume a certain distribution, but minimize the loss using the kernel embedding and the algebraic operations it supports. The importance weight $\frac{P_t(Y)}{P_s(Y)}$ is estimated along the transformation parameters.

There are some other proposals that also use Bayesian transfer learning. Gönen and Margolin (2014) present a multi-task learning framework in which they apply kernel-based dimensionality reduction and use task-specific projection matrices to jointly find a common subspace. They define a different transformation of the data for each task, each of which is modeled as a projection matrix. The classifier is also part of the probabilistic

model, and they do inference on the transformations and the classifier at the same time. Our work differs in two substantial ways: (1) we find a transformation from the feature space of one "task" (in our case of one survey) to the space of another one, while this method creates a new common space that is different from the original space of all the tasks; and (2) the specific classification task is not part of our model, only the transformation. This means that our method can be used with any classifier and that if we change the classifier we do not need to do inference again to find the transformation.

Another Bayesian method is proposed by Finkel and Manning (2009), who present a hierarchical Bayesian framework for multiple domain adaptation. For each domain, there is an arbitrary probabilistic model for which a normally distributed prior is put on its parameters. In the next level of the hierarchy, another normally distributed prior is added to the domain specific parameter priors. This hierarchy can be extended further for an arbitrary number of levels, reflecting related super-domains, super-super-domains, and so on.

4. BACKGROUND THEORY

4.1. Time Domain Astronomy

Digital synoptic sky surveys enabled the emergence of the time domain research field in astronomy. Automated telescopes allow surveying large areas of the sky repeatedly in short scales of time. While prior star catalogs recorded a single apparent magnitude and position for a set of stars, current surveys provide multiple temporal measurements of each star over up to several years and even decades. The study of the time dimension in astronomy allows for the characterization of both existing and undiscovered phenomena. For example, type Ia supernovae, the result of the sudden thermonuclear explosion of a white dwarf in a gigantic emission of visible energy, can be used to map out the geometry of the universe (Committee for a Decadal Survey of Astronomy and Astrophysics, 2011). According to the Committee for a Decadal Survey of Astronomy and Astrophysics of the National Academy of Sciences in the United States (2011):

> "By surveying large areas of the sky repeatedly, once every few days, we anticipate the discovery of the wholly unanticipated. Endpoints of stellar evolution we have yet to imagine, and the behavior of ordinary stars outside our experience, could be discoveries that cause us to dramatically revise our cosmic understanding. Exotic objects and events never before anticipated may be revealed. The full realization of time-domain studies is one of the most promising discovery areas of the decade" (p. 45).

4.1.1. Light Curves

Timed brightness measurements of a celestial object can be tabulated and represented as a light curve, a plot of apparent magnitude versus time. Light curves are the fundamental analysis tool in time domain astronomy. Figure 4.1 shows four examples of light curves from the MACHO survey (Alcock et al., 1997).

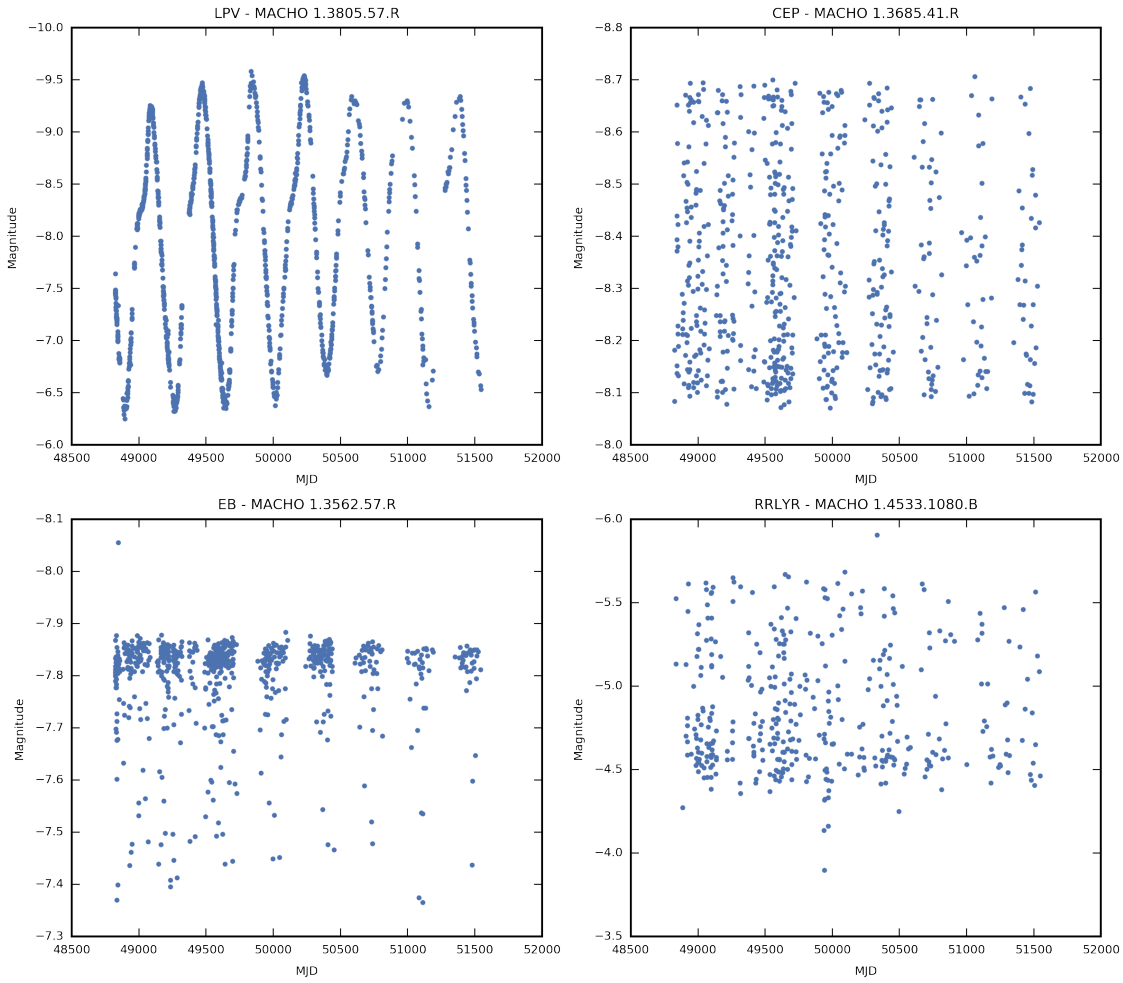


Figure 4.1. Example light curves for stars of four different variable classes.

4.1.2. Variable Stars

The apparent magnitude of every star changes along their typical life cycle: from the ignition of the collapsing nebula of gas and dust that marks the star's birth, through its growth into a red (super) giant, to its end as a planetary nebula and the formation of a white dwarf, or its spectacular death as a supernova and the formation of a neutron star or a black hole. However, a star is called *variable* when its brightness varies significantly during a human-perceptible time scale, ranging from minutes to centuries. This variation can be periodic, semi-periodic or irregular (Sterken & Jaschek, 2005).

Table 4.1. Descriptions of general types of variable stars. Adapted from Sterken and Jaschek (2005) citation of Kholopov et al. (1985).

Family	Type	Description
Intrinsic	**Eruptive**	Stars varying their brightness because of violent processes and flares taking place in their chromospheres and coronae.
	Pulsating	Stars showing periodic expansion and contraction of their surface layers. Pulsation may be radial or non-radial.
	Cataclysmic	Explosive variables showing outbursts caused by thermonuclear processes in their surface layers (Novae) or deep in their interiors (Supernovae).
Extrinsic	**Rotating**	Stars with non-uniform surface brightness or ellipsoidal shape. Their variability is caused by their axial rotation with respect to an observer. The non uniform surface brightness distribution may be caused by the presence of spots or by some thermal or chemical inhomogeneity of stellar atmospheres caused by magnetic fields.
	Eclipsing binary (EB)	Systems of two stars with their orbital plane aligned to Earth. Periodic drops in brightness are observed when one of the stars moves in front of the other, blocking its light.

The cause of brightness variability in stars can be roughly identified with one of two families depending on the nature of the underlying variability process. *Intrinsic* variable stars due their variability to some internal astrophysical process that may involve thermodynamic, gravitational, electrochemical, and other elements. For example, pulsating variables suffer periodic expansions and contractions of their outer layers, which momentarily change their size, temperature and brightness. On the other hand, *extrinsic* variable stars' brightness is perceived as changing to an observer due to external phenomena, such as stellar eclipses and stellar rotation. Table 4.1 describes some general star types belonging to these two families.

Depending on the time-scale, the magnitude of the amplitude variation and the shape of the light curve, variable star types are further classified into a series of classes (Sterken & Jaschek, 2005). Table 4.2 presents a description for some of them. The most important classes in the family, type and class topology are shown in Figure 4.2.

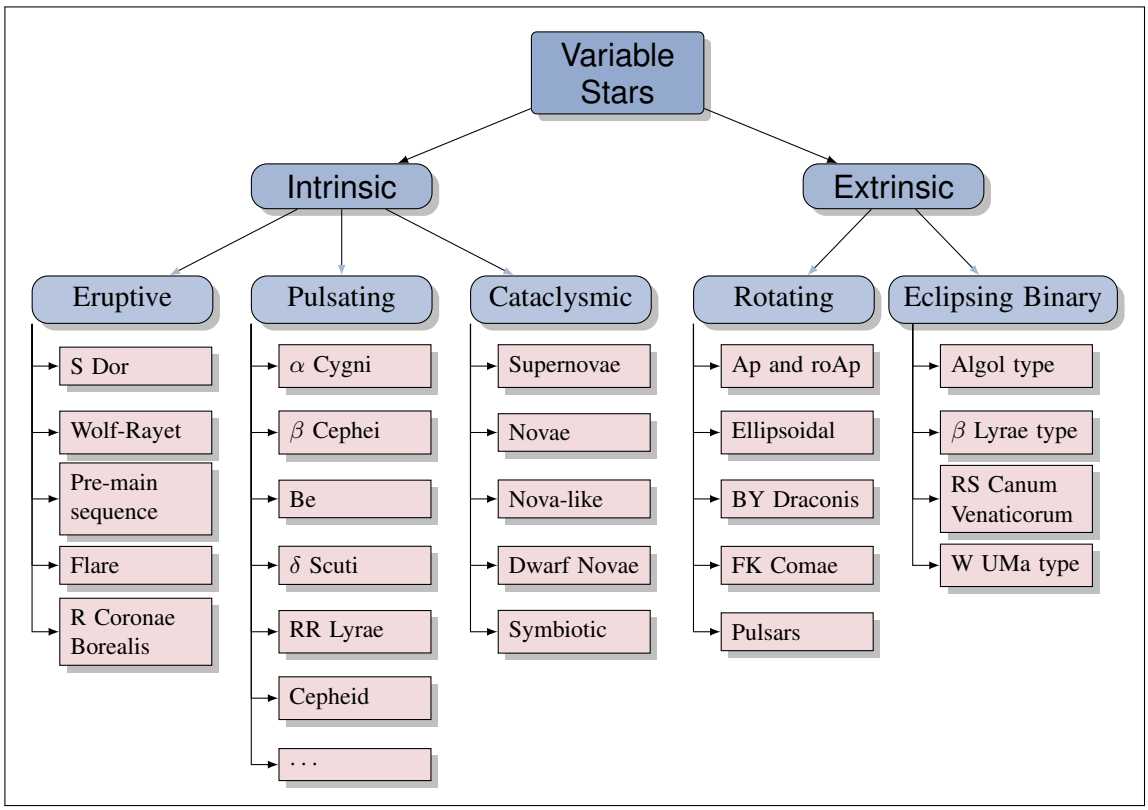


Figure 4.2. Classification of variable stars according to Sterken and Jaschek (2005).

Table 4.2. Descriptions of some variable star classes. Adapted from Sterken and Jaschek (2005).

