Automated Supernova Ia Classification Using Adaptive Learning Techniques

Kinjal Dhar Gupta*, Renuka Pampana*, Ricardo Vilalta*, Emille E. O. Ishida†, Rafael S. de Souza‡
*Department of Computer Science, University of Houston
4800 Calhoun Rd., Houston, Texas 77204-3010, USA
Email: {kdhargupta,rpampana,rvilalta@uh.edu}
†Laboratoire de Physique Corpusculaire, Université Blaise Pascal
4 Avenue Blaise Pascal, TSA 60026, CS 60026, 63178, Aubière Cedex, France
Email: emille.ishida@clermont.in2p3.fr
‡MTA Eötvös Loránd University, EIRSA, Astrophysics Research Group
Budapest, Hungary
Email: rafael@caesar.elte.hu

***Abstract*—While the current supernova (SN) photometric classification system is based on high resolution spectroscopic observations, the next generation of large scale surveys will be based on photometric light curves of supernovae gathered at an unprecedented rate. Developing an efficient method for SN photometric classification is critical to cope with the rapid growth of data volumes in current astronomical surveys. In this work, we present an adaptive mechanism that generates a predictive model to identify a particular class of SN known as Type Ia, when the source set is made of spectroscopic data, while the target set is made of photometric data. The method is applied to simulated data sets derived from the Supernova Photometric Classification Challenge, and preprocessed using Gaussian Process Regression for all objects with at least 1 observational epoch before -3 and after +24 days since the SN maximum brightness. The main difficulty lies in the compatibility of models between spectroscopic (source) data and photometric (target) data, since the underlying distributions on both, source and target domains, are expected to be significantly different. A solution is to adapt predictive models across domains. Our methodology exploits machine learning techniques by combining two concepts: 1) domain adaptation is used to transfer properties from the source domain to the target domain; and 2) active learning is used as a means to rely on a set of confident labels on the target domain. We show how a combination of both concepts leads to high generalization (i.e., predictive) performance.**

I. Introduction

The ever-increasing importance of large scale-surveys is a major trend in contemporary observational astronomy. The unexpected broad scientific achievements of current sky surveys, most notably the Sloan Digital Sky Survey[1] (SDSS) [1], inspired a new generation of astronomical instruments. Projects like the Dark Energy Survey[2] (DES) [2], the Large Synoptic Survey Telescope[3] (LSST) [3] and the ESA EUCLID[4] mission are expected to revolutionize our understanding of the Universe.

[1]http://www.sdss.org/
[2]https://www.darkenergysurvey.org/
[3]https://www.lsst.org/
[4]http://sci.esa.int/euclid/

Beyond the technological challenges in dealing with such large data sets (e.g., LSST is supposed to run for 10 years, gathering every 5 nights the same volume of data SDSS accumulated over 1.5 decades [4]), the necessity of high-resolution (spectroscopic) observations is a crucial, and many times overlooked, bottleneck.

Spectroscopy can be described as the high-resolution decomposition of electromagnetic radiation as a function of wavelength. It allows one to determine the presence of individual chemical elements (spectral lines) and to infer the distance (redshift) to extragalactic sources, the two basic pieces of information which enable parameter inference from astronomical data. However, it is also an expensive and time-consuming process. In the current data paradigm, it is unfeasible to obtain spectroscopic measurements for all cataloged objects, and the situation will not be easier in the light of new instruments. Most of the observations are/will be obtained through its low-resolution counterpart: photometry.

Photometric measurements summarize the intensity of electromagnetic radiation in a handful of broad wavelength windows (filters). Therefore, although the brightness throughout the electromagnetic spectrum is measured, information on individual spectral lines is not accessible. If one wishes to make use of the bulk of information contained in these large datasets, the challenge is to infer spectroscopic properties from purely photometric data.

One of the most urgent problems related to this subjects is the classification of supernovae (SNe). Almost all new surveys have the photometric observation of type Ia SNe for cosmology among their key scientific goals. However, most of them will deliver purely photometric data. In trying to solve this problem, a number of photometric classifiers were suggested [5], [6]. Unfortunately, up to this point, cosmological results from purely photometric samples are not in complete agreement with spectroscopic ones [7]. Consequently, the last data release of the SDSS SN survey [8] contains $\sim$4600 SN candidates, from which at least $\sim$20% are expected to be type Ias, but the spectroscopically confirmed sample used

for cosmology holds ∼500 objects. In other words, there are approximately twice as many SNe stored in the SDSS database which are simply not used due to the lack of a reliable photometric classification.

One strategy to deal with spectroscopic and photometric datasets, both types exhibiting different distribution, is to use adaptive learning techniques, also known in machine learning as *domain-adaptation* techniques. Since its inception, domain adaptation has seen numerous theoretical and real-world applications [9]–[15]. The central idea is to exploit information from a source domain for which class labels are known with high confidence, on a target domain where class labels are scarce and the underlying distribution between the two domains is different. In our work, the source domain corresponds to spectroscopic data where supernovae are confidently classified (as type class Ia or different). The target domain corresponds to the new generation of abundant photometric light-curve datasets where class labels are scarce. Our goal is to take advantage of the source domain to attain high predictive performance on the target domain.

Our experimental work shows how the direct application of domain adaptation techniques does improve generalization performance, albeit to a small degree. To boost the performance of the predictive model on the target dataset, it is necessary to invoke *active-learning* techniques. Active learning points to those few instances on the target set where knowing the class label with confidence suffices to attain an accurate model. We conclude by showing how the automated classification of Supernovae Ia is best achieved when domain adaptation and active learning are combined synergistically.

