A General Approach to Domain Adaptation with Applications in Astronomy

Ricardo Vilalta*, Kinjal Dhar Gupta, Dainis Boumber
Department of Computer Science
University of Houston
Houston TX, 77204-3010, USA
*corresponding author email: rvilalta@uh.edu

Mikhail M. Meskhi
Department of Computer Science
North American University
Stafford TX, 77477, USA

***Abstract*—The ability to build a model on a source task and subsequently adapt such model on a new target task is a pervasive need in many astronomical applications. The problem is generally known as *transfer learning* in machine learning, where *domain adaptation* is a popular scenario. An example is to build a predictive model on spectroscopic data to identify Supernovae IA, while subsequently trying to adapt such model on photometric data. In this paper we propose a new general approach to domain adaptation that does not rely on the proximity of source and target distributions. Instead we simply assume a strong similarity in model complexity across domains, and use active learning to mitigate the dependency on source examples. Our work leads to a new formulation for the likelihood as a function of empirical error using a theoretical learning bound; the result is a novel mapping from generalization error to a likelihood estimation. Results using two real astronomical problems, Supernova Ia classification and identification of Mars landforms, show two main advantages with our approach: increased accuracy performance and substantial savings in computational cost.**

***Index Terms*—Supervised Learning, Domain Adaptation, Maximum A Posteriori, Model Complexity.**

I. Introduction

In this paper we propose a new approach to domain adaptation that is particularly well suited for astronomical applications. Our general setting assumes the learner is embedded in a *domain adaptation* framework [1]–[8], where the goal is to obtain a predictive model on a target domain, where examples abound, but labeled data is scarce. We assume the existence of a source domain with abundant labeled data, but with different distribution, such that the naive approach of directly applying the source model on the target becomes inadequate. Instead we follow a Maximum A Posteriori (MAP) approach to estimate model complexity by extracting the prior distribution from previous experience (i.e., from a previous task), and by taking the (scarce) target data as evidence to compute the likelihood. The result is a new approach to domain adaptation that is exempt from the common restriction that demands close proximity between source and target distributions [2].

We show how using a prior distribution from a previous task to estimate a posterior of model complexity on a new task, not only yields an increase in accuracy performance, but in addition has an enormous impact on computational cost. Our focus is on astronomical problems where we are witnessing a rapid growth of data volumes corresponding to a variety of astronomical surveys; data repositories have gone from gigabytes into terabytes, and we expect those repositories to reach the petabytes in the coming years [9], [10]. Our proposed methodology assumes an exhaustive search for the right model complexity on a source domain, where we generate a prior distribution on model complexity. The arrival of a new target task dispels with such exhaustive search; instead it generates a posterior distribution that directly leads to finding a near-optimal figure of model complexity. This is particularly important for big-data applications where lengthy computational tasks are unavoidable, even under the availability of an efficient high-performance-computing infrastructure.

We report on experiments using two real-world astronomical domains: classification of Supernovae Ia using photometric data, and characterization of landforms on Mars using Digital Elevation Maps (DEMs). Both domains can produce massive amounts of data with a strong need for efficient computational solutions. Results show how the use of a source prior to guide the search for an optimal value of model complexity can significantly improve on generalization performance.

The rationale for assuming similar model-complexity values across tasks is based on the nature of distributional discrepancies in many physical domains. The idea is useful not only to astronomical data analysis, but to many other real-world problems where the shift in distribution originates from more sophisticated equipment (e.g., modern telescopes), different instrumentation, or different coverage on the feature space, while the complexity of the classification problem experiences little change. For example, while spectroscopic and photometric observations capture data at different levels of resolution, the identification itself of specific astronomical objects shares a similar degree of difficulty. In short, we assume that the change in distribution from source to target does not affect model complexity significantly.

This paper is organized as follows. We begin by providing basic concepts in classification and domain adaptation, followed by a detailed description of our proposed approach that shows how to extract a prior distribution from a source domain. We then show our experiments and empirical results. The last section provides summary and conclusions.