Type	Class name	Description
Eruptive	**Flare**	Dwarf stars that undergo sudden brightening at irregular time intervals due to the violent ejection of material. The increase of their brightness can be more than 6 magnitudes. The time intervals between consecutive flares is usually between several hours and several days.
Pulsating	**RR Lyrae (RRLYR)**	Radial pulsators with periods from 0.2 to 1 days. Shock waves propagate outwards through their atmospheres once per pulsation cycle.
	Cepheid (CEP)	Strictly periodic variables with periods ranging from 1 day to 50 days with a few extreme cases of up to 200 days. They expand and contract periodically. Their brightness is very high and they can be detected in very distant galaxies. For this reason they are the basis of the extra-galactic distance scale.
	Long Period Variable (LPV)	The most studied pulsating red super giant stars. They are three orders of magnitude brighter than our sun and have periods from 80 to 1000 days. Their light curve shows maximums that often vary by a magnitude or more between cycles.
Rotating	**Pulsars**	Rapidly rotating neutron stars that emit regular pulses with periods between 1.558 ms and 4.308 s.
Cataclysmic	**Supernovae**	A rare type of stellar explosion in which large amounts of matter (several times the mass of our sun) are expelled at several thousands kilometers per second.

4.2. Machine Learning Background

Machine learning is the field of computer science concerned with developing models able to learn tasks from data without being explicitly programmed to perform said tasks (Samuel, 1959). It can also be defined as a set of methods of data analysis that can detect patterns in data automatically, and then use the discovered patterns in order to predict future data or to make decisions under uncertainty (Murphy, 2012).

Machine learning techniques are traditionally divided into two main types defined by the general goal of the task one wants to perform. *Supervised learning* aims to learn a mapping between input data X, known as *features* or *covariates*, to output data Y, known as *response variables* or *labels*. On the other hand, *unsupervised learning* aims to learn structure or interesting patterns in data, without using the notion of labels.

4.2.1. Supervised Learning

The most common tasks in supervised learning are *classification* and *regression*. In classification, the output or response variables are categorical. Each possible value of the output variable Y is known as a category or *label*. In regression, each output variable Y is continuous. Examples of classification tasks are hand-written digit recognition (Liu, Nakashima, Sako, & Fujisawa, 2003) and variable star classification (Pichara & Protopapas, 2013). Some examples of regression tasks are predicting stock market prices (Kim, 2003) and photometric redshift estimation (Benitez, 2000).

All forms of supervised learning require a set of data containing input-output pairs (i.e. feature-label or feature-value pairs) $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$. Such a dataset is known as a *labeled dataset* and each (x_i, y_i) tuple as a *labeled instance*.

Three commonly used supervised learning models, decision trees, random forests and support vector machines, are explained in the following subsections.

4.2.1.1. Decision Trees

A *decision tree* recursively partitions the feature space based on the value of a feature. Each region of the partition is defined as a node in a tree, and each partitioning condition as an edge. The simplest form of decision tree is a binary decision tree, which partitions the space in two in each recursion step. The label or regression value of an instance can be predicted by traversing the tree from its root, following the edges that satisfy the instance's feature values. At the end of the tree, each leaf contains the predicted label or value. Figure 4.3 shows a representation of a binary decision tree as a graph and the corresponding feature space partition.

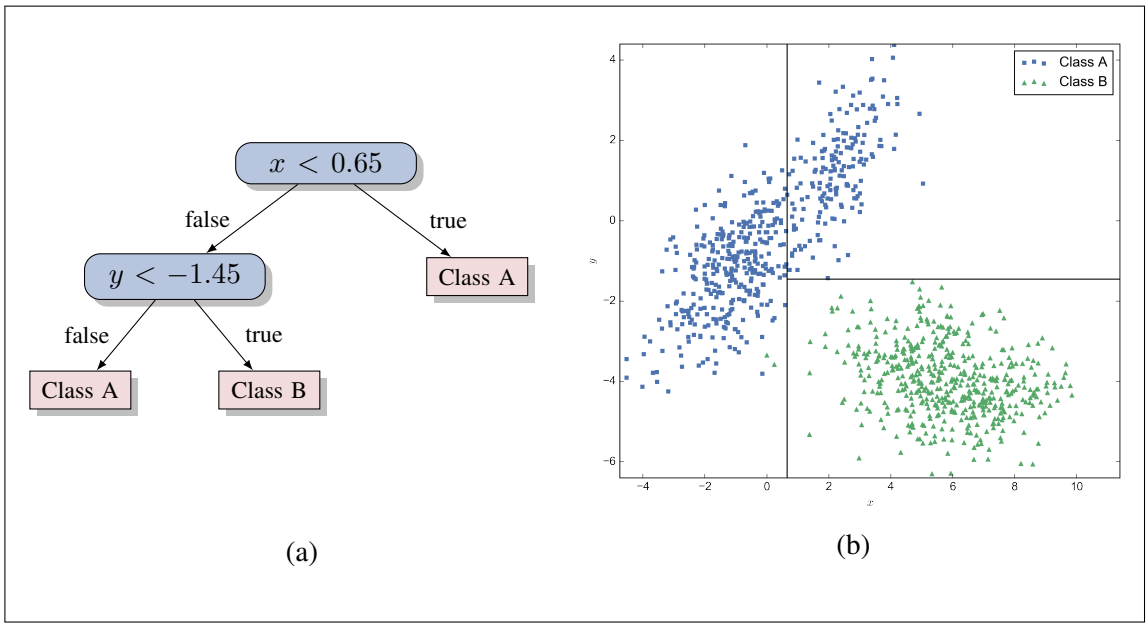


Figure 4.3. A binary decision tree. Example binary decision tree for classification in a 2D feature space with 2 labels. (a) Graph representation. (b) Equivalent space partition representation.

Decision trees are learned recursively by greedily choosing the partition that maximizes some criterion of "goodness", such as the information gain (the change of information entropy) from the previous state of the tree to the state resulting from the added partition.

In a binary decision tree, the learning algorithm starts with the whole dataset. The partitioning condition in the first node is chosen such that the resulting two partitions best separate the two classes (i.e. each partition contains mostly objects from a single class, which is the opposite of that of the other partition). Then, this step is repeated recursively with each generated partition. The process stops at any given node when the partitions only contain objects from one class, or when the objects from the "wrong" class are below some threshold.

4.2.1.2. Random Forests

Random Forests (Breiman, 2001) is a technique that employs multiple decision trees, each trained on a random subset of the data. To make a prediction, each tree casts a "vote", which is aggregated with that of the rest of the trees (for example, by taking the mode of the classification, or the mean in the case of regression). This general technique is known as ensemble learning (Dietterich et al., 2000). To reduce correlation among the trained trees, each of them is trained using a random subset of input features, as well as a random subset of input data.

Random Forests have been used with success in astronomical datasets (Pichara, Protopapas, Kim, Marquette, & Tisserand, 2012; Carliles, Budavári, Heinis, Priebe, & Szalay, 2010; Dubath et al., 2011; Johnston & Oluseyi, 2017).

4.2.1.3. Support Vector Machines

Support Vector Machines (SVM) (Boser, Guyon, & Vapnik, 1992; Vapnik, 1963) is a supervised learning framework that builds hyperplanes with the most possible distance between data points of different classes – maximum margin separators – and uses them as decision boundaries to classify instances. The hyperplanes learned by SVMs are not limited to linearity. By applying the "kernel trick", data can be embedded into a higher-dimensional space using a kernel function. The linear hyperplane learned in the higher dimension is then non-linear in the original space (Russell & Norvig, 2009).

Kernel functions can be understood as a similarity measure between a pair of points. An ideal kernel function assigns a high similarity score to two points that belong to the same class, and a lower score to points belonging to different classes (Hofmann, 2006). One of the most used kernels is the radial basis function (RBF) kernel, which for a pair of data points x, x' is defined as:

$$\kappa(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \tag{4.1}$$

With σ a free parameter.

4.2.2. Unsupervised Learning

Unlike supervised learning, unsupervised learning does not need a labeled dataset. Instead, an *unlabeled dataset* $\mathcal{D} = \{x_i\}_{i=1}^N$ is available. Each x_i is known as an *unlabeled instance*. Since there is no explicit output to be found, the goal is to find inherent structure and patterns in the data.

The most common application of unsupervised learning is *clustering*. Clustering consists in dividing a dataset into a discrete number of subsets called *clusters*, such that each instance is associated with one of them. One of the most popular clustering techniques is the K-means algorithm.

4.2.2.1. K-means

The K-means algorithm (MacQueen et al., 1967) is a method for partitioning a set of points into a given number of clusters. Each cluster is defined in terms of a centroid and each point is assigned to the cluster of the closest centroid. Note that the clustering is completely defined in terms of the centroids. The learning algorithm initializes the centroids of a given number K of clusters at random and then proceeds iteratively through two stages. In the first stage, each data instance is mapped to the cluster with the closest

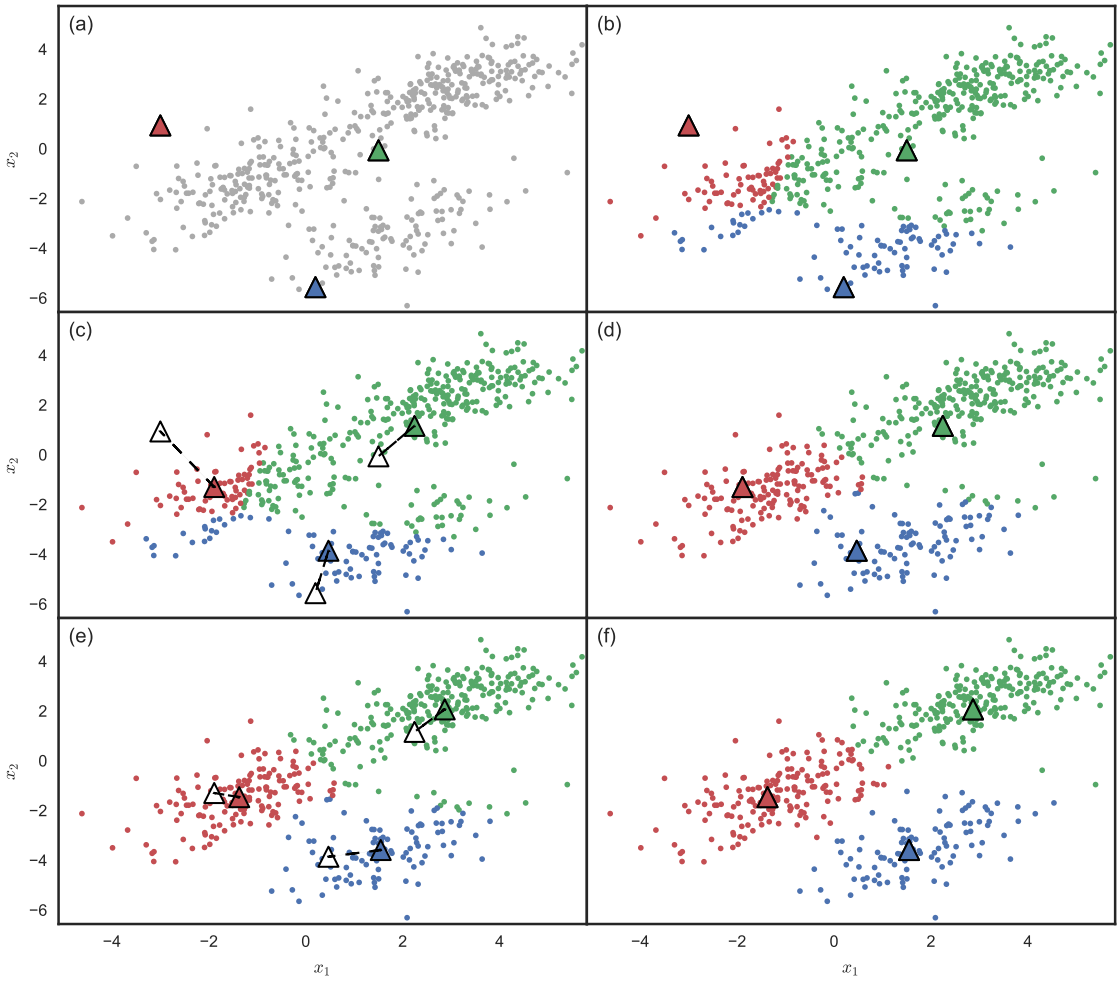


Figure 4.4. K-means example in 2D space. (a) Cluster centroids are initialized at random positions. (b) First step: each data point is mapped to the closest centroid. (c) Second step: the centroids are moved to the mean of each cluster's mapped points. (d, e, f) The steps are repeated iteratively until convergence.

centroid. Then, in the second stage, each cluster centroid is re-calculated as the mean of all the points mapped to them. These steps are repeated until convergence (i.e. when no or very small variations in the centroid coordinates exist between two iterations). Figure 4.4 shows two iterations of an example execution of K-means.