This paper is organized as follows. Section II explains basic concepts in classification, domain adaptation, and active learning. Section III explains our approach to combine domain adaptation with active learning. Section IV shows our empirical results. Lastly, Section V gives a summary and conclusions.

II. Preliminaries

A. Basic Notation

A classifier receives as input a set of training examples, $T = \{(\mathbf{x_i}, y_i)\}_{i=1}^{p}$, where $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ is a vector in the input space $\mathcal{X}$, and y is a value in the (discrete) output space $\mathcal{Y}$. We assume the training set T consists of independently and identically distributed (i.i.d.) examples obtained according to a fixed but unknown joint probability distribution, $P(\mathbf{x}, y)$, in the input-output space $\mathcal{X} \times \mathcal{Y}$. The outcome of the classifier is a hypothesis or function $f_\theta(\mathbf{x})$ (parameterized by θ) mapping the input space to the output space, $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$. Function f_θ is drawn from a space of functions or hypotheses $\mathcal{H}$.

We commonly choose the hypothesis that minimizes the expected value of a loss function $L(y, f(\mathbf{x}|\theta))$, also known as the risk:

$$R(\theta, P(\mathbf{x}, y)) = E_{\sim P}[L(y, f(\mathbf{x}|\theta))] \quad (1)$$

where we typically adopt the zero-one loss function $L(y, f(\mathbf{x}|\theta)) = 1_{\{\mathbf{y} \neq f(\mathbf{x}|\theta)\}}(\mathbf{x})$; $1(\cdot)$ is an indicator function.

We consider the (domain adaptation) scenario where we work with two domains: source and target, from which we gather two samples $T_s = \{(\mathbf{x_i}, y_i)\}_{i=1}^{p}$ and $T_t = \{\mathbf{x_i}\}_{i=1}^{q}$. Our principal goal is to obtain an accurate predictive model from the target sample, for which one is allowed to exploit information from the source sample. While T_s is drawn from a joint probability distribution, $P_s(\mathbf{x}, y)$, T_t is drawn from the marginal distribution $P_t(\mathbf{x})$ according to a different joint distribution, $P_t(\mathbf{x}, y)$, such that $P_t(\mathbf{x}) = \int_y P_t(\mathbf{x}, y) d_y$. Notice that T_{t} lacks class labels, and hence poses a special challenge during classification.

B. Domain Adaptation and Active Learning

Domain adaptation, a subfield of transfer learning [16], aims at learning a task by leveraging experience gained on previous tasks [9]. The field has recently gained much popularity because many real-world problems are characterized by having samples exhibiting different distributions [17], [18]. Many approaches have been proposed in domain adaptation. Early work, for example, focused on re-weighting the source dataset by increasing the weight of examples that overlap with regions of high density in the target domain (as estimated from the target dataset). A piece of work along these lines is known as covariate shift [19]–[23]. Other approaches include searching for a subspace where source and target overlap [24]–[26]; theoretical work considering the distance between source and target distributions [9], [10], [27]; and using regularization terms to learn models that perform well on both source and target domains [28].

Active learning provides a mechanism to query those unlabeled examples deemed relevant to build an accurate classifier. In the context of domain adaptation, active learning can be invoked to query examples in the target dataset [13], [29]–[33], to overcome the common difficulty of working with a target dataset lacking in class labels.

III. Methodology

Our approach to build accurate classifiers from photometric data (target) while exploiting information from spectroscopic data (source) is as follows. We first reduce the dimensionality of the data, followed by the application of domain adaptation to transfer information from source to target. Finally, we apply active learning to increase the classification accuracy and purity of the target classifier. The following sections describe our methodology in detail.

A. Dimensionality Reduction

Our first step aims at reducing the number of dimensions of the source data and the target data using Kernel Principal Component Analysis (Kernel PCA); the technique is a generalization of Principal Component Analysis (PCA) by using non-linear components, as proposed by Schölkopf et al., 1997 [34]. Unlike PCA, Kernel PCA finds principal components in the *feature space* $\mathcal{F}$, rather than the original attribute (input) space $\mathcal{X}$. In Kernel PCA, the data is (implicitly) mapped to $\mathcal{F}$ through a transformation $\Phi = R^n \rightarrow \mathcal{F} \in R^m$, where $m \geq n$.

Assuming the data is centered with zero mean, the covariance matrix in $\mathcal{F}$ (using a sample of size p) is given by:

$$C_{\mathcal{F}} = \frac{1}{p}\sum_{i=1}^{p}\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^T \tag{2}$$

We denote by $K_{\mathcal{F}}$ the Kernel Matrix, where

$$K_{\mathcal{F}}(\mathbf{x_i}, \mathbf{x_j}) = \Phi(\mathbf{x_i})^T\Phi(\mathbf{x_j}) = k(\mathbf{x_i}, \mathbf{x_j}) \tag{3}$$

The Kernel Matrix, also known as Gram Matrix, stores the dot product of two vectors lying in feature space $\mathcal{F}$ without actually mapping the data into a high dimensional space. This is known as the *kernel trick*, and is achieved through kernel function $k(\mathbf{x_i}, \mathbf{x_j})$. To find principal components, we diagonalize the covariance matrix, given in equation 2, by finding eigenvalues and eigenvectors after solving the equation:

$$\lambda\mathbf{v} = C_{\mathcal{F}}\mathbf{v} \tag{4}$$

where λ stands as an eigenvalue, $\lambda \geq 0$, and $\mathbf{v}$ stands for an eigenvector. Now, there exist coefficients α_i $(i = 1, 2, \ldots, p)$, such that

$$\mathbf{v} = \sum_{i=1}^{p}\alpha_i\Phi(\mathbf{x_i}) \tag{5}$$

Using Gram Matrix $K_{\mathcal{F}}$, we can rewrite equation 4 as

$$p\lambda K_F\boldsymbol{\alpha} = K_F^2\boldsymbol{\alpha} \tag{6}$$

where $\boldsymbol{\alpha}$ denotes the column vector with entries $\alpha_1, \ldots, \alpha_p$. We solve equation 6 to find the eigenvalues and the eigenvectors of the Gram Matrix. In our notation, to every eigenvector $\mathbf{v}^l$, will correspond an eigenvalue λ^l, and coefficient vector $\boldsymbol{\alpha}^l = \{\alpha_i^l\}$.