II. Preliminary Concepts

A. Basic Notation

In supervised learning or classification, we assume the existence of a training set of examples, $T = \{(\mathbf{x_i}, y_i)\}_{i=1}^{p}$, where vector $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ is an instance of the input space $\mathcal{X}$, and y is an instance of the output space $\mathcal{Y}$. It is often assumed that sample T contains independently and identically distributed (i.i.d.) examples that come from a fixed but unknown joint probability distribution, $P(\mathbf{x}, y)$, in the input-output space $\mathcal{X} \times \mathcal{Y}$. The output of the learning algorithm is a 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 comes from a space of functions $\mathcal{H}$. The idea is to search for the hypothesis that minimizes the expectation of a loss function $L(y, f(\mathbf{x}|\theta))$, a.k.a. the risk:

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

where we usually employ the zero-one loss function:

$$L(y, f(\mathbf{x}|\theta)) = 1_{\{\mathbf{x}|\mathbf{y}(\mathbf{x}) \neq f(\mathbf{x}|\theta)\}}(\mathbf{x}) \tag{2}$$

such that $1(\cdot)$ is an indicator function, and $\mathbf{y}(\mathbf{x})$ is the true class of $\mathbf{x}$.

Domain Adaptation. In domain adaptation, we assume the existence of a source domain, corresponding to a previous task from which experience can be leveraged, and a target domain, corresponding to the present task. Each domain enables us to draw a dataset: $T_s = \{(\mathbf{x_i}, y_i)\}_{i=1}^{p}$ for the source, and $T_t = \{\mathbf{x_i}\}_{i=1}^{q}$ for the target. T_s is an instantiation of a joint probability distribution, $P_s(\mathbf{x}, y)$, while T_t is an instantiation of the marginal distribution $P_t(\mathbf{x})$ (from the joint distribution $P_t(\mathbf{x}, y)$, such that $P_t(\mathbf{x}) = \int_y P_t(\mathbf{x}, y) d_y$). The emphasis is always placed on the target domain, corresponding to the task at hand. The main objective is to induce a model from the target dataset; when building the model, one can exploit knowledge from the source dataset. A major difficulty in domain adaptation stems from the lack of labels on T_t. We will assume the possibility of *querying* some of those examples to attain a few labeled examples as part of an active learning setting [11].

Most domain adaptation methods assume similar class posteriors across source and target domains, i.e., $P_\mathrm{s}(y|\mathbf{x}) = P_\mathrm{t}(y|\mathbf{x})$, but different marginals $P_\mathrm{s}(\mathbf{x}) \neq P_\mathrm{t}(\mathbf{x})$; this is known as the covariate-shift assumption. Different from previous work, we will consider the case where both source and target differ in the marginal distributions and class posteriors.

Parameter Estimation. We also consider the problem of parameter estimation, which can play a major role in classification as a means to estimate an optimal figure of model complexity. Examples include finding the number of hidden nodes in a neural network, or finding the degree of a polynomial kernel in support vector machines. In Maximum a Posteriori (MAP), the goal is to obtain a point estimate that maximizes the posterior distribution of the parameter given the data or evidence. The posterior probability is essentially a function of two main factors: the prior probability (i.e., degree of belief of model complexity before data analysis) and the likelihood (i.e., probability of data sample conditioned on model complexity). When data abounds, the likelihood bears a stronger influence on the posterior, while the opposite takes place when data is scarce; here the prior bears more influence on the posterior. An important question is how to obtain a reliable prior when data is scarce (i.e., when the prior plays a strong role in estimating the posterior).

B. Related Work

Domain adaptation induces a model by exploiting experience gathered from previous tasks [2]. It is considered a subfield of transfer learning [12], and has become increasingly popular in recent years due to the pervasive nature of task domains exhibiting differences in sample distribution [13], [14]. The central question is if a previously constructed (source) model can be adapted to a new task, or if it is better to build a new (target) model from scratch.

Domain adaptation papers can be classified into two types: instance-based and feature-based methods. The idea in instance-based methods is to assign high weights to source examples occupying regions of high density in the target domain. A popular approach is known as covariate shift [15]–[19]. The covariance-shift assumption is that one can build a model on the newly-weighted source sample and apply it directly to the target domain [20], [21]. A stringent requirement is that source and target distributions must be close to each other.

Feature-based domain adaptation methods attempt to project source and target datasets into a latent feature space, where the covariate-shift assumption holds. A model is then built on the transformed space, and used as the classifier on the target. Examples are structural corresponding learning [3], subspace alignment methods [22], among others [8], [23], [24].