4.2.3. Probabilistic Graphical Models

Probabilistic graphical models are a framework for conveniently representing and manipulating joint probability distributions that draws from probability theory and graph theory. By using a graph data structure, graphical models leverage representations and algorithms from computer science to allow for representation, learning and inference of otherwise unmanageable probability distributions.

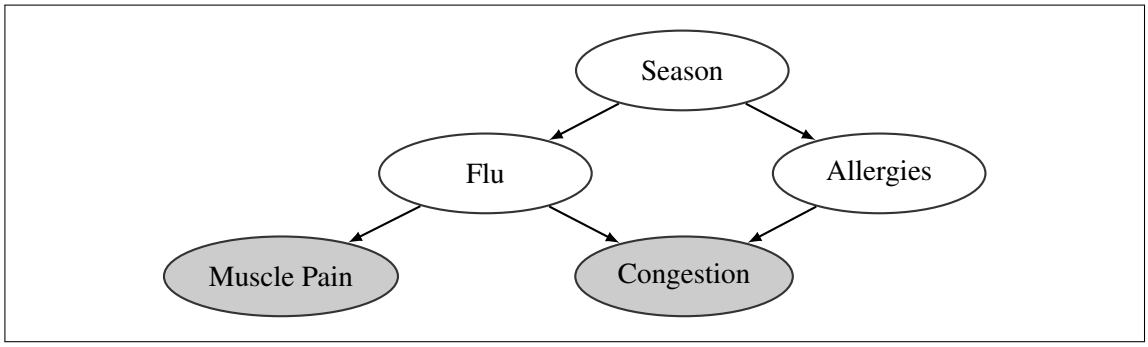


Figure 4.5. Probabilistic graphical model example. Graphical representation of an hypothetical model for disease diagnosis.

In a graphical model, random variables are represented as nodes in a graph, while dependence relationships are encoded as graph edges. In this work, we are interested in a form of directed graphical models known as *Bayesian networks* (Koller & Friedman, 2009), which encode a set of conditional independences in a joint probability distribution. In a Bayesian network, a directed edge from variable X to variable Y indicates that variable Y directly depends on variable X. The network satisfies the *local independence assumption*, which holds that every node $X_i \in \{X_1, ..., X_n\}$ of an n node graph is conditionally independent of its non-descendants given its parents (Koller & Friedman, 2009):

$$(X_i \perp \text{NonDescendants}_{X_i} \mid \text{Parents}_{X_i}) \quad \forall i \in 1, \ldots, n \tag{4.2}$$

Consider the following hypothetical example for medical diagnosis from Koller and Friedman (2009). Imagine we want to model the probability of having two diseases – flu

(F) and allergies (A) – given the season of the year (S) and two symptoms – muscle pain (M) and nasal congestion (C). The graphical model for this example is shown in figure 4.5. These are some of the conditional independencies encoded by the model:

$$(\text{Flu} \perp \text{Allergies} \mid \text{Season})$$

$$(\text{Muscle Pain} \perp \text{Congestion}, \text{Allergies} \mid \text{Flu})$$

$$(\text{Congestion} \perp \text{Muscle Pain} \mid \text{Flu})$$

If we are interested in knowing the probability of muscle pain, and we know that the patient has the flu, then knowing if they have congestion is no longer informative. This does not mean that muscle pain is independent of congestion: if we do not know if the patient has the flu, then knowing they suffer congestion changes our belief on whether they have this disease or not. This ultimately affects the probability they also suffer muscle pain.

4.2.3.1. Plate Notation

In this work, we represent graphical models using the *plate notation* (Koller & Friedman, 2009). Identically distributed variables that are repeated many times are enclosed by a rectangle or plate, capturing the notion that they correspond to a "stack" of identical variables. Variables in a plate are indexed and repeated according to the indication on the lower right corner of the plate. Edges connecting two nodes in the same plate connect variables with the same index. Edges connecting nodes inside a plate with nodes outside of it connect the outer variable with all the instances of the repeated variable. Edges from one plate to a different plate connect all instances in one plate with all the instances in the second plate. In this way, models can be represented in a more compact way by plotting each variable instance once. Consider a model for medical diagnosis with P patients in which the presence of each of N diseases $D_{k,1}, \ldots, D_{k,N}$ depends on the manifestation on

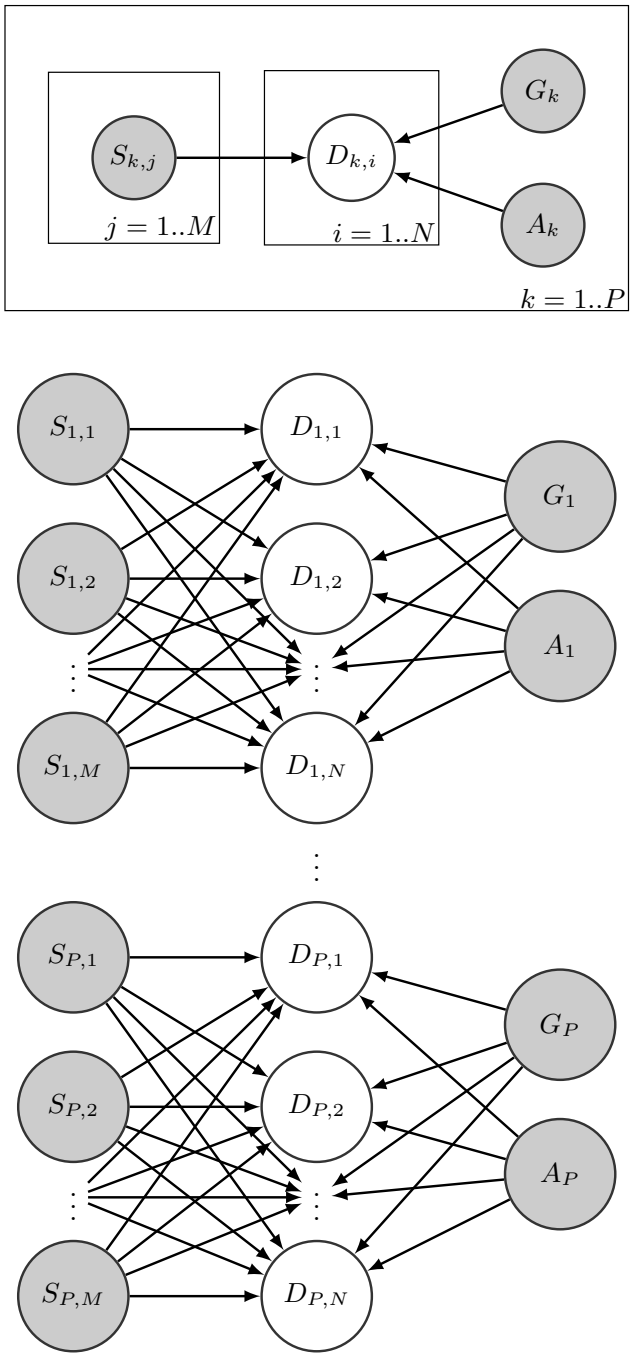


Figure 4.6. Example of a graphical model in plate notation. The top panel shows the model in plate notation. Indexed variables are displayed once. The bottom panel shows the same model using no plates. Each variable is explicitly displayed.

patient k of M symptoms $S_{k,1}, \ldots, S_{k,M}$ and the gender G_k and age A_k of the patient. Figure 4.6 shows a graphical representation for this model, both using plates and by explicitly displaying all variables.

4.2.3.2. Mixture Models

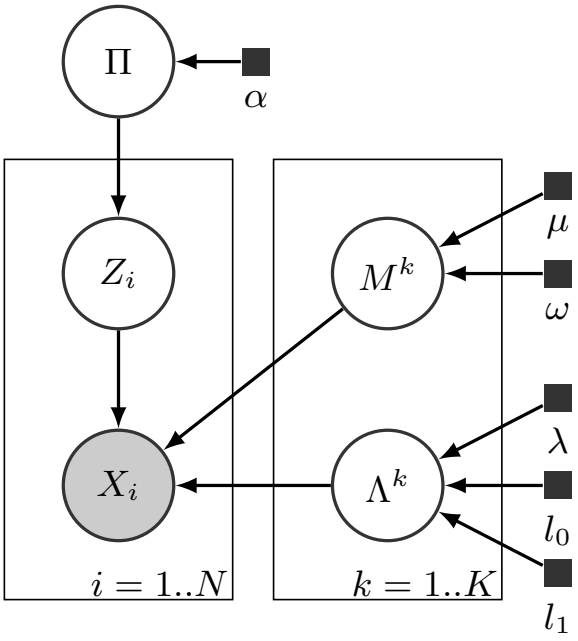


Figure 4.7. Gaussian mixture model in plate notation. Each data point X is generated by the component indicated by Z. Each of the K components has mean vector and precision matrix priors M^k and Λ^k, respectively. Black squares indicate prior hyperparameters.

A mixture model is a convex combination (i.e. relative weights sum to one) of probability distributions (Bishop, 2006). Suppose we have K Gaussian distributions generating a set of data. We call each distribution a component. Let Z be a binary vector indicating which component generated a certain value of X, and give it a representation where Z^k is equal to 1 if X was generated by component k and 0 otherwise (e.g. $Z = [0\ 1\ 0\ 0]$ when the value is generated by component 2). Let Π^k be the mixture coefficients (i.e. the relative weight or importance of each component in generating the data). Then we can write the marginal distribution of Z:

$$P(Z) = \prod_{k=1}^{K} (\Pi^k)^{Z^k} \tag{4.3}$$

And the conditional distribution of X given it was generated by component k as:

$$P(X|Z) = \prod_{k=1}^{K} \mathcal{N}(X|M^k, \Sigma^k)^{Z^k} \tag{4.4}$$

Where $\mathcal{N}(X|M^k, \Lambda^k)$ is the probability density function for the multivariate Gaussian distribution with mean vector M^k and covariance matrix Σ^k. Unfortunately, using covariance matrices is inefficient, as it involves expensive matrix inversion operations when computing the likelihood. For this reason, we use the inverse covariance matrix, also known as precision matrix, $\Lambda^k = (\Sigma^k)^{-1}$:

$$P(X|Z) = \prod_{k=1}^{K} \mathcal{N}(X|M^k, \Lambda^k)^{Z^k} \tag{4.5}$$

We use equations (4.3) and (4.5) to obtain the mixture's density by marginalizing over Z:

$$\begin{aligned}
P(X) &= \sum_{Z} P(X|Z)P(Z) \\
&= \sum_{Z} \prod_{k=1}^{K} \mathcal{N}(X|M^k, \Sigma^k)^{Z^k} \prod_{k=1}^{K} (\Pi^k)^{Z^k} \\
&= \sum_{Z} \prod_{k=1}^{K} \left(\Pi^k \, \mathcal{N}(X|M^k, \Sigma^k)\right)^{Z^k} \\
&= \Pi^1 \, \mathcal{N}(X|M^1, \Sigma^1) + ... + \Pi^K \, \mathcal{N}(X|M^K, \Sigma^K) \\
&= \sum_{k=1}^{K} \Pi^k \mathcal{N}(X|M^k, \Lambda^k)
\end{aligned} \tag{4.6}$$

Thus, we can interpret each Π^k as the prior probability of assigning a sample X to component k (Bishop, 2006).

We can derive the corresponding posterior probability Γ using Bayes' theorem:

$$\begin{aligned} \Gamma^k(X) &= P(Z^k = 1|X) \\ &= \frac{\Pi^k \mathcal{N}(X|M^k, \Lambda^k)}{\sum_{l=1}^{K} \Pi^l \mathcal{N}(X|M^l, \Lambda^l)} \end{aligned} \tag{4.7}$$

Γ^k is referred to as the responsibility of component k and it represents how strongly component k contributed to generating sample X (Bishop, 2006).

Figure 4.7 shows a Gaussian Mixture model in plate notation.

4.2.3.3. Precision Matrix Modeling

As we are developing a Bayesian model, we assign priors to the mixture component's parameters. For the components' means a Gaussian prior is commonly used. As for the precisions, the Wishart distribution is a popular choice due to its conjugacy to the multivariate Gaussian distribution when a dependency to the mean is introduced. However, as Barnard, McCulloch, and Meng (2000) point out, when specifying a prior it is more convenient to work in terms of standard deviation and correlation. For this purpose, they suggest a separation strategy, decomposing a covariance matrix Σ into a standard deviation vector σ and a correlation matrix C:

$$\Sigma = \text{diag}(\sigma)\, C\, \text{diag}(\sigma) \tag{4.8}$$

Here $\text{diag}(v)$ represents the square diagonal matrix whose main diagonal contains the elements in vector v. This provides the advantage that we can express our prior knowledge

of the standard deviation and correlation separately on the original scale of the standard deviation (Barnard et al., 2000).