Now, for any data point $\mathbf{x}$, from source or target dataset, we can compute a projection using the *l*-th eigenvector, $\mathbf{v}^l$, as follows:

$$(\mathbf{v}^l.\Phi(\mathbf{x})) = \sum_{i=1}^{p}\alpha_i^l k(\mathbf{x_i}, \mathbf{x}) \tag{7}$$

All these equations are based on the assumption of centered data with zero mean in feature space. However, projecting the data over $\mathcal{F}$ by using the Kernel Matrix does not ensure centered data. To guarantee this we use the following:

$$K_{ij}^{'} = (K_F - 1_pK_F - K_F1_p + 1_pK_{\mathcal{F}}1_p) \tag{8}$$

where $(1_p)_{ij} = 1/p$ for all i, j. We summarize the steps of Kernel PCA in Algorithm 1.

B. Domain Adaptation Methods

In this section we describe two well-known methods in domain adaptation: Kernel Mean Matching, and Subspace Alignment. We experiment with these methods to show the advantage of using domain adaptation when confronted with datasets displaying unequal distributions.

Algorithm 1 Kernel Principal Component Analysis

1: Compute Gram Matrix $K_{\mathcal{F}}$ from source data using eq. 3.
2: Center Kernel Matrix using eq. 8 to generate K'.
3: Solve eq. 6 as an Eigen-decomposition problem and compute eigenvalues and eigenvectors using K' instead of K_F.
4: Project source and target data using eigenvectors of Kernel Matrix using eq. 7.

1) Kernel Mean Matching: A popular instance-based domain adaptation method is Kernel Mean Matching [35]. Like any other instance-based domain adaptation method, it also assumes that the difference in source $P_s(\mathbf{x}, y)$ and target $P_t(\mathbf{x}, y)$ distributions is caused due to a covariate shift, i.e., $P_s(\mathbf{x}) \neq P_t(\mathbf{x})$, whereas, the conditional probabilities remain the same $P_s(y|\mathbf{x}) = P_t(y|\mathbf{x})$. From the field of importance sampling, we know that the risk on the target dataset can be estimated as follows:

$$R(\theta, P_t(\mathbf{x}, y)) = E_{\sim P_t}[L(y, f(\mathbf{x}|\theta))] \tag{9}$$

$$R(\theta, P_t(\mathbf{x}, y)) = E_{\sim P_s}[\frac{P_t(\mathbf{x}, y)}{P_s(\mathbf{x}, y)}L(y, f(\mathbf{x}|\theta))] \tag{10}$$

$$R(\theta, P_t(\mathbf{x}, y)) = E_{\sim P_s}[\beta(\mathbf{x}, y)L(y, f(\mathbf{x}|\theta))] \tag{11}$$

where $\beta(\mathbf{x}, y) = \frac{P_t(\mathbf{x},y)}{P_s(\mathbf{x},y)}$, provided the support of $P_t(\mathbf{x}, y)$ is contained in the support of $P_s(\mathbf{x}, y)$. Under the covariate shift assumption, we can write the following equation

$$\frac{P_t(\mathbf{x}, y)}{P_s(\mathbf{x}, y)} = \frac{P_t(\mathbf{x})}{P_s(\mathbf{x})}\frac{P_t(y|\mathbf{x})}{P_s(y|\mathbf{x})} \tag{12}$$

$$\frac{P_t(\mathbf{x}, y)}{P_s(\mathbf{x}, y)} = \frac{P_t(\mathbf{x})}{P_s(\mathbf{x})} = \beta(\mathbf{x}, y) \tag{13}$$

We denote the value of $\beta(\mathbf{x_i}, y_i)$ as β_i. Thus by obtaining the β_i value for the source instance $(\mathbf{x_i}, y_i) \in T_s = \{(\mathbf{x_i}, y_i)\}_{i=1}^{p}$, we can minimize the risk, as shown in equation 11.

Kernel Mean Matching minimizes the maximum mean discrepancy (MMD) between two distributions by projecting them into a reproducing Hilbert kernel space (RKHS). MMD is formulated as the difference of the means of source and target when projected onto the new space:

$$\text{MMD}(T_s, T_t) = \| \frac{1}{p}\sum_{i=1}^{p}\phi(\mathbf{x_i^s}) - \frac{1}{q}\sum_{i=1}^{q}\phi(\mathbf{x_i^t}) \|_H \tag{14}$$

where $\phi(\mathbf{x_i^s})$ and $\phi(\mathbf{x_i^t})$ are projections of the p source instances $T_s = \{\mathbf{x_i}\}_{i=1}^{p}$ and q target instances $T_t = \{\mathbf{x_i}\}_{i=1}^{q}$ respectively. Thus the minimization problem is as follows:

$$\arg\min_{\beta} \| \frac{1}{p}\sum_{i=1}^{p}\beta(\mathbf{x_i^s})\phi(\mathbf{x_i^s}) - \frac{1}{q}\sum_{i-1}^{q}\phi(\mathbf{x_i^t}) \|_H \tag{15}$$

subject to $\beta(\mathbf{x_i^s}) > 0$ and $E_{x\sim P_s(x)}[\beta(\mathbf{x})] = 1$, where $\beta(\mathbf{x_i^s}) = \beta(\mathbf{x_i}, y_i)$