From a theoretical view, previous work has tried to estimate the distance between source and target distributions [1], [2], [4]; and employ regularization terms to find models with good generalization performance on both source and target domains [25].

III. Methodology

We begin by providing a general description of the proposed methodology. The main idea is to assist in finding the right configuration (model complexity) for a learning algorithm by leveraging information from a previous similar task. For example, when trying to find a predictive model to classify Supernovae, or to predict the class of a transient star, searching for a model with the right degree of complexity by varying a configuration parameter(s) may turn frustratingly cumbersome. As an example, setting the architecture of a (shallow) neural network by varying the number of hidden nodes would lead to a huge number of experiments to assess model quality for each architecture. To alleviate this situation, we *learn* a range of values of model complexity from a previous task using a Maximum a Posteriori approach, where there is a high

likelihood of finding a good value of model complexity (e.g., number of hidden nodes) on the new task. Moreover, different from previous work, our method disregards many assumptions made by previous work: we do not follow the covariate-shift assumption; no data projection is required to transform the feature space (thus incurring no loss of information); and the dependence on the source is not based on transferring source examples to build the target model. To summarize, the main idea is to learn about the model-building process employed in a previous task (source domain), and to transfer that experience to the new task (target domain). Experience is here understood as a distribution of optimal values of model complexity.

A. Active Learning

Our basic strategy is to step aside from the common approach followed by many domain-adaptation techniques that selectively gather source examples to enlarge the set of target examples. When source and target distributions differ significantly, such approach can lead to a biased model. Under high distribution discrepancy, an optimal strategy would simply rely on target instances. But the classical setting in domain adaptation provides none (or very few) class labels on the target dataset. A solution to this conundrum is to provide target class labels using active learning [11], [26], [27], where a selective mechanism queries an expert for (target) class labels under a limited budget (i.e., a limited number of queries).

The use of active learning in domain adaptation obviates using source examples while building the target model, opening new research avenues in the field of transfer learning. Here we investigate a mechanism that generates a distribution of model complexity on the source domain, and re-utilizes such distribution as a prior in a Bayesian setting over the target domain. The resulting point-estimate over the posterior distribution of model complexity depends on the prior (source domain) and the likelihood, or evidence (target domain).

B. Model Complexity as a Transferable Item

We assume that optimal predictive models for both source and target domains share a similar degree of model complexity. For example, assuming both domains are best modeled using Support Vector Machines with a polynomial kernel parameterized by θ (corresponding to the degree of a polynomial), we can then further assume that $P_s(\theta^*) \sim P_t(\theta^*)$, where θ^* is the polynomial degree that minimizes a loss function. Such assumption focuses on the similarity of complexity-parameter distributions, and not on the similarity of joint input-output distributions.

Specifically, iteratively sampling and building predictive models on the source domain leads to a distribution of model parameters, $P_s(\theta)$. Our goal is then to estimate an optimal parameter value θ^* that maximizes the posterior distribution on the target domain $P_t(\theta|D)$, where D is the data or evidence (i.e., target sample T_t). By using the distribution gathered from the source domain as a reliable prior, we can formulate the posterior using Bayes formula:

$$P_t(\theta|D) = \frac{P_t(D|\theta)P_s(\theta)}{\sum_i P_t(D|\theta_i)P_s(\theta_i)} \tag{3}$$

where $P_t(D|\theta)$ is the likelihood, and $P_s(\theta)$ the prior. This is precisely how we propose to adapt a model across domains; assuming the complexity of the model built on the source domain is similar to that on the target domain, we look at the source prior $P_s(\theta)$ as the *transferable item* to be used in the new target domain.

Since the denominator in Eq. (3) is constant, we can simplify the equation as follows:

$$P_t(\theta|D) = ZP_t(D|\theta)P_s(\theta) \propto P_t(D|\theta)P_s(\theta) \tag{4}$$

where Z is a normalization factor. To optimize $P_t(\theta|D)$, we optimize the product of $P_t(D|\theta)$ and $P_s(\theta)$, and disregard the value of Z, as it is not a function of parameter θ. Hence, our goal is not to obtain a distribution for the posterior $P_t(\theta|D)$, but only to estimate the value of θ that maximizes the product of the likelihood and prior[1], a technique known as Maximum a Posteriori, a.k.a. MAP. We now explain how to compute the prior $P_s(\theta)$ and likelihood $P_t(D|\theta)$ to obtain a point estimate of model complexity on the target domain.