A random value for σ can be generated using any continuous distribution. We generate the correlation matrix C using the method proposed by Lewandowski, Kurowicka, and Joe (2009), from which we can get a random correlation matrix C of any given dimension with density proportional to $|C|^{\lambda-1}$ for a shape parameter $\lambda > 1$. We will refer to this distribution over the space of correlation matrices as LKJ. In section 5.1 we explain how we use this principle to model the precision matrix priors for the mixture components.

4.2.4. N-Dimensional Rotations

Rotations in 2D and 3D space are commonly understood as rotations about an axis by a certain angle. Duffin and Barrett (1994) argue that it is better to think about them as occurring in a plane: the plane perpendicular to the axis of rotation in 3D, and the only plane in 2D. They generalize the concept to rotation in n-dimensional space in principal planes formed by two coordinate axes.

The rotation matrix for the rotation of axis X_a in the direction of axis X_b by an angle of θ is as follows (Duffin & Barrett, 1994):

$$R_{ab}(\theta) = \left\{ r_{ij} \;\middle|\; \begin{array}{ll} r_{ii} = 1 & i \neq a, i \neq b \\ r_{aa} = \cos\theta & \\ r_{bb} = \cos\theta & \\ r_{ab} = -\sin\theta & \\ r_{ba} = \sin\theta & \\ r_{ij} = 0 & \text{elsewhere} \end{array} \right\} \tag{4.9}$$

An arbitrary rotation in n-dimensional space can then be specified as the composition of rotations in the $\binom{d}{2}$ principal planes.

5. METHOD DESCRIPTION

We propose a probabilistic model based on the Gaussian Mixture Model (GMM) to describe the $P_s(X, Y)$ and $P_t(X, Y)$ distributions. We model mixture weights, component mean vectors and precision matrices for the source distribution in the usual manner. However, each of the target distribution mixture parameters is modeled as a separate transformation of the respective parameter of the source mixture: each target mean vector is equal to the respective source mean vector plus a translation and each target precision matrix is equal to a scaling and a rotation of the respective source precision matrix. Note that each component needs not undergo the same transformation as the others, since there are separate transformation variables for each one. Here, we are making the assumption that we can capture the domain shift between the datasets as a mixture of transformations in the subspaces defined by each multivariate Gaussian. That is, we propose a model that describes a mixture of Gaussians over the source dataset and a linear transformation for

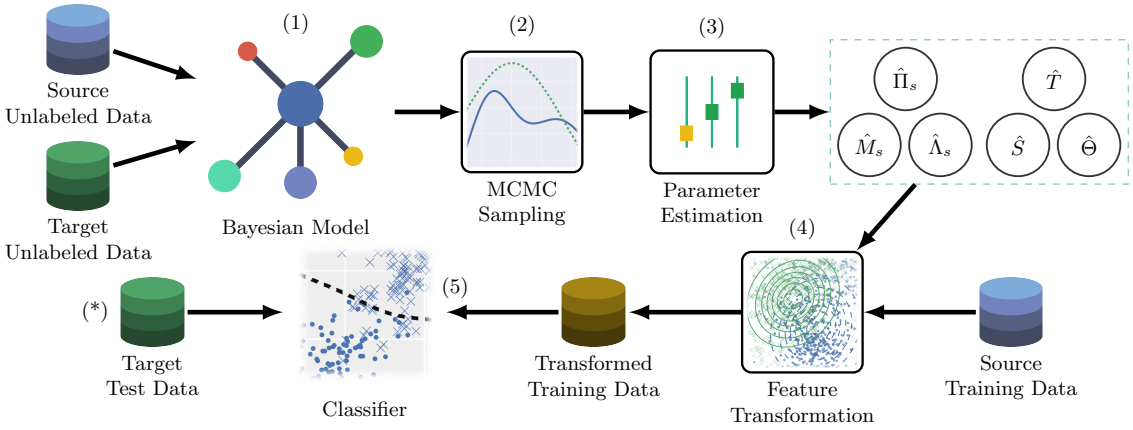


Figure 5.1. Domain adaptation method overview. (1) The probabilistic model is constructed according to the specification and the unlabeled input data. (2) MCMC techniques are used to sample from the posterior distributions of the transformation parameters. (3) The transformation parameters are estimated by averaging the samples. (4) The transformation is applied to the source training data according to the estimated parameters. (5) A classifier is trained on the transformed input data from the source domain. (*) In our experiments, the classifier is tested on labeled data from the target domain to assess performance.

each of its components, which result in a transformed mixture of Gaussians over the target dataset. Note that since the target mixture is fully determined by the source mixture and the transformation, the mapping between the corresponding components is given implicitly – the target component that corresponds to a source component is simply the one resulting from applying the component's transformation. Furthermore, we assume that the training set suffers the same shift between the domains than the unlabeled dataset and that by inferring it from the latter we will be able to correct it for the former as well. The latter implies that we assume that there is no significant unrepresented population in the unlabeled data.

Our method comprises 5 steps, as shown in Figure 5.1: (1) the model is set-up using the unlabeled data from the source and target domains (the detailed model specification follows in Section 5.1). An optional step of randomly sampling from the datasets may be performed here, depending on the amount of data and computational resources available. (2) The mixture and transformation parameters variables are sampled using the Metropolis Hastings MCMC method. (3) The samples are used to make a point estimate of the parameters. Steps 2 and 3 are explained in Section 5.2. (4) The estimated parameters are used to apply the modeled transformation to the training set that is available for the source domain, in order to correct the shift. The transformation is explained in detail in Section 5.3. (5) Using the transformed training data, a classifier expected to perform well on the target domain is trained.

In our experiments, presented in Chapter 6, we perform an additional step of testing on a target domain labeled dataset in order to assess the method's performance. This dataset is not used at any moment in the previous steps.

5.1. Model Specification

First, we specify the mixtures that represent the source and target datasets. Let X_s^i and X_t^j be random variables for the i-eth and j-eth unlabeled sample in the source and target

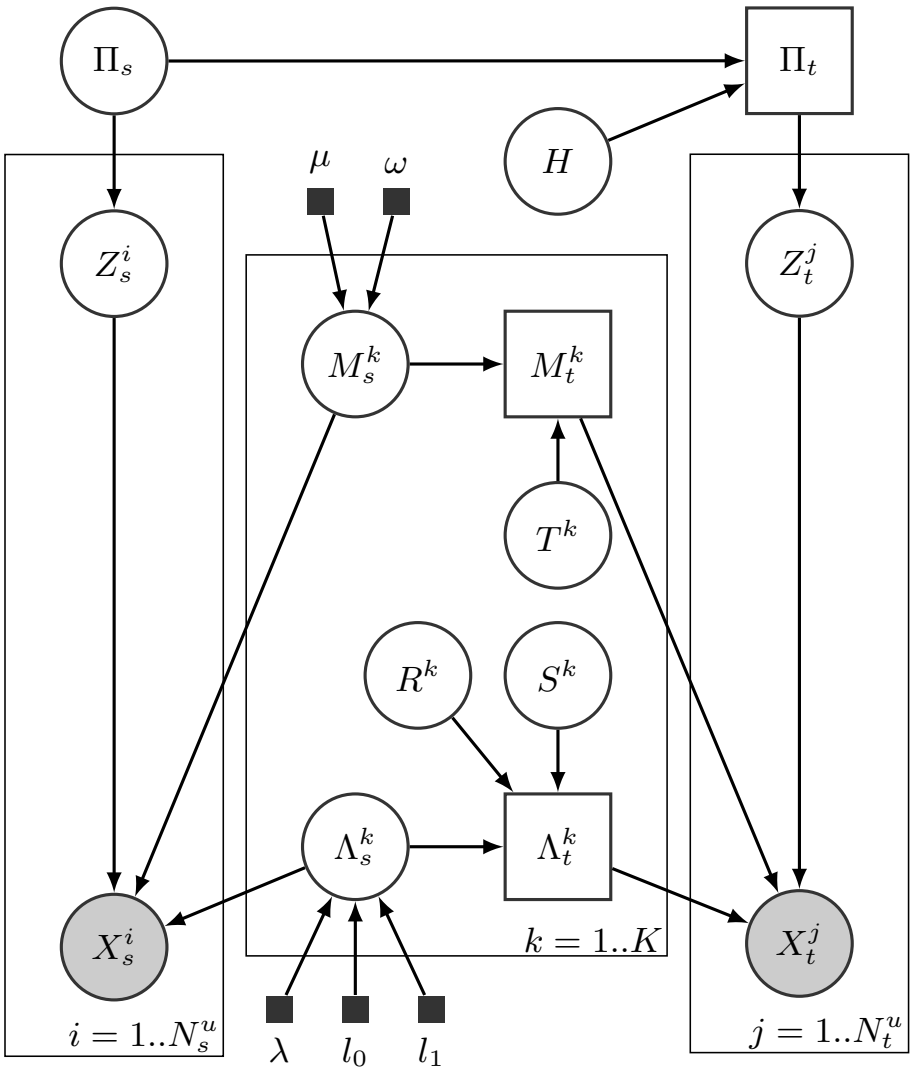


Figure 5.2. Proposed model in plate notation. Random variables are shown in circles. Variables derived deterministically from other variables are shown in squares. Prior hyperparameters are shown as small black squares. Observed variables are shaded in gray. Some hyperparameters are omitted for clarity.

datasets, respectively, and let d be the dimensionality of the data. Let Z_s^i and Z_t^j denote the component assignments for source sample i and target sample j, M_s^k and M_t^k the mean for component $k \in 1..K$ and Λ_s^k and Λ_t^k the precision for component k in the source and target domains, respectively. Let Π_s and Π_t be the priors for the component weights.

The following distributional assumptions are made:

$$
\begin{aligned}
&\Pi_s \sim \mathcal{D}(\alpha) && && H \sim \mathcal{D}(\eta) && \\
&Z_s^i \sim \mathcal{C}(\Pi_s) && \forall i && Z_t^j \sim \mathcal{C}(\Pi_t) && \forall j \\
&M_s^k \sim \mathcal{N}(\mu, \omega) && \forall k && && \\
&X_s^i \sim \mathcal{N}(M_s^{Z_s^i}, \Lambda_s^{Z_s^i}) && \forall i && X_t^j \sim \mathcal{N}(M_t^{Z_t^j}, \Lambda_t^{Z_t^j}) && \forall j
\end{aligned}
$$

Where $\mathcal{D}(\alpha)$ denotes the Dirichlet distribution with concentration parameter vector α and dimension K, $\mathcal{C}(\pi)$ represents the categorical distribution with event probability vector π of dimension K, and $\mathcal{N}(\mu, \Lambda)$ represents the normal distribution of dimension d with mean vector μ and precision matrix Λ.

The target component weights are determined by the source component weights and the Dirichlet distributed variable H, which scales each weight like so:

$$
\Pi_t^k = \Pi_s^k \, H^k / \sum_{l=1}^{K} \Pi_s^l \, H^l \quad \forall k \tag{5.1}
$$

Higher values for the hyperparameter η will favor small differences in component weight between domains.

As explained in section 4.2.4, the source domain components' precision matrices Λ_s^k are generated using the separation strategy defined in equation (4.8). Each resulting covariance matrix is then inverted to yield the corresponding precision matrix:

$$
\begin{aligned}
&L^k \sim \mathcal{LKJ}(\lambda) && \forall k && \Sigma^k \sim \mathcal{U}(l_0, l_1) && \forall k \\
&\Lambda_s^k = \left(\text{diag}(\Sigma^k) L^k \text{diag}(\Sigma^k)\right)^{-1} && \forall k && &&
\end{aligned}
$$

Where $\mathcal{U}(l_0, l1)$ denotes the uniform distribution of dimension d with minimum value l_0 and maximum value l_1, and $\mathcal{LKJ}(\lambda)$ denotes the LKJ distribution of dimension d with shape parameter λ.