Thus, if $\beta(\mathbf{x_i^s}) \in [0, B]$ is a fixed function in $x \in \mathcal{X}$ having finite mean and non-zero variance, the weights $\beta(\mathbf{x_i^s})$ can be estimated using the following equation:

$$\arg\min_{\beta} \parallel \frac{1}{p}\sum_{i=1}^{p}\beta(\mathbf{x_i^s})\phi(\mathbf{x_i^s}) - \frac{1}{q}\sum_{i-1}^{q}\phi(\mathbf{x_i^t}) \parallel_H \tag{16}$$

such that

$$\beta(\mathbf{x_i^s}) \in [0, B] \text{ and } |\frac{1}{p}\sum_{i=1}^{p}\beta(\mathbf{x_i^s}) - 1| < \epsilon$$

where ϵ is a small quantity. The condition $\beta(\mathbf{x_i^s}) \in [0, B]$ minimizes the discrepancy between $P_s(\mathbf{x}, y)$ and $P_t(\mathbf{x}, y)$. The condition $|\frac{1}{p}\sum_{i=1}^{p}\beta(\mathbf{x_i^s}) - 1| < \epsilon$ ensures that the measure $\beta(\mathbf{x})P_s(\mathbf{x}, y)$ is a probability distribution. A good choice of ϵ is $O(B/\sqrt{p})$.

The $\beta(\mathbf{x_i^s})$ values act as the weights $\beta(\mathbf{x_i}, y_i)$ [36] for the source examples. The error of each source example is multiplied by the corresponding $\beta(\mathbf{x_i^s})$ value while calculating the risk of the training model, thus giving appropriate weights to the source examples.

2) Subspace Alignment: This is a common domain adaptation method based on feature subset selection [37]. The idea is to apply PCA on source T_s and target T_t datasets separately by choosing a common space with d dimensions. It then attempts to align the projected source dataset with the projected target dataset in this common subspace using a *subspace alignment matrix*. Once source and target are aligned, a classifier is built on the transformed source dataset T_s^α and subsequently applied to the transformed target dataset T_t^α. Algorithm 2 outlines these ideas. In step 3, $(T_s^*)^{'}T_t^*$ corresponds to the subspace alignment matrix; $(T_s^*)^{'}$ is the transpose of T_s^*. The output is a model M that can be applied to the transformed target dataset T_t^α.

Algorithm 2 Subspace Alignment Algorithm

Input : Source Data T_s (labeled), Target Data T_t (unlabeled), Dimensionality d

Output : Classifier M_s and transformed target data T_t^α.

1: $T_s^* \leftarrow \text{PCA}(T_s, d)$ (source defined through d eigenvectors)
2: $T_t^* \leftarrow \text{PCA}(T_t, d)$ (target defined through d eigenvectors)
3: $Z = T_s^*(T_t^*)^{'}T_t^*$ (aligning source with target)
4: $T_s^\alpha = T_s^*Z$ (source data in aligned space)
5: $T_t^\alpha = T_eT_t^*$ (target data in aligned space)
6: Build model M_s from T_s^α.

C. Domain Adaptation with Active Learning

We now describe our approach to combine domain adaptation with active learning. As mentioned before, active learning selectively chooses (unlabeled) examples deemed relevant for classification. In our study, we begin by randomly choosing a set of instances from the target dataset and querying them (each class label incurs a fixed cost). A model is then built on this labeled target sample, followed by an iterative process that queries the next most informative instance from the target dataset, and builds a new model after the last queried instance is added to the sample. The algorithm stops when it reaches a maximum cost i.e., when it runs out of budget.

In this paper we use pool-based [38] active learning with margin sampling [39] as the uncertainty-sampling technique [40]. Here, the most informative instance to query, $\mathbf{x}^*$, is selected based on the concept of *margin* as a measure of class-label uncertainty. While many interpretations exist for the margin, here we define it as the difference between the first $P(y'|\mathbf{x})$ and second $P(y''|\mathbf{x})$ highest class posterior probabilities conditioned on $\mathbf{x}$. Instance $\mathbf{x}^*$ is then selected as follows:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} P(y'|\mathbf{x}) - P(y''|\mathbf{x}) \tag{17}$$

Our approach to combine domain adaptation with active learning is described in Algorithm 3. We use the model M_s created by the domain adaptation algorithm as the initial model for active learning. This step obviates randomly querying instances from the target dataset to create an initial labeled sample, and can be seen as a form of transfer of knowledge between source and target. The final goal is to build an accurate model M_t from the target domain.

The algorithm begins be selecting k instances from T_t with lowest margin (eq. 17) using model M_s, and queries their class labels (cost increases k units). Invoking the source model is important to avoid the known "cold start" problem [41] in active learning, where the efficacy of the method depends on an initial accurate model, but such model requires a representative labeled sample to be constructed in the first place. We circumvent the problem by relying on M_s at this initial stage. After that an iterative process simply looks for the next instance with lowest margin, queries the class label, and adds to the target sample (cost increases one unit). The cycle continues until cost reaches the maximum budget. The output is a new model M_t trained on the target labeled sample.