C. Estimating the Likelihood

Our approach to estimate the likelihood is as follows. We first use active learning to obtain a (small) labeled sample from the target domain. We then introduce a novel mechanism to compute $P_t(D|\theta)$ by mapping generalization error to a likelihood probability. We explain both steps next.

1) Active Learning: To lessen the dependence on the source domain, we resort to active learning to produce an informative sample of labeled instances from the target domain. We use pool-based active learning with margin sampling [31] as the uncertainty sampling technique [32], [33]. Specifically, the algorithm randomly selects an initial set of instances from the unlabeled target dataset T_t, and queries their class labels; it then iteratively builds a model $f_t(\mathbf{x}|\theta)$ on the labeled target instances as follows. At every iteration, the algorithm identifies the instance $\mathbf{x}_i$ from the remaining unlabeled target instances with the minimum margin (i.e., minimum distance to the decision boundary), queries $\mathbf{x}_i$ to obtain class label y_i, and adds $(\mathbf{x}_i, y_i)$ to the set of labeled target instances. The process repeats until a budget (i.e., maximum number of allowed queries) is exhausted. The result is a labeled sample that will be used to compute the likelihood $P(D|\theta)$.

2) Mapping Error to a Likelihood: In general, the likelihood $P(D|\theta)$ estimates the probability of seeing data D given parameter θ. This estimation is particularly complex in our study because θ is normally understood as a parameter of a probabilistic or generative model. $P(D|\theta)$ indicates how likely it is to obtain D from a probabilistic model parameterized by θ. In our case θ is unconventionally defined as a (complexity)

[1]This is different from Bayesian estimation where the output is a full posterior probability distribution over model parameters [28]–[30].

parameter of a predictive model $f(\mathbf{x}|\theta)$ (e.g., degree of a polynomial kernel).

We contend that $P(D|\theta)$ can be re-interpreted as *the probability of D given the empirical error of $f(\boldsymbol{x}|\theta)$* on D. In general, the lower the empirical error, the higher the likelihood that model $f(\cdot)$ can reproduce the class labels contained in D.

Our definition of empirical error on D refers to the error incurred on sample D alone, and is denoted as a function of θ and D, $\widehat{g}(\theta|D)$. Empirical error is also known as in-sample error.

Our re-interpretation of the likelihood leads naturally to the assumption that the probability of D being classified correctly by a hypothesis $f(\mathbf{x}|\theta)$ is inversely proportional to the error made by $f(\cdot)$ on D. We formulate this inverse relation assuming an exponential distribution:

$$P(D|\theta) = \lambda \exp(-\lambda \widehat{g}(\theta|D)) \tag{5}$$

where λ is the rate parameter. This formulation simply states that the likelihood $P(D|\theta)$ decreases exponentially with error $\widehat{g}(\theta|D)$, but it is clearly handicapped, as different values of complexity θ are mapped to the same likelihood as long as the empirical error $\widehat{g}(\theta|D)$ is identical. However, under equal error rates, we would like to assign a higher likelihood to simpler models. We propose a solution to this next.

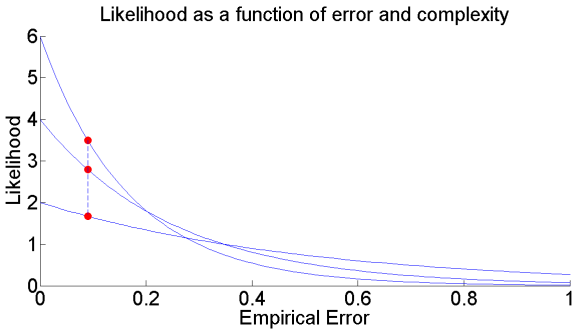

Fig. 1. Likelihood of the data given 1) empirical error $\widehat{g}(\theta|D)$ and 2) a scaled version of error variance as a function of the VC-dimension $d_{\mathrm{VC}}(\mathcal{H})$. Equal values for $\widehat{g}(\theta|D)$ do not map into the same likelihood.