Second, we introduce random variables for the component transformations. Each component suffers a translation of its mean vector, and a rotation and a scaling of its precision matrix. The translation of component's k mean, T^k, and the resulting target mean vectors M_t^k are as follows:

$$T^k \sim \mathcal{N}(0, \kappa I) \quad \forall k$$
$$M_t^k = M_s^k + T^k \quad \forall k$$

Where 0 represents the zero vector of dimension d and I represents the $d \times d$ identity matrix. The κ hyperparameter specifies the a priori belief about the magnitude of the translations, so that smaller κ values will favor larger translations.

Each precision matrix rotation is modeled as a $\binom{d}{2}$-dimensional vector of angles Θ^k, representing the rotation of each principal plane. The precision matrices scalings are modeled as factors multiplying each dimension centered at the component's mean:

$$\Phi^k \sim \mathcal{B}^{\binom{d}{2}}(\beta, \beta) \quad \forall k \qquad S^k \sim \mathcal{B}^d(\epsilon, \epsilon) + 0.5 \quad \forall k$$
$$\Theta^k = (2\Phi^k - 1)\,\rho \quad \forall k$$

The notation $\mathcal{B}^d(\alpha, \beta)$ corresponds to the beta distribution of dimension d with shape parameters α and β. We abuse the notation $\mathcal{B}^d(\alpha, \beta) + \delta$ to represent the same beta distribution with its range offset by δ, resulting in a support in the range $[\delta, 1.0 + \delta]$. We let the rotation in each principal plane be in the interval $[-\rho, \rho]$. In order to do so, we draw from each Φ^k prior $\binom{d}{2}$ values between 0 and 1 and use them to interpolate between the rotation limits and get Θ^k, the angles of rotation. The hyperparameter β represents the a priori belief about the magnitude of the rotations. Larger values of β mean a more tight distribution around 0.5, which equals to a null rotation. Then, for each of the $\binom{d}{2}$ main planes of rotation we use equation 4.9 to build a rotation matrix and compose them like so:

$$R^k = \overrightarrow{\prod_{a=1}^{d-1}} \overrightarrow{\prod_{b=a+1}^{d}} R_{ab}(\Theta_l^k) \quad \forall k \tag{5.2}$$

$$l = (a-1)(d - a/2) + b - a \tag{5.3}$$

Where $\rightarrow\prod$ denotes aggregated left side matrix multiplication so that equation (5.2) expands to:

$$\begin{aligned} R^k = & R_{(d-1)\,d}\left(\Theta^k_{\binom{d}{2}}\right) \ldots R_{2d}(\Theta^k_{2d-3}) \ldots \\ & \ldots R_{24}(\Theta^k_{d+1})\; R_{23}(\Theta^k_d)\; R_{1d}(\Theta^k_{d-1}) \ldots \\ & \ldots R_{13}(\Theta^k_2)\; R_{12}(\Theta^k_1) \qquad\qquad \forall k \end{aligned} \tag{5.4}$$

The scaling factors for each dimension are allowed in the range $[0.5, 1.5]$. The ϵ determines how tightly around a 1 scaling factor the distribution will be.

We get the precision matrix that would result from scaling the data in each component k by S^k and then rotating it according to R^k as:

$$\Lambda_t^k = R^k (S^k)^{-1} \Lambda_s^k (S^k)^{-1} (R^k)^{-1} \quad \forall k \tag{5.5}$$

5.2. Parameter Estimation

In order to apply the transformation, we first estimate the T^k, S^k and Θ^k transformation parameters, and the Π_s, Π_t, M_s^k, Λ_s^k model parameters by sampling from their posterior distributions using the Gibbs MCMC sampler. To accelerate convergence, we run K-means on the source dataset to find centroids for the source components and initialize the mean vectors to their values. We step through MCMC iterations until the standard

deviation of the samples is below a certain threshold. We then use the mean point estimate of the samples as the parameter values.

5.3. Feature Transformation

Let $\hat{\Pi}_s$ and $\hat{\Pi}_t$ be the estimates for the source and target component weights Π_s and Π_t, respectively. Let the $K \times d$ matrix $\hat{T}$ contain the estimate of the translation T^k of each component as rows, the $K \times d$ matrix $\hat{S}$ contain the estimates of the component scalings S^k as rows, and the $K \times \binom{d}{2}$ matrix $\hat{\Theta}$ contain the estimate for the rotation angles of the $\binom{d}{2}$ principal planes of each component as rows. Similarly, let $\hat{M}_s^k$ and $\hat{\Lambda}_s^k$ be the estimates for the mean vector and the precision matrix for each component k, respectively.

We then apply a transformation Ψ to the source domain training set D_s^l in order to obtain a labeled dataset $D_\star^l = \{(\Psi(x_i^{sl}), y_i^{sl})\}_{i=1}^{N_s^l}$ suitable for training a classifier for the target domain. Let X be a $N_s^l \times d$ matrix containing the source training samples, such that $X_i = x_i^{sl}$ for $i = 1, ..., N_s^l$.

We compute the $N_s^l \times K$ matrix W containing the component transformation weights for each instance given by equation 4.7:

$$W_{ik} = \Gamma^k(X_i) \quad \forall i, \forall k \tag{5.6}$$

The translation for each instance is given by the weighted average of the component responsibilities and the component translations:

$$\Delta = W\,\hat{T} \tag{5.7}$$

Where Δ is a $N_s^l \times d$ matrix containing the translation for each instance.

Scaling proportional to component responsibility is applied by computing the translation Ξ_i that results from scaling centered under each component:

$$\Xi_i = \sum_{k=1}^{K} W_{ik} \left(\text{diag}(\hat{S}^k) - I \right) (X_i - \hat{M}_s^k) \quad \forall i \tag{5.8}$$

Where I is the $d \times d$ identity matrix and $\text{diag}(v)$ is the square diagonal matrix whose main diagonal contains the elements in vector v.

Finally, we rotate the data with respect to each component center. First we compose the rotation matrices for each instance and component similarly as in equation 5.2, but using weighted transformation angles:

$$\hat{R}_i^k = \overrightarrow{\prod_{a=1}^{d-1}} \overrightarrow{\prod_{b=a+1}^{d}} R_{ab}(W_{ik}\, \hat{\Theta}_l^k) \quad \forall i, \forall k \tag{5.9}$$

With l as given by equation 5.3.

Then the transformation $\Psi(X_i) = X^\star$ is given by the following algorithm:

$X^\star = X_i + \Xi_i$
for $k = 1 \rightarrow K$ **do**
  $X^\star = \hat{R}_i^k (X^\star - \hat{M}_s^k) + \hat{M}_s^k$
end for
$X^\star = X^\star + \Delta_i$

Which applies the offset produced by the scalings, rotates the data according to each component and finally applies the weighted translation.

6. EXPERIMENTAL RESULTS AND ANALYSIS

6.1. Methodology

We transfer the training knowledge from the source to the target catalog by performing the steps illustrated in Figure 5.1:

(i) Build the Bayesian model with the source catalog and target catalog unlabeled datasets.
(ii) Perform MCMC sampling from the posterior distributions of the transformation parameters until convergence.
(iii) Estimate the transformation parameters using the mean of the samples.
(iv) Take the source catalog's training set and transform it using the parameters obtained in the previous step.
(v) Train a classifier using the adapted training set.

We then measure the performance of our method by testing the classifier on a labeled dataset from the target catalog left out for this purpose.

The classifiers used in our experiments are the Radial Basis Function (RBF) kernel Support Vector Machine (SVM) (Boser et al., 1992) and the Random Forest (RF) (Breiman, 2001) classifier.

6.2. Simulations

We generated datasets and simulated domain shifts of different nature in order to study the performance and behavior of our model under conditional shift and generalized target shift.

First, we simulated covariate and conditional shift. The left two panels in Figure 6.1 show a simulated dataset generated using two multivariate Gaussians. In the top panel, the

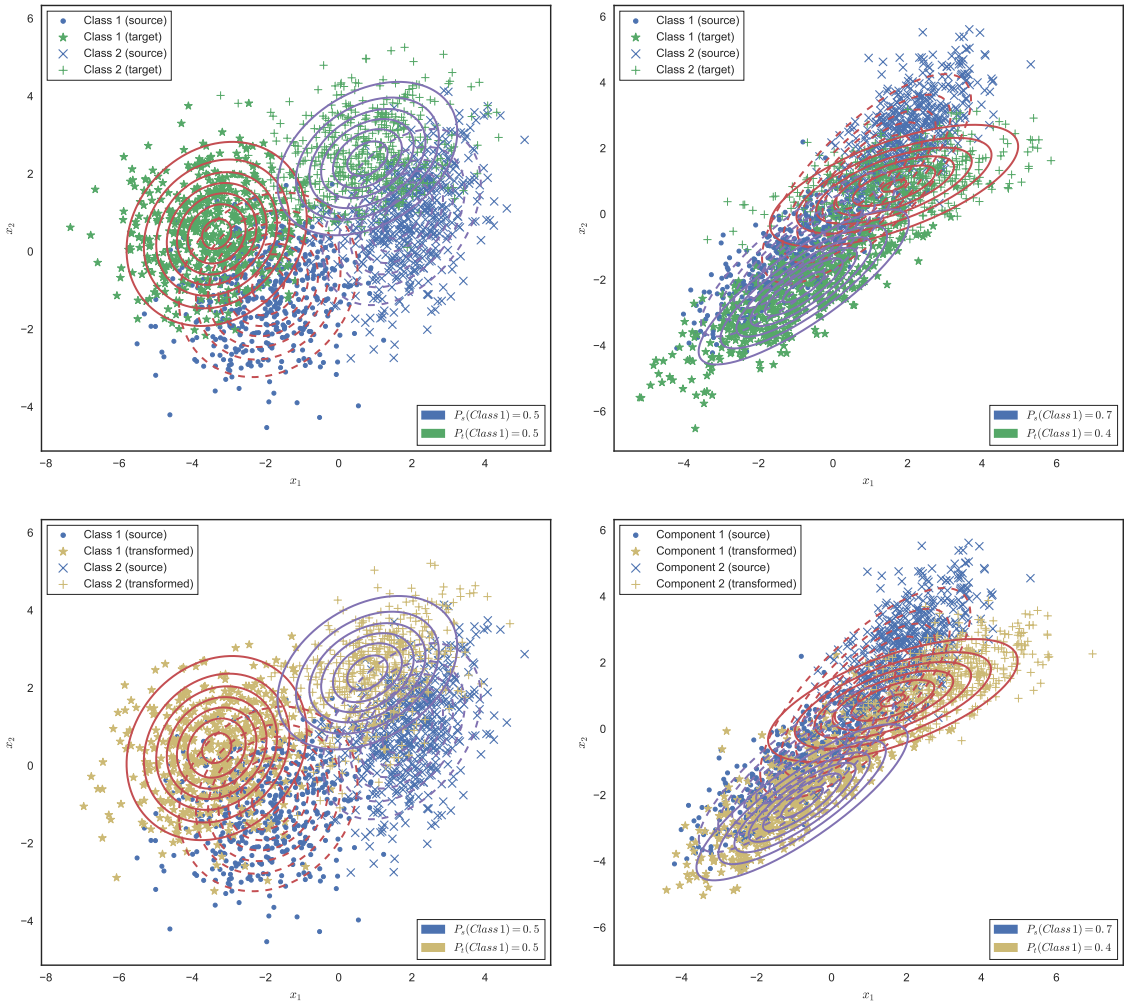


Figure 6.1. Model visualization on simulated data. Left two panels: simulation under ConS. Right two panels: simulation under GeTarS. Top panels show the fitted models. The dashed level curves represent the mixture components for the source dataset and the continuous lines show the components after applying the transformation found. the bottom panels show the transformation. The source dataset points are shown in blue along with their image after applying the transformation in yellow. The class distributions for each dataset are shown on the lower right corner.

target dataset, shown in green, was generated by translating, scaling and rotating the source dataset distribution, shown in blue. The components of the fitted model are represented as level curves. The bottom panel shows the transformed source dataset in yellow, along with the original source dataset in blue.