IV. Experiments

We show results after applying our approach to simulated data stemming from the *Supernova Photometric Classification Challenge* [5], traditionally called post-SNPCC. This is simulated data built to mimic the characteristics of Dark Energy Survey (DES) data. It is composed of a spectroscopic (source) and a photometric (target) sample, which resembles biases present in real spec/photo datasets (specifically, spectroscopic data are less numerous, brighter, closer, and less noisy than its photometric counterparts) and consequently stands as an ideal ground to test the combination of domain adaptation and active learning techniques.

We follow the same data treatment described in [6] for the construction of a sample: we select only objects having at least 3 observed epochs per filter, with at least 1 of them being before -3 days and at least 1 after +24 days since maximum brightness. In each filter, light curve fitting is performed using

Algorithm 3 Domain Adaptation with Active Learning

Input: Target Data T_t, Source Model M_s, Budget b, Initial Sample Size k.
Output: Model M_t built on a labeled target sample.
Initialize: T_t^l={}, Cost=0

Use M_s to select k instances from T_t (margin sampling).
Query class labels and add instances to T_t^l.
Cost = Cost + k.
Build model M_t from T_t^l .
T_t = T_t - T_t^l.
while cost $\leq b$ **do**
    Find x* from T_t^l with minimum margin.
    Query x* and add instance to T_t^l.
    Remove $\mathbf{x}^*$ from T_t.
    Build model M_t from T_t^l.
    Cost = Cost + 1.
Return model M_t.

Gaussian process regression [42], and the resulting function is sampled with a window of 1 day. No quality cuts are imposed (SNR>0)[5].

After the selection cuts, our surviving spectroscopic (source) sample embeds 719 SNe, while the photometric (target) sample embeds 4791 SNe. Both matrices hold 108 columns (27 epochs $\times$ 4 filters).

A. Dimensionality Reduction on Source Dataset

We first apply Kernel Principal Component Analysis (KPCA) as a form of dimensionality reduction, and compare results with Principal Component Analysis (PCA). We estimate classification accuracy using 10-fold cross-validation, repeating the whole experiment ten times for each learning algorithm; results are shown in Table I (numbers within parentheses refer to standard deviations).

The first column in Table I shows the learning algorithms. We experiment with four basic algorithms using default values in WEKA [43]: Neural Networks with a single hidden layer (comprising 11 hidden nodes and 30% training data as validation set); Support Vector Machines (SVMs) with Polynomial Kernel (degrees 1-3); The second column show classification accuracy with all (108) features. The third and fourth columns shows similar results but using PCA, and KPCA. Besides accuracy, we also estimate precision as an alternative performance metric[6]; results are shown in Table II.

From Table I we observe that classifiers built on the dataset reduced by KPCA tend to yield similar classification accuracy than PCA for most algorithms. The same can be said in general between KPCA and no dimensionality reduction. Table II shows similar results when precision is used as the performance metric. These results simply justify the use of a dimensionality reduction technique for our analysis. Since we observe a considerable decrease in the number of dimensions when using KPCA, we adopt this technique to transform our source and target datasets, T_s and T_t.

[5]We refer the reader to [6], section 4, for a detailed description of the data preparation process.

[6]Precision is defined as the number of true positives divided by the number of true positives and false positives. Precision is also known as "purity" in the astronomical jargon.

Table I
ACCURACY ON SOURCE DATASET. TESTING THE VALUE OF DIMENSIONALITY REDUCTION.

Learning Algorithm	Accuracy on Supernova Dataset		
	Original	PCA	KPCA
Neural Networks	94.16 (2.80)	93.37 (2.77)	93.76 (2.66)
SVM Polynomial 1	93.17 (2.94)	93.62 (2.54)	93.55 (2.53)
SVM Polynomial 2	96.02 (2.10)	94.85 (2.66)	95.32 (2.43)
SVM Polynomial 3	95.02 (2.17)	95.43 (2.49)	96.02 (2.37)

Table II
PRECISION ON SOURCE DATASET. TESTING THE VALUE OF DIMENSIONALITY REDUCTION.

Learning Algorithm	Purity on Supernova Dataset		
	Original	PCA	KPCA
Neural Networks	0.94 (0.04)	0.94 (0.04)	0.94 (0.03)
SVM Polynomial 1	0.93 (0.04)	0.93 (0.04)	0.93 (0.04)
SVM Polynomial 2	0.96 (0.03)	0.94 (0.04)	0.95 (0.03)
SVM Polynomial 3	0.95 (0.03)	0.95 (0.03)	0.96 (0.03)

B. Domain Adaptation with No Active Learning

We apply two domain adaptation methods (Section III-B): Kernel Mean Matching(KMM) and Subspace Alignment (SA) to our transformed datasets. Results are obtained using 10-fold cross validation and repeating the experiment ten times. Table III displays our results (numbers within parentheses refer to standard deviations). The first column refers to the learning algorithms. The second column shows accuracy when the classifiers are built on the source dataset and directly applied to the target dataset without any domain adaptation. The third and fourth columns show accuracy on the target dataset when classifiers are built on the source dataset using domain adaptation techniques: KMM with Gaussian Kernel (0.3 variance), and SA.

Similar results are shown in Table IV with precision as the performance metric. We plot results of Table III and Table IV in Figure 1 and Figure 2, respectively.

From Tables III and IV, and Figures 1 and 2, we observe that KMM yields better accuracy and precision than Subspace Alignment; this is also true when KMM is compared to the direct use of the source model on the target dataset (with the sole exception of SVM degree 3). We take this as evidence of the advantage of domain adaptation using KMM for our specific application domain.

Table III
ACCURACY ON TARGET DATASET. TESTING THE VALUE OF DOMAIN ADAPTATION METHODS.