3) Adding Robustness to the New Likelihood: Our formulation of the likelihood as a function of empirical error (Eq. (5)) can be made more robust by taking into account the variance component of error induced by models that belong to families exhibiting high VC-dimension (Vapnik-Chervonenkis dimension [34]). In short, we suggest to penalize those scenarios where VC-dimension is high. To start, we define $g(\theta|D)$ as the expected error across the whole input-output distribution:

$$g(\theta|D) = \int_{\mathcal{X}} \int_{\mathcal{Y}} L(y, f(\mathbf{x}|\theta)) P(\mathbf{x}, y) dx dy \tag{6}$$

where the loss $L(y, f(\mathbf{x}|\theta)$ is the zero-one loss function. $g(\theta|D)$ is also known as generalization error. Now, we know from Vapnik-Chervonenkis inequality [35] that with probability $1-\delta$, an upper bound on $g(\theta|D)$ is given as follows:

$$g(\theta|D) \leq \widehat{g}(\theta|D) + \sqrt{\frac{8}{N} \ln \frac{4 m_H(2N)}{\delta}} \tag{7}$$

where δ is user defined, N is the number of training instances, and $m_{\mathcal{H}}(q)$ is a polynomial function that defines that largest number of dichotomies on q training instances given the class of hypotheses $\mathcal{H}$:

$$m_H(q) \leq \sum_{i=0}^{d_{\mathrm{VC}}(H)} \frac{N!}{i!N-i!} \tag{8}$$

where $d_{\mathrm{VC}}(H)$ is the VC-dimension [34], defined as the maximum number of examples that can be shattered by $\mathcal{H}$ [35], and depends on the complexity of the hypothesis (i.e., on θ). The VC-dimension of various classes of hypotheses is well known. For example, the VC-dimension $d_{\mathrm{VC}}(\mathcal{H})$ of neural networks with a sigmoid gate function has a lower bound of $\sigma(w \log w)$ and an upper bound of $O(w^2)$ [36], where w is the number of weights in the network. In this example, parameter θ can be interpreted as the number of hidden nodes h in a feed-forward neural network (NN). The $d_{\mathrm{VC}}(H)$ of a neural network (NN) with h hidden nodes can then be estimated using the lower bound $w \log w$ and defining w=(i +1)$\times$ h + (h + 1)$\times$ o, where i and o are the number of input features and output classes respectively.

We now show how to strengthen our definition of the likelihood. In essence, we keep the exponential distribution the same (Eq. (5)), but make parameter λ a function of model complexity θ, $\lambda(\theta)$:

$$P_t(D|\theta) = \lambda(\theta) \exp(-\lambda(\theta) \widehat{g}(\theta|D)) \tag{9}$$

and define function $\lambda(\theta)$ as a scaled version of the second part of the Vapnik-Chervonenkis inequality (Eq. (7)):

$$\lambda(\theta) = \alpha \sqrt{\frac{8}{N} \ln \frac{4 m_\theta(2N)}{\delta}} \tag{10}$$

where α is a user-defined scale factor that decides how much weight is placed on the variance component of error. By transforming parameter λ into a function parameterized by θ, $\lambda(\theta)$, we achieve our goal of assigning higher likelihoods to simpler models when comparing hypotheses showing similar empirical error. We illustrate these concepts in Figure 1.

The ideas mentioned above have been tried using different strategies. Examples include a full Bayesian approach in transfer learning that finds a common subspace across tasks using a kernel-based dimensionality-reduction technique [37]; transferring priors across multiple tasks using a Hierarchical Bayesian approach [38]; finding clusters of tasks under a Dirichlet process prior [39]; finding a Gaussian prior from previous tasks [40]; and theoretical studies using the PAC learning framework on a Bayesian setting [41]. All these

methods are contingent on the proximity of source and target distributions, whereas our approach relies primarily on the similarity of model complexity.

Embedding the empirical error in an exponential function to compute the likelihood has been tried before [42], albeit without considering the capacity of each hypothesis. The novelty of our approach lies in transforming the likelihood function according to the VC-dimension of $\mathcal{H}$.