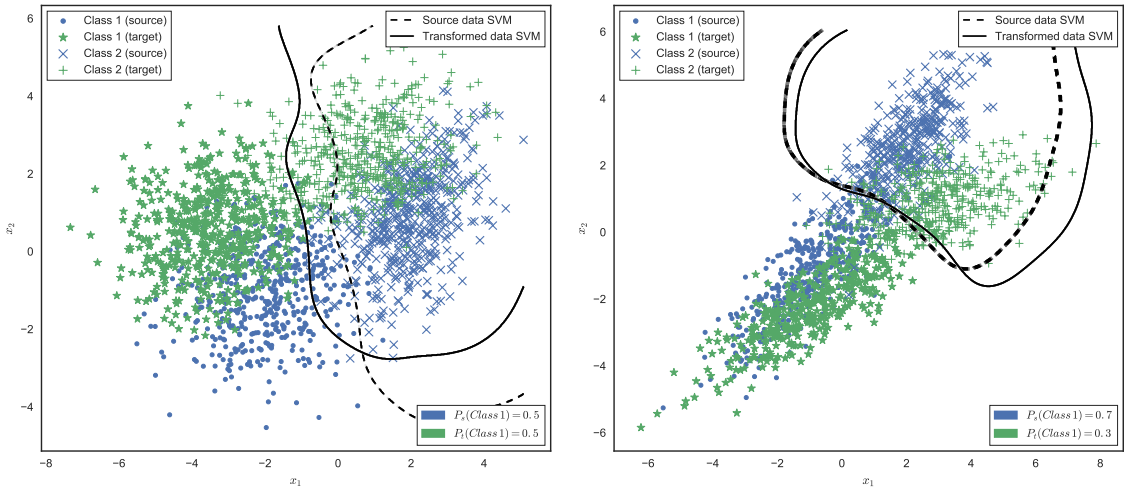


Figure 6.2. SVM classifier adaptation visualization. Decision boundaries for RBF kernel SVM's trained on the source training set and the transformed source training set. Left panel: simulation under ConS. Right panel: decision boundaries in a second simulation under GeTarS. The class distributions for each dataset are shown on the lower right corner.

Table 6.1. F1 scores for simulated ConS and GeTarS experiments.

	ConS		GeTarS	
Classifier	Original	Transformed	Original	Transformed
SVM	88%	95%	87%	93%
RF	84%	94%	85%	91%

Then, we simulated the GeTarS scenario by having a different class distribution between the source and target datasets. The two right panels in Figure 6.1 show the generated datasets and the fitted model in the same manner.

Figure 6.2 shows how the decision boundary of a radial basis kernel (RBF) support vector machine (SVM) adapts to classify better in the target dataset when trained with the transformed data.

The classification F1 scores for both experiments are shown in Table 6.1.

Note that even tough each mixture component suffers a linear transformation, the overall transformation is not necessarily linear since the transformation under each component

is combined with the others. For example, it is possible to rotate or scale some parts of the space while maintaining others relatively constant. This suggests that it is possible to capture more complex transformations, such as non-affine transformations where collinearity, line parallelism, convexity, length and area ratios, etc., are not preserved.

6.3. Real Datasets

We apply our method to variable star classification using lightcurves from three different survey catalogs: the *Expérience pour la Recherche d'Objets Sombres II* (EROS) survey, the Massive Compact Halo Object (MACHO) survey and the High Cadence Transient Survey (HiTS). Sections 6.3.1, 6.3.2 and 6.3.3 contain a brief description of each survey, followed by a comparison in Section 6.3.4.

We extract features from the lightcurves using the Feature Analysis for Time Series (FATS) Python package (Nun et al., 2015). In addition to the two top features according to the importance ranking presented by Nun et al. (2015), we selected the mean apparent magnitude in the band with the lowest frequency (herein referred to as "Mean") and the skewness of the distribution of all the observed apparent magnitudes in the lowest frequency band in a light curve (herein referred to as "Skew"). We know that the mean magnitude is a proxy of the absolute magnitude for MACHO and EROS. Since these two surveys observe the Magellanic Clouds, the distance to the observed stars is approximately constant. We found that using 5 mixture components was enough to get reasonable results. The extracted features are described in Table 6.2.

6.3.1. The EROS Survey

The *Expérience pour la Recherche d'Objets Sombres II* (EROS-II or simply referred to as EROS in this paper) collaboration is an astronomical survey that started operation in 1990 at the European Southern Observatory at La Silla, Chile. Its main purpose was to search for microlensing events in the directions of the Magellanic Clouds, the Galactic

Center and four areas within the Galactic Plane (Beaulieu et al., 1995). The EROS-II instrument was a 0.98 m diameter Ritchey-Chrétien telescope located at the European Southern Observatory in La Silla, Chile. It operated at $f/5$ with a 0.7° RA and 1.4° Dec field of view. The telescope featured a dichroic beam splitter that allowed for simultaneous observations in two wide pass-bands – a blue one and a red one (Perdereau, 1998; Bauer et al., 1998; European Southern Observatory, 2017).

6.3.2. The MACHO Survey

The Massive Compact Halo Object (MACHO) project is a gravitational microlensing survey whose main goal was to find massive compact halo objects in the Milky Way halo

Table 6.2. Features used in the experiments

	Name	Description
1	Color	Difference between the mean apparent magnitude of observations from two different bands. We used the two lowest frequency bands available in each survey.
2	Mean	Mean apparent magnitude. The arithmetic mean of all the lightcurve observations. We used the lowest frequency band available in each survey.
3	Psi_CS	Range of a cumulative sum of apparent magnitudes applied to the phase-folded light curve (using the period estimated from the Lomb-Scargle method). We used the lowest frequency band.
4	Skew	The skewness of the distribution of observed apparent magnitudes in each light curve in the lowest frequency band.

Notes. Names and descriptions are as in the FATS package. See Nun et al. (2015) for a detailed definition.

The range of a cumulative sum R_{cs} of a light curve with N observations $m_1, m_2, ..., m_N$ is defined as (Ellaway, 1978):

$$R_{cs} = max(S) - min(S)$$

$$S = \frac{1}{N\sigma} \sum_{i=1}^{l} (m_i - \bar{m})$$

With $l = 1, 2, ..., N$, and $\bar{m}$, σ are the mean magnitude and the standard deviation of the magnitudes, respectively.

to assess their mass contribution (Alcock et al., 1997). Observations were made in the direction of the Large Magellanic Cloud (LMC), the Small Magellanic Cloud (SMC) and the Galactic Bulge. The MACHO project instrument was the 1.27 meter telescope at Mount Stromlo Observatory, Australia. It operated at $f/3.8$ with a 1° diameter field of view. A dichroic beamsplitter and filters allowed image capture in the "red" (approx. 6,300 - 7,800 Å) and "blue" bands (approx. 4,500 - 6,300 Å). Each image has a sky coverage of 0.72 x 0.72 degrees. The exposure times were of 300 seconds for the LMC, 600 seconds for the SMC and 150 seconds for the bulge (Alcock et al., 1997; Hart et al., 1996; Cook, 1995). In this paper we only consider the LMC data.

6.3.3. The HiTS Survey

The High Cadence Transient Survey (HiTS) first campaign started in 2013 with the objective of exploring transient and periodic objects with characteristic timescales between a few hours and days. This discovery survey uses high cadency data obtained from the Dark Energy Camera (DECam) mounted on a 4 m telescope at Cerro Tololo Interamerican Observatory (CTIO). The large *etendue* (product of collecting area and field of view) of the DECam allows the observation of apparent magnitudes as low as 24.5 mag. It operated at $f/2.7$ with a 2.2° field of view (Förster et al., 2016; Flaugher, 2006; Fukugita et al., 1996).

6.3.4. Dataset Comparison

Among the three surveys studied, MACHO and EROS are the most similar. They observed along two comparable bands ("blue" band wavelength limits differ in less than 7%, and "red" band wavelength limits in less than 13%), had an analogous limiting magnitude (a difference of half a magnitude), and we used data from the same observed area for our experiments. However, as we can see from Tables 6.5, 6.6, and 6.10, the classification performance drops significantly when training in one of these datasets and classifying in the other. F1 score drops from 85% to 60% when training in MACHO and classifying in

Table 6.3. Telescope and survey comparison

	EROS	MACHO	HiTS
Instrument	"MarLy" Ritchey-Chrétien Telescope	Great Melbourne Telescope (Renovation)	Víctor M. Blanco Telescope
Institution	European Southern Observatory	Australian National University	Cerro Tololo Interamerican Observatory
Location	La Silla, Chile	Mount Stromlo, Australia	Cerro Tololo, Chile
Altitude (masl)	2,375	770	2,207
Diameter (m)	0.98	1.27	4
Aperture	$f/5$	$f/3.8$	$f/2.7$
Field of view	0.7° (RA) 1.4° (Dec)	1°	2.2°
Bands (Å)	Blue (4,200 - 6,500) Red (6,500 - 9,000)	Blue (4,500 - 6,300) Red (6,300 - 7,800)	Blue-green (4,000 - 6,200) Red (4,850 - 7,650) Far-red (6,200 - 9,200) Near-infrared (7,900 - 10,000)
Limiting magnitude (visual)	20	20.5	24.5
Observed area[1]	Magellanic Clouds	Magellanic Clouds	Southern Galactic Cap

EROS using Random Forest with four features, versus training and testing in MACHO. When training in EROS and classifying in MACHO, the drop is from 90% to 71%.

In contrast, EROS and MACHO are comparatively more dissimilar than HiTS. Some differences are: HiTS observed in four bands instead of two, it had a limiting magnitude around four points higher, a wider field of view, and it observed a different region of the

Table 6.4. Dataset class composition

	Class	Description	# EROS	# MACHO	# HiTS
1	CEP	Cepheids.	472	14	35
2	EB	Eclipsing Binaries.	12,061	207	15
3	QSO	Quasars.	217	55	2,309
4	RRLYR	RR Lyrae.	11,787	611	105
5	LPV	Long Period Variables.	1,468	217	0
		Total	26,005	1,104	2,464

sky. Table 6.3 shows a comparison of the three surveys and their instruments. Unsurprisingly, the classification performance drops dramatically when using HiTS as a training set for classifying in EROS or MACHO, and vice versa. When classifying in HiTS using Random Forest and four features, the F1 score drops from 94% to 8% when training in EROS, and to 11 % when training in MACHO. When using HiTS to train, the F1 score using Random Forest and four features drops from 85% to 3% classifying in EROS, and from 90% to 3% classifying in MACHO.

The datasets used in our experiments are also different in the amount of labeled data available. While our labeled dataset for EROS has more than 25,000 labeled stars, the MACHO and HiTS datasets have only about 1,000 and 2,5000, respectively. Moreover, the class representation is different in each dataset. Table 6.4 shows a description and the amount of instances for each class present in the labeled datasets of each survey.

6.3.5. Baseline Results

Tables 6.5, 6.6 and 6.7 present the per-class classification F1 scores obtained by cross-validation in each dataset. These results serve as a baseline for the performance that can be achieved by both training and testing in a same dataset using the same features we transfer in our experiments.

[1]Of the data used in the experiments. See the survey description for the complete observation area.