Learning Algorithm	Accuracy on Supernova Dataset		
	No DA or AL	KMM	SA
Neural Networks	66.88 (0.29)	75.36 (3.18)	49.85 (1.00)
SVM Polynomial 1	68.34 (0.30)	70.05 (0.29)	53.15 (0.84)
SVM Polynomial 2	70.42 (0.47)	75.47 (2.09)	49.46 (0.47)
SVM Polynomial 3	70.39 (0.43)	66.90 (0.57)	48.89 (0.43)

Table IV
PRECISION ON TARGET DATASET. TESTING THE VALUE OF DOMAIN ADAPTATION METHODS.

Learning Algorithm	Purity on Supernova Dataset.		
	No DA or AL	KMM	SA
Neural Networks	0.37 (0.01)	0.47 (0.40)	0.23 (0.01)
SVM Polynomial 1	0.38 (0.01)	0.41 (0.01)	0.28 (0.01)
SVM Polynomial 2	0.41 (0.01)	0.48 (0.03)	0.23 (0.01)
SVM Polynomial 3	0.42 (0.01)	0.38 (0.01)	0.22 (0.01)

The poor performance of Subspace Alignment can be explained as the result of a preliminary step to unify source and target into the same attribute space, as is done by the eigenvector decomposition described in Section III-A; it is unlikely to see improvement using SA once both source and target are transformed to a common space.

One more observation is that best performance in terms of both accuracy and precision on the target dataset is reached with Neural Networks. Hence, we use this classifier to build the initial model during active learning.

C. Domain Adaptation With Active Learning

We now describe experiments that combine domain adaptation with active learning. To proceed, we divide the target dataset randomly into two equal parts - pool set and test set. Our algorithm queries target instances only from the pool set, while performance (accuracy and precision) is measured on the test set (not available during training). We form ten such pairs of pool and test sets and apply domain adaptation with active learning on each of these pairs (ten times). Results are shown in Table V. Values are average accuracy and average precision over 100 runs. Numbers within parentheses refer to standard deviations.

In Table V, the first column corresponds to the learning algorithms, while the second and third columns show results for accuracy and precision. The second row refers to the case where the source model is used directly on the target without invoking any domain adaptation technique. The third and fourth rows refer to the use of Subspace Alignment(SA) and Kernel Mean Matching (KMM), respectively. The fifth-eighth columns refer to the case where the model built using KMM

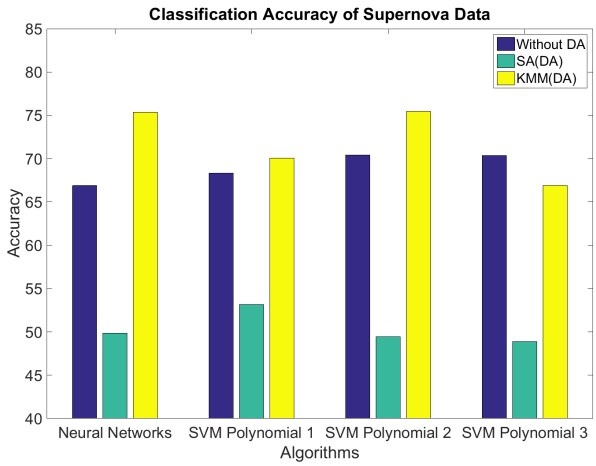

Figure 1. Accuracy on Supernova Target Data with and without Domain Adaptation

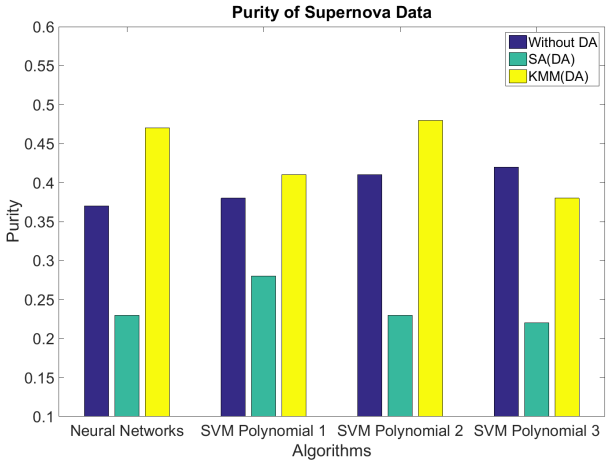

Figure 2. Precision on Supernova Target Data with and without Domain Adaptation

is used along with active learning with a budget of 50, 500, 1000 and 2000 labels respectively (the budget specifies the maximum number of queried instances). We plot our results in Figure 3.

Results show a significant advantage gained when domain adaptation is combined with active learning. Even with a modest budget size, the combination yields better performance than the use of domain adaptation alone. Of course, growing the budget size provides increased performance gain, but there is a trade-off between performance and the cost of querying class labels. As mentioned in Section I, the precise labeling of a supernova is normally done through spectroscopy, which is an expensive and time-consuming process. As a result, the practical application of our proposed methodology needs to consider how much we can stretch the budget (using spectroscopy) to facilitate the construction of accurate models (built mainly using photometry).

Table V
ACCURACY AND PRECISION ON TARGET DATASET. TESTING THE VALUE OF DOMAIN ADAPTATION AND ACTIVE LEARNING.

Learning Algorithm	Accuracy	Precision
Without DL or AL	66.88 (0.29)	0.37 (0.01)
SA (DA)	49.85 (1.00)	0.23 (0.01)
KMM (DA)	75.36 (3.18)	0.47 (0.40)
KMM with AL Budget: 50	80.27 (4.47)	0.86 (0.04)
KMM with AL Budget: 500	91.32 (1.94)	0.94 (0.01)
KMM with AL Budget: 1000	92.99 (0.52)	0.95 (0.01)
KMM with AL Budget: 2000	93.46 (0.43)	0.96 (0.01)

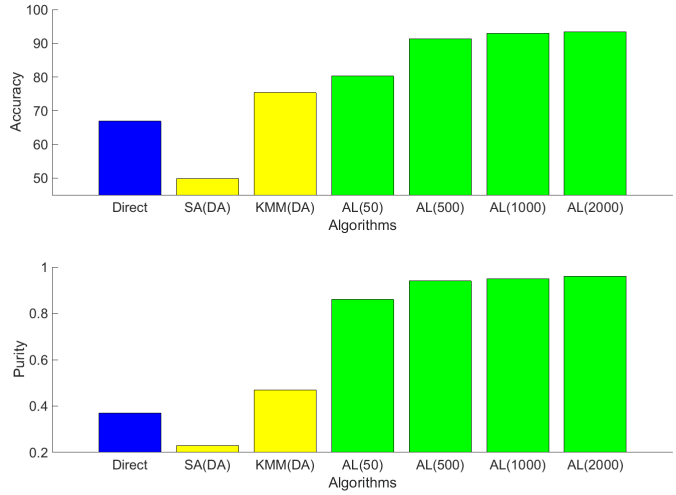

Figure 3. Accuracy and Precision on Target Data using Domain Adaptation and Active Learning.

V. SUMMARY AND CONCLUSIONS

In this paper we provide an efficient method to classify supernovae Ia, a problem of high priority in the astronomical community. The urgency in solving this problem comes from the need to analyze massive amounts of data gathered from recent (unlabeled) photometric observations, without ignoring the already existing (labeled) data obtained through the use of spectroscopy. Our methodology exploits both the available (small) set of spectroscopic data (source domain), and the (large) set of unlabeled photometric data (target domain). The proposed adaptive learning technique combines domain adaptation with active learning. Specifically, our proposed methodology exploits information from the source dataset to build an accurate model on the target dataset using Neural Networks as the learning algorithm, and Kernel Mean Matching as the domain adaptation technique. Best results are obtained when domain adaptation is combined with pool-based active learning with margin sampling.

The following conclusions can be drawn from our experimental results: 1) Kernel Principal Component Analysis is an efficient technique to reduce the dimensionality of the data; 2) Domain adaptation techniques can be used to improve performance beyond a model that is simply built on the source domain and tested on the target domain; 3) A combination of domain adaptation and active learning shows best results in terms of accuracy and precision. In this last approach, a large budget (i.e., large number of allowed queries) yields excellent performance results, but the practical deployment of such technique in a real scenario must take into account the feasibility of having a large budget, due to the inherent cost of class labeling (currently based on a costly spectroscopic analysis).

ACKNOWLEDGMENTS

This work was partly supported by the Center for Advanced Computing and Data Systems (CACDS), and by the Texas Institute for Measurement, Evaluation, and Statistics (TIMES) at the University of Houston.

REFERENCES

[1] S. Alam, F. D. Albareti, C. Allende Prieto, F. Anders, S. F. Anderson, T. Anderton, B. H. Andrews, E. Armengaud, É. Aubourg, S. Bailey, and et al., "The Eleventh and Twelfth Data Releases of the Sloan Digital Sky Survey: Final Data from SDSS-III," *The Astrophysical Journal Supplement Series*, vol. 219, p. 12, Jul. 2015.

[2] Dark Energy Survey Collaboration, T. Abbott, F. B. Abdalla, J. Aleksić, S. Allam, A. Amara, D. Bacon, E. Balbinot, and et al., "The Dark Energy Survey: more than dark energy - an overview," *Monthly Notices of the Royal Astronomical Society*, vol. 460, pp. 1270–1299, Aug. 2016.

[3] Z. Ivezic, J. A. Tyson, B. Abel, E. Acosta, R. Allsman, Y. AlSayyad, S. F. Anderson, J. Andrew, and et al., "LSST: from Science Drivers to Reference Design and Anticipated Data Products," *ArXiv e-prints*, May 2008.

[4] E. D. Feigelson and G. J. Babu, "Big data in astronomy," *Significance*, vol. 9, no. 4, pp. 22–25, 2012. [Online]. Available: http://dx.doi.org/10.1111/j.1740-9713.2012.00587.x

[5] R. Kessler, B. Bassett, P. Belov, V. Bhatnagar, H. Campbell, A. Conley, J. A. Frieman, A. Glazov, and et al., "Results from the Supernova Photometric Classification Challenge," *Publications of the Astronomical Society of Pacific*, vol. 122, pp. 1415–1431, Dec. 2010.

[6] E. E. O. Ishida and R. S. de Souza, "Kernel PCA for Type Ia supernovae photometric classification," *Monthly Notices of the Royal Astronomical Society*, vol. 430, pp. 509–532, Mar. 2013.

[7] R. Hlozek, M. Kunz, B. Bassett, M. Smith, J. Newling, M. Varughese, R. Kessler, J. P. Bernstein, and et al., "Photometric Supernova Cosmology with BEAMS and SDSS-II," *The Astrophysical Journal*, vol. 752, p. 79, Jun. 2012.

[8] M. Sako, B. Bassett, A. C. Becker, P. J. Brown, H. Campbell, R. Cane, D. Cinabro, C. B. D'Andrea, and et al., "The Data Release of the Sloan Digital Sky Survey-II Supernova Survey," *ArXiv e-prints*, Jan. 2014.