Table 6.5. Baseline F1 scores for variable star classification in EROS

		Random Forest			Support Vector Machine		
	Class	2 Features	3 Features	4 Features	2 Features	3 Features	4 Features
1	CEP	53%	72%	79%	32%	16%	72%
2	EB	75%	80%	85%	76%	78%	84%
3	QSO	1%	21%	24%	0%	0%	0%
4	RRLYR	74%	80%	84%	77%	79%	83%
5	LPV	90%	92%	92%	91%	91%	92%
	Weighted Average	74%	80%	85%	76%	77%	83%

Table 6.6. Baseline F1 scores for variable star classification in MACHO

		Random Forest			Support Vector Machine		
	Class	2 Features	3 Features	4 Features	2 Features	3 Features	4 Features
1	CEP	52%	61%	69%	13%	13%	80%
2	EB	73%	74%	82%	65%	66%	82%
3	QSO	33%	67%	59%	0%	0%	10%
4	RRLYR	89%	91%	93%	89%	89%	93%
5	LPV	98%	98%	98%	98%	98%	98%
	Weighted Average	84%	88%	90%	81%	81%	87%

Table 6.7. Baseline F1 scores for variable star classification in HiTS

		Random Forest			Support Vector Machine		
	Class	2 Features	3 Features	4 Features	2 Features	3 Features	4 Features
1	CEP	35%	41%	35%	10%	10%	33%
2	EB	0%	42%	33%	0%	0%	0%
3	QSO	97%	97%	98%	97%	97%	97%
4	RRLYR	24%	36%	46%	22%	23%	19%
	Weighted Average	92%	94%	94%	92%	92%	93%

6.3.6. 2D Experiment Visualization

To illustrate the functioning of the model, we first apply it to a two-dimensional space using the Mean and Color features (see table 6.2). We use EROS as the source dataset and MACHO as the target. Hence our goal is to classify MACHO instances using the transformed EROS training set. We use 10,000 unlabeled instances from each of the domains

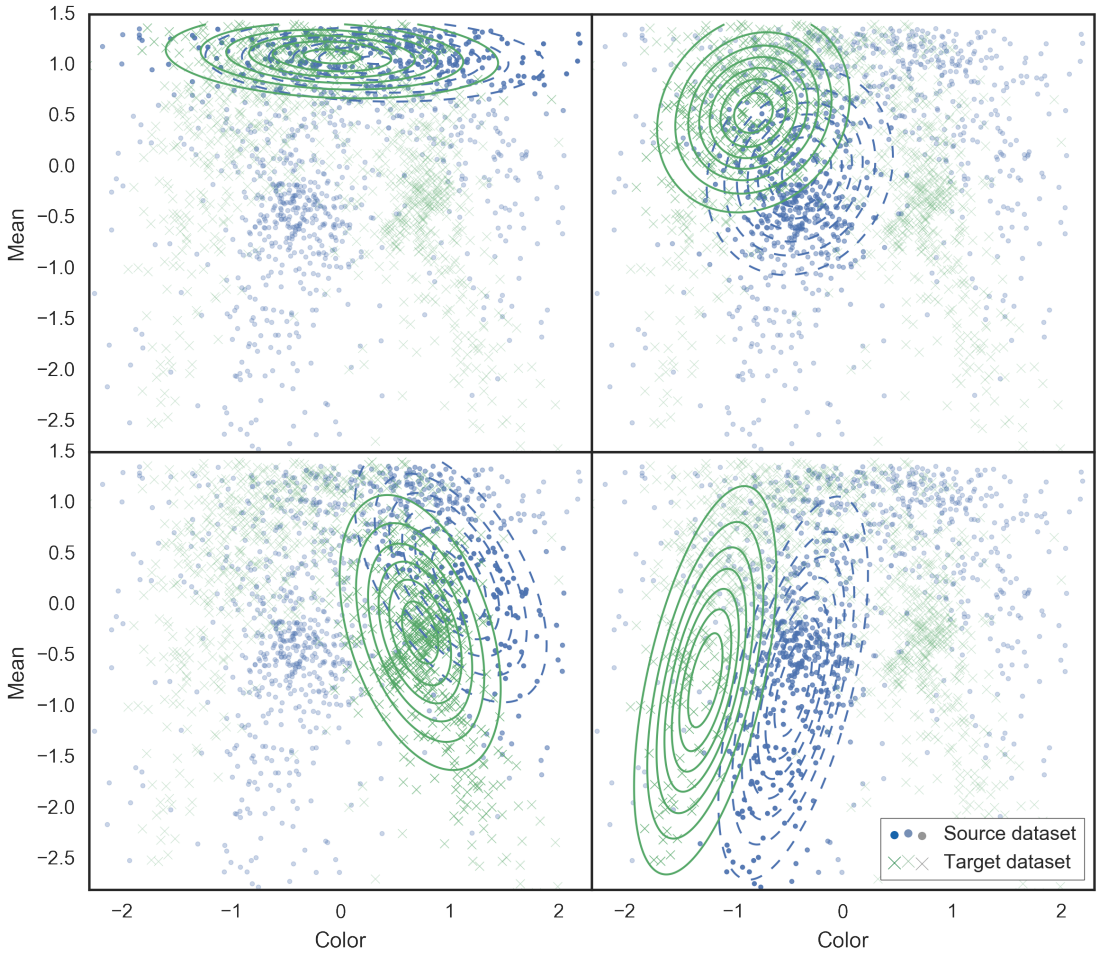


Figure 6.3. Main transformation components from EROS to MACHO. The four components with the highest weights for a 2D transformation from the EROS to the MACHO dataset are shown. The source components are shown with a blue dashed line and the target components with a continuous green line. Point transparency represents the responsibility of the plotted component for that point, with higher opacity representing higher responsibility.

to fit our transformation model with 5 components. Figure 6.3 shows the transformation found for the four components with the highest weights. Note how each component "focuses" on a different region of the distribution and then transforms instances to "match' the target distribution region.

Table 6.8. F1 scores for classification experiments with 2 features

	EROS → MACHO		EROS → HiTS	
Classifier	Original	Transformed	Original	Transformed
RF	51%	68%	6%	27%
SVM	48%	65%	0%	0%

	MACHO → EROS		MACHO → HiTS	
Classifier	Original	Transformed	Original	Transformed
RF	47%	70%	13%	50%
SVM	59%	72%	1%	0%

	HiTS → EROS		HiTS → MACHO	
Classifier	Original	Transformed	Original	Transformed
RF	1%	10%	2%	7%
SVM	1%	4%	1%	6%

Notes. Column "Original" displays the score obtained when training in the untransformed source dataset and testing on the target dataset. Column "Transformed" shows the score when training on the transformed source dataset and testing on the target dataset.

6.3.7. Further Experiments

We continue applying the method to an increasing number of features from table 6.2. We repeat the experiments using all possible dataset pairs. Tables 6.8, 6.9 and 6.10 show the F1 scores for this experiment when transforming two, three and four features, respectively. The "Original" column displays the score obtained when training in the untransformed source dataset and testing on the target dataset. The "Transformed" column shows the score when training on the transformed source dataset and testing on the target dataset.

Tables 6.11 through 6.28 show the F1 scores for each class in each experiment when transforming two, three, and four features.

Table 6.9. F1 scores for classification experiments with 3 features

	EROS → MACHO		EROS → HiTS	
Classifier	Original	Transformed	Original	Transformed
RF	63%	73%	2%	35%
SVM	75%	81%	0%	1%

	MACHO → EROS		MACHO → HiTS	
Classifier	Original	Transformed	Original	Transformed
RF	50%	62%	19%	63%
SVM	68%	74%	56%	60%

	HiTS → EROS		HiTS → MACHO	
Classifier	Original	Transformed	Original	Transformed
RF	1%	26%	2%	7%
SVM	0%	5%	1%	4%

Table 6.10. F1 scores for classification experiments with 4 features

	EROS → MACHO		EROS → HiTS	
Classifier	Original	Transformed	Original	Transformed
RF	71%	84%	8%	21%
SVM	84%	90%	1%	1%

	MACHO → EROS		MACHO → HiTS	
Classifier	Original	Transformed	Original	Transformed
RF	60%	70%	11%	30%
SVM	78%	78%	45%	44%

	HiTS → EROS		HiTS → MACHO	
Classifier	Original	Transformed	Original	Transformed
RF	3%	11%	3%	16%
SVM	0%	0%	1%	1%

Notes. Column "Original" displays the score obtained when training in the untransformed source dataset and testing on the target dataset. Column "Transformed" shows the score when training on the transformed source dataset and testing on the target dataset.

Table 6.11. F1 scores for classification using 2 features to transfer from EROS to MACHO

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	3%	4%	7%	4%
2	EB	31%	37%	31%	36%
3	QSO	0%	0%	0%	0%
4	RRLYR	48%	79%	43%	77%
5	LPV	93%	90%	94%	81%
	Weighted Average	51%	68%	48%	65%

Table 6.12. F1 scores for classification using 3 features to transfer from EROS to MACHO

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	13%	0%	7%	5%
2	EB	36%	45%	48%	59%
3	QSO	48%	40%	53%	38%
4	RRLYR	63%	79%	80%	90%
5	LPV	96%	95%	95%	94%
	Weighted Average	63%	73%	75%	81%

Table 6.13. F1 scores for classification using 4 features to transfer from EROS to MACHO

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	7%	33%	12%	44%
2	EB	54%	73%	71%	81%
3	QSO	19%	37%	50%	60%
4	RRLYR	74%	91%	89%	95%
5	LPV	95%	93%	96%	94%
	Weighted Average	71%	84%	84%	90%

Table 6.14. F1 scores for classification using 2 features to transfer from EROS to HiTS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	15%	0%	15%	0%
2	EB	2%	1%	2%	2%
3	QSO	6%	29%	0%	0%
4	RRLYR	3%	6%	0%	8%
	Weighted Average	6%	27%	0%	0%

Table 6.15. F1 scores for classification using 3 features to transfer from EROS to HiTS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	0%	0%	0%	0%
2	EB	1%	2%	1%	1%
3	QSO	2%	37%	0%	0%
4	RRLYR	11%	15%	5%	21%
	Weighted Average	2%	35%	0%	1%

Table 6.16. F1 scores for classification using 4 features to transfer from EROS to HiTS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	0%	0%	0%	0%
2	EB	1%	2%	1%	1%
3	QSO	8%	21%	0%	0%
4	RRLYR	10%	14%	17%	16%
	Weighted Average	8%	21%	1%	1%

Table 6.17. F1 scores for classification using 2 features to transfer from MACHO to EROS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	10%	50%	5%	46%
2	EB	57%	68%	67%	67%
3	QSO	2%	3%	0%	1%
4	RRLYR	42%	73%	55%	77%
5	LPV	31%	85%	43%	91%
	Weighted Average	47%	70%	59%	72%

Table 6.18. F1 scores for classification using 3 features to transfer from MACHO to EROS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	4%	3%	11%	0%
2	EB	59%	62%	72%	72%
3	QSO	12%	16%	25%	21%
4	RRLYR	45%	64%	70%	80%
5	LPV	36%	73%	52%	82%
	Weighted Average	50%	62%	68%	74%

Table 6.19. F1 scores for classification using 4 features to transfer from MACHO to EROS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	14%	25%	28%	7%
2	EB	63%	76%	80%	81%
3	QSO	16%	2%	27%	5%
4	RRLYR	63%	69%	82%	81%
5	LPV	32%	58%	48%	76%
	Weighted Average	60%	70%	78%	78%

Table 6.20. F1 scores for classification using 2 features to transfer from MACHO to HiTS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	12%	9%	18%	0%
2	EB	1%	2%	0%	1%
3	QSO	14%	53%	1%	0%
4	RRLYR	1%	2%	1%	1%
	Weighted Average	13%	50%	1%	0%

Table 6.21. F1 scores for classification using 3 features to transfer from MACHO to HiTS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	17%	0%	5%	0%
2	EB	1%	1%	1%	3%
3	QSO	20%	67%	59%	63%
4	RRLYR	2%	6%	9%	9%
	Weighted Average	19%	63%	56%	60%

Table 6.22. F1 scores for classification using 4 features to transfer from MACHO to HiTS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	22%	3%	15%	0%
2	EB	1%	3%	2%	2%
3	QSO	11%	38%	47%	47%
4	RRLYR	16%	9%	13%	10%
	Weighted Average	11%	36%	45%	44%

Table 6.23. F1 scores for classification using 2 features to transfer from HiTS to EROS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	22%	7%	38%	2%
2	EB	0%	13%	0%	0%
3	QSO	2%	2%	2%	2%
4	RRLYR	0%	7%	0%	7%
	Weighted Average	1%	10%	1%	4%

Table 6.24. F1 scores for classification using 3 features to transfer from HiTS to EROS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	15%	10%	9%	2%
2	EB	0%	37%	0%	0%
3	QSO	2%	2%	2%	2%
4	RRLYR	0%	15%	0%	11%
	Weighted Average	1%	26%	0%	5%

Table 6.25. F1 scores for classification using 4 features to transfer from HiTS to EROS

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	26%	19%	5%	0%
2	EB	4%	20%	0%	0%
3	QSO	2%	2%	2%	2%
4	RRLYR	0%	1%	0%	0%
	Weighted Average	3%	11%	0%	0%

Table 6.26. F1 scores for classification using 2 features to transfer from HiTS to MACHO

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	11%	6%	18%	0%
2	EB	6%	9%	0%	0%
3	QSO	12%	11%	12%	13%
4	RRLYR	0%	6%	0%	8%
	Weighted Average	2%	7%	1%	6%

Table 6.27. F1 scores for classification using 3 features to transfer from HiTS to MACHO

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	10%	0%	9%	0%
2	EB	0%	7%	0%	0%
3	QSO	12%	14%	12%	13%
4	RRLYR	2%	7%	0%	5%
	Weighted Average	2%	7%	1%	4%

Table 6.28. F1 scores for classification using 4 features to transfer from HiTS to MACHO

	Class	Unadapted RF	Adapted RF	Unadapted SVM	Adapted SVM
1	CEP	18%	9%	32%	0%
2	EB	10%	68%	0%	0%
3	QSO	12%	8%	12%	11%
4	RRLYR	0%	0%	0%	0%
	Weighted Average	3%	16%	1%	1%

7. CONCLUSIONS

We present a method for survey invariant classification of variable stars by transforming feature representations between surveys. Our probabilistic model does not assume a particular classifier and can be used to work with the data of one survey as if it belonged to the other, allowing for the reuse of existing training sets in related domains where no, or not enough, labeled data is available. This will become increasingly important for practical applications as the volume of unlabeled data keeps growing surpassing the rate at which labeled data becomes available. No explicit assumptions are made of the domain shift, which we consider to follow a generalized target shift. We apply our method to simulated data and to three astronomical surveys. First, we do inference in our model to find the transformation parameters. Then, we apply the transformation to the source domain's training set, train a classifier and test in the target domain. Our results show that a significant performance gain in classification can be obtained by adapting a training set with our model, only making use of unlabeled data in both domains.

REFERENCES

Alcock, C., Allsman, R. A., Alves, D., Axelrod, T. S., Bennett, D. P., Cook, K. H., ... Lehner, M. J. (1997, April). The Macho Project: 45 Candidate Microlensing Events from the First Year Galactic Bulge Data. *The Astrophysical Journal*, *479*, 119-146.

Barnard, J., McCulloch, R., & Meng, X.-L. (2000). Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica*, 1281–1311.

Bauer, F., De Kat, J., Afonso, C., Aubourg, E., Aune, S., Bareyre, P., ... others (1998). The two EROS 4K x 8K CCD mosaic cameras. In *Optical detectors for astronomy* (pp. 191–202). Springer.

Beaulieu, J., Ferlet, R., Grison, P., Vidal-Madjar, A., Kneib, J., Maurice, E., ... Moreau, O. (1995). Spectroscopic studies of the two EROS candidate microlensed stars. *Astronomy and Astrophysics*, *299*, 168.

Benavente, P., Protopapas, P., & Pichara, K. (2017). Automatic survey-invariant classification of variable stars. *The Astrophysical Journal*, *845*(2), 147.

Benitez, N. (2000). Bayesian photometric redshift estimation. *The Astrophysical Journal*, *536*(2), 571.

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer.

Blitzer, J., McDonald, R., & Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 120–128).

Blomme, J., Sarro, L., O'Donovan, F., Debosscher, J., Brown, T., Lopez, M., ... Dunham,

E. (2011). Improved methodology for the automated classification of periodic variable stars. *Monthly Notices of the Royal Astronomical Society*, *418*(1), 96–106.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152).

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

Brescia, M., Cavuoti, S., D'Abrusco, R., Longo, G., & Mercurio, A. (2013). Photometric redshifts for quasars in multi-band surveys. *The Astrophysical Journal*, *772*(2), 140.

Bromley, B. C., Press, W. H., Lin, H., & Kirshner, R. P. (1998). Spectral classification and luminosity function of galaxies in the las campanas redshift survey. *The Astrophysical Journal*, *505*(1), 25.

Carliles, S., Budavári, T., Heinis, S., Priebe, C., & Szalay, A. S. (2010). Random forests for photometric redshifts. *The Astrophysical Journal*, *712*(1), 511.

Cavuoti, S., Brescia, M., D'Abrusco, R., Longo, G., & Paolillo, M. (2014). Photometric classification of emission line galaxies with machine-learning methods. *Monthly Notices of the Royal Astronomical Society*, *437*(1), 968–975.

Chan, Y. S., & Ng, H. T. (2005). Word sense disambiguation with distribution estimation. In *Proceedings of the 19th international joint conference on artificial intelligence* (pp. 1010–1015).

Christlieb, N., Wisotzki, L., & Grasshoff, G. (2002). Statistical methods of automatic spectral classification and their application to the hamburg/eso survey. *Astronomy & Astrophysics*, *391*(1), 397–406.

Colak, T., & Qahwaji, R. (2009). Automated solar activity prediction: A hybrid computer platform using machine learning and solar imaging for automated prediction of solar

flares. *Space Weather*, *7*(6).

Collister, A. A., & Lahav, O. (2004). Annz: estimating photometric redshifts using artificial neural networks. *Publications of the Astronomical Society of the Pacific*, *116*(818), 345.

Committee for a Decadal Survey of Astronomy and Astrophysics. (2011). *New worlds, new horizons in astronomy and astrophysics*. National Academies Press.

Cook, K. H. (1995). A dual CCD mosaic camera system searching for massive compact halo objects (MACHOs). In *New developments in array technology and applications* (Vol. 167, p. 285).

Daumé III, H. (2009). Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

Dietterich, T. G., et al. (2000). *Ensemble methods in machine learning*. Springer. Retrieved from `https://doi.org/10.1007/3-540-45014-9` doi: 10.1007/3-540-45014-9

Dubath, P., Rimoldini, L., Süveges, M., Blomme, J., López, M., Sarro, L., ... others (2011). Random forest automated supervised classification of hipparcos periodic variable stars. *Monthly Notices of the Royal Astronomical Society*, *414*(3), 2602–2617.

Duffin, K. L., & Barrett, W. A. (1994). Spiders: A new user interface for rotation and visualization of n-dimensional point sets. In *Proceedings of the conference on visualization'94* (pp. 205–211).

Ellaway, P. (1978). Cumulative sum technique and its application to the analysis of peristimulus time histograms. *Electroencephalography and clinical neurophysiology*, *45*(2), 302–304.

European Southern Observatory. (2014, June 19). *Groundbreaking for the E-ELT.* Retrieved 2016-06-15, from http://www.eso.org/public/news/eso1419/

European Southern Observatory. (2017, feb). *MarLy 1-metre telescope (decommissioned).* Retrieved from https://www.eso.org/public/teles-instr/lasilla/marly/

Feigelson, E. D., & Babu, G. J. (2012). Big data in astronomy. *Significance*, *9*(4), 22–25.

Finkel, J. R., & Manning, C. D. (2009). Hierarchical bayesian domain adaptation. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics* (pp. 602–610).

Flaugher, B. (2006). The dark energy survey instrument design. In *Spie astronomical telescopes+ instrumentation* (pp. 62692C–62692C).

Foster, G., Goutte, C., & Kuhn, R. (2010). Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 451–459).

Freed, M., & Lee, J. (2013). Application of support vector machines to the classification of galaxy morphologies. In *Computational and information sciences (ICCIS), 2013 fifth international conference on* (pp. 322–325).

Fukugita, M., Ichikawa, T., Gunn, J., Doi, M., Shimasaku, K., & Schneider, D. (1996). The Sloan digital sky survey photometric system. *The Astronomical Journal*, *111*, 1748.

Förster, F., Maureira, J. C., Martín, J. S., Hamuy, M., Martínez, J., Huijse, P., ... Vera, E. (2016). *The High Cadence Transient Survey (HiTS) - I. survey design and supernova shock breakout constraints.*

GMTO Corporation. (2016). *Overview — Giant Magellan Telescope.* Retrieved 2016-06-15, from http://www.gmto.org/overview/

Gönen, M., & Margolin, A. A. (2014). Kernelized bayesian transfer learning. In *Twenty-eighth aaai conference on artificial intelligence* (pp. 1831–1839).

Gong, B., Shi, Y., Sha, F., & Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on* (pp. 2066–2073).

Gopalan, R., Li, R., & Chellappa, R. (2011). Domain adaptation for object recognition: An unsupervised approach. In *Computer vision (iccv), 2011 IEEE international conference on* (pp. 999–1006).

GRANTECAN S.A. (2013). *Gran telescopio CANARIAS.* Retrieved 2016-06-15, from `http://www.gtc.iac.es/`

Graßhoff, G. (2013). *The history of ptolemy's star catalogue*. Springer New York. Retrieved from `https://books.google.com/books?id=pR_nBwAAQBAJ`

Hart, J., van Hermelen, J., Hovey, G., Freeman, K., Peterson, B., Axelrod, T., ... others (1996). The telescope system of the MACHO program. *Publications of the Astronomical Society of the Pacific*, *108*(720), 220.

Hofmann, M. (2006). Support vector machines-kernels and the kernel trick.

Huang, J., Gretton, A., Borgwardt, K. M., Schölkopf, B., & Smola, A. J. (2006). Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems* (pp. 601–608).

Jiang, J. (2008). A literature survey on domain adaptation of statistical classifiers. *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, *3*.

Johnston, K. B., & Oluseyi, H. M. (2017). Generation of a supervised classification algorithm for time-series variable stars with an application to the linear dataset. *New Astronomy*, *52*, 35–47.

Jurić, M., Kantor, J., Lim, K., Lupton, R. H., Dubois-Felsmann, G., Jenness, T., ... others (2015). The lsst data management system. *arXiv preprint arXiv:1512.07914*.

Kholopov, P. N., Samus', N. N., Frolov, M. S., Goranskij, V. P., Gorynya, N. A., Kazarovets, E., et al. (1985). General catalog of variable stars.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1), 307–319.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques*. MIT Press.

Kulis, B., Saenko, K., & Darrell, T. (2011). What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer vision and pattern recognition (CVPR), 2011 IEEE conference on* (pp. 1785–1792).

LBTO. (2016). *Overview — Large Binocular Telescope Observatory.* Retrieved 2016-06-15, from `http://www.lbto.org/overview.html`

Lewandowski, D., Kurowicka, D., & Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of multivariate analysis*, *100*(9), 1989–2001.

Liu, C.-L., Nakashima, K., Sako, H., & Fujisawa, H. (2003). Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*, *36*(10), 2271–2285.

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297).

Murphy, K. (2012). *Machine learning: A probabilistic perspective*. MIT Press. Retrieved from `https://books.google.com/books?id=NZP6AQAAQBAJ`

Nun, I., Pichara, K., Protopapas, P., & Kim, D.-W. (2014). Supervised detection of anomalous light curves in massive astronomical catalogs. *Astrophysical Journal*, *793*(1).

Nun, I., Protopapas, P., Sim, B., Zhu, M., Dave, R., Castro, N., & Pichara, K. (2015). FATS: Feature analysis for time series. *arXiv preprint arXiv:1506.00010*.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, *22*(10), 1345–1359.

Patel, V. M., Gopalan, R., Li, R., & Chellappa, R. (2015). Visual domain adaptation: A survey of recent advances. *Signal Processing Magazine, IEEE*, *32*(3), 53–69.

Perdereau, O. (1998). First results from the EROS-II microlensing experiment. *arXiv preprint astro-ph/9812045*.

Pichara, K., & Protopapas, P. (2013). Automatic classification of variable stars in catalogs with missing data. *The Astrophysical Journal*, *777*(2), 83.

Pichara, K., Protopapas, P., Kim, D.-W., Marquette, J.-B., & Tisserand, P. (2012). An improved quasar detection method in eros-2 and macho lmc data sets. *Monthly Notices of the Royal Astronomical Society*, *427*(2), 1284–1297.

Pichara, K., Protopapas, P., & León, D. (2016). Meta classification for variable stars. *The Astrophysical Journal*.

Ptolemei, C. (1515). *Almagestum.* Petri Liechtenstein.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2009). *Dataset shift in machine learning*. The MIT Press.

Racine, R. (2004). The historical growth of telescope aperture. *Publications of the Astronomical Society of the Pacific*, *116*(815), 77-83. Retrieved from `http://www.jstor.org/stable/10.1086/380955`

Raina, R., Battle, A., Lee, H., Packer, B., & Ng, A. Y. (2007). Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on machine learning* (pp. 759–766).

Rogers, J. H. (1998, February). Origins of the ancient constellations: I. The Mesopotamian traditions. *Journal of the British Astronomical Association*, *108*, 9-28.

Russell, S., & Norvig, P. (2009). *Artificial intelligence: A modern approach* (3rd ed.). Upper Saddle River, NJ, USA: Prentice Hall Press.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *3*(3), 210–229.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, *90*(2), 227–244.

Sterken, C., & Jaschek, C. (2005). *Light curves of variable stars: A pictorial atlas*. Cambridge University Press.

TMT Observatory Corporation. (2007, September 12). *Thirty meter telescope construction proposal.*

Vapnik, V. (1963). Pattern recognition using generalized portrait method. *Automation and remote control*, *24*, 774–780.

Zhang, K., Muandet, K., & Wang, Z. (2013). Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* (pp. 819–827).

Zhang, Y., & Zhao, Y. (2015). Astronomy in the big data era. *Data Science Journal*, *14*.