[9] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, pp. 151–175, 2010. [Online]. Available: http://www.springerlink.com/content/q6qk230685577n52/

[10] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 129–136. [Online]. Available: http://papers.nips.cc/paper/3212-learning-bounds-for-domain-adaptation.pdf

[11] H. Daumé, III and D. Marcu, "Domain adaptation for statistical classifiers," *J. Artif. Int. Res.*, vol. 26, no. 1, pp. 101–126, May 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622559.1622562

[12] Y. Mansour, M. Mohri, and A. Rostamizadeh, *Domain adaptation: Learning bounds and algorithms*, 2009.

[13] X. Shi, W. Fan, and J. Ren, *Actively Transfer Domain Knowledge*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 342–357.

[14] R. Vilalta, K. D. Gupta, and L. Macri, "Domain adaptation under data misalignment: An application to cepheid variable star classification," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, Aug 2014, pp. 3660–3665.

[15] K. Zhang, B. Schlkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, S. Dasgupta and D. Mcallester, Eds., vol. 28, no. 3. JMLR Workshop and Conference Proceedings, May 2013, pp. 819–827. [Online]. Available: http://jmlr.org/proceedings/papers/v28/zhang13d.pdf

[16] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[17] B. Liu, M. Huang, J. Sun, and X. Zhu, "Incorporating domain and sentiment supervision in representation learning for domain adaptation," in *Proceedings of the 24th International Conference on Artificial Intelligence*, ser. IJCAI'15. AAAI Press, 2015, pp. 1277–1283. [Online]. Available: http://dl.acm.org/citation.cfm?id=2832415.2832427

[18] J. Xu, S. Ramos, D. Vzquez, and A. M. Lpez, "Hierarchical adaptive structural svm for domain adaptation." *CoRR*, vol. abs/1408.5400, 2014. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1408.html#XuRVL14

[19] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. The MIT Press, 2009.

[20] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, Oct. 2000. [Online]. Available: http://www.sciencedirect.com/science/article/B6V0M-4136355-5/1/6432c256e0be03b1503bbf79e4e91d1a

[21] T. Kanamori, S. Hido, and M. Sugiyama, "A least-squares approach to direct importance estimation," *J. Mach. Learn. Res.*, vol. 10, pp. 1391–1445, Dec. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1577069.1755831

[22] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Advances in neural information processing systems*, 2008, pp. 1433–1440.

[23] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning under covariate shift," *J. Mach. Learn. Res.*, vol. 10, pp. 2137–2155, Dec. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1577069.1755858

[24] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 2006.

[25] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, Dec. 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1046920.1194905

[26] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 513–520.

[27] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation." in *NIPS*, B. Schlkopf, J. C. Platt, and T. Hofmann, Eds. MIT Press, 2006, pp. 137–144. [Online]. Available: http://dblp.uni-trier.de/db/conf/nips/nips2006.html#Ben-DavidBCP06

[28] A. Kumar, A. Saha, and H. Daume, "Co-regularization based semi-supervised domain adaptation," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 478–486. [Online]. Available: http://papers.nips.cc/paper/4009-co-regularization-based-semi-supervised-domain-adaptation.pdf

[29] G. Matasci, D. Tuia, and M. Kanevski, "Svm-based boosting of active learning strategies for efficient domain adaptation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 5, pp. 1335–1343, Oct 2012.

[30] C. Persello and L. Bruzzone, "Active learning for domain adaptation in the supervised classification of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 11, pp. 4468–4483, Nov 2012.

[31] P. Rai, A. Saha, H. Daumé, III, and S. Venkatasubramanian, "Domain adaptation meets active learning," in *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, ser. ALNLP '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 27–32. [Online]. Available: http://dl.acm.org/citation.cfm?id=1860625.1860629

[32] A. Saha, P. Rai, H. Daumé, S. Venkatasubramanian, and S. L. DuVall, "Active supervised domain adaptation," in *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III*, ser. ECML PKDD'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 97–112. [Online]. Available: http://dl.acm.org/citation.cfm?id=2034161.2034169

[33] X. Wang, T. kuo Huang, and J. Schneider, "Active transfer learning under model shift," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, T. Jebara and E. P. Xing, Eds. JMLR Workshop and Conference Proceedings, 2014, pp. 1305–1313. [Online]. Available: http://jmlr.org/proceedings/papers/v32/wangi14.pdf

[34] B. Schölkopf, A. Smola, and K.-R. Müller, *Kernel principal component analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 583–588. [Online]. Available: http://dx.doi.org/10.1007/BFb0020217

[35] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf, "Covariate shift by kernel mean matching," *Dataset shift in machine learning*, vol. 3, no. 4, p. 5, 2009.

[36] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Advances in neural information processing systems*, 2006, pp. 601–608.

[37] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Subspace alignment for domain adaptation," *CoRR*, vol. abs/1409.5241, 2014. [Online]. Available: http://arxiv.org/abs/1409.5241

[38] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., 1994, pp. 3–12.

[39] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *International Symposium on Intelligent Data Analysis*. Springer, 2001, pp. 309–318.

[40] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proceedings of the eleventh international conference on machine learning*, 1994, pp. 148–156.

[41] D. Kale and Y. Liu, "Accelerating active learning with transfer learning," in *2013 IEEE 13th International Conference on Data Mining*. IEEE, 2013, pp. 1085–1090.

[42] M. A. Chilenski, M. Greenwald, Y. Marzouk, N. T. Howard, A. E. White, J. E. Rice, and J. R. Walk, "Improved profile fitting and quantification of uncertainty in experimental measurements of impurity transport coefficients using gaussian process regression," *Nuclear Fusion*, vol. 55, no. 2, p. 023012, 2015. [Online]. Available: http://stacks.iop.org/0029-5515/55/i=2/a=023012

[43] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.