D. Estimating The Prior

Regarding the prior distribution of θ (model complexity) on the source domain, we adopt a parametric model assuming a univariate Gaussian distribution,

$$P_s(\theta) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{\theta-\mu}{\sigma})^2}$$

where μ and σ^2 are the mean and variance respectively. Specifically, our methodology generates k samples of the source dataset T_s using uniform random sampling without replacement; k is user-defined, and can be regarded as an experimental design parameter.

We construct classifiers on each of the k samples using a range of model complexity values, $\theta \in \{\theta_1, \theta_2, \theta_3, ..\theta_m\}$. For each sample S_i, we find a value θ_i^*, $1 \leq i \leq m$, that minimizes the expected loss (i.e., maximizes accuracy). The result is a sample with k optimal values of model complexity. Our estimate of the prior is finally obtained by fitting these values to the Gaussian model.

E. Estimating the Posterior

Once we have estimated the prior $P_s(\theta)$ and likelihood $P_t(D|\theta)$, we can estimate the numerator of the posterior distribution (Eq. (4)): $P_t(\theta|D) = ZP_t(D|\theta)P_s(\theta) \propto P_t(D|\theta)P_s(\theta)$. Since we are interested in obtaining a point estimate for the posterior, we look for an optimal value $\theta^* = \arg\max_\theta P_t(D|\theta)P_s(\theta)$.

To reduce the space of complexity values during optimization, we limit the values of θ to the range $[\mu-\sigma, \mu+\sigma]$, where μ and σ are the mean and standard deviation of the source prior distribution $P_s(\theta)$. The final value θ^* is used to build a classifier $f_t(\mathbf{x}|\theta^*)$ on the target domain using the queried instances as our training dataset.

Our methodology is outlined in Algorithm 1. The algorithm assumes as input the labeled source dataset T_s, the unlabeled target dataset T_t, the size r of a small labeled sample to generate a model on the target, and a budget b of possible queries to obtain additional labeled instances on the target. The first step is to build a prior distribution $P_s(\theta) \sim N(\mu, \sigma)$ on the source domain by exhaustively looking for an optimal figure of model complexity; this step can be computationally expensive, but can save substantial amounts of time when it is re-used on a future (target) task. The next steps compute the likelihood $P_t(D|\theta_i)$ in an iterative manner, by using labeled instances from the target obtained through active learning. The search is made narrow by limiting values of model complexity to just one standard deviation away from the source mean. The last steps build a (proportional) posterior distribution, and a predictive model using our optimal point estimate for model complexity.

Algorithm 1 Model Complexity Estimation Using Domain Adaptation and Active Learning

Input : Source Dataset T_s, Target Dataset T_t, Budget b, Initial Sample Size r.
Output : Predictive Target Model $f_t^*(\mathbf{x}|\theta)$

1: Estimate prior $P_s(\theta) \sim N(\mu, \sigma)$ using source dataset T_s
2: Set $\theta_{\min} = \mu - \sigma$ and $\theta_{\max} = \mu + \sigma$
3: Use the small set of r labeled instances from T_t to build model $f_t(\mathbf{x}|\theta)$
4: Use $f_t(\mathbf{x}|\theta)$ and active learning to label b target instances from T_t
5: **for** $\theta_i \leftarrow \theta_{\min}, \theta_{\max}$ **do**
6: Build model $f_t^i(\mathbf{x}|\theta)$ with θ_i as model complexity
7: Compute $\widehat{g}(\theta_i|D)$ and $\lambda(\theta_i)$ to estimate likelihood $P_t(D|\theta_i)$
8: Estimate (proportional) posterior: $P_t(D|\theta_i)P_s(\theta_i)$
9: **end for**
10: Let $\theta^* = \arg\max_{\theta_i} P_t(D|\theta_i)P_s(\theta_i)$
11: Build $f_t(\mathbf{x}|\theta^*)$
12: **return** $f_t(\mathbf{x}|\theta^*)$

IV. Experiments

We describe our experiments in detail next. All our code and datasets have been made available for reproducibility as a github project[2].

We report empirical results on two different scientific areas to validate our methodology. The first area refers to the automatic classification of supernovae using photometric light curves. The second area is centered on the classification of landforms on planet Mars using digital elevation maps.

A. Supernova Datasets

The automatic identification of Supernovae Ia (SNe Ia) has become a key step in many astronomical endeavors [43], [44]. Among different types of supernovae, SNe Ia are of particular relevance because they can be used as standard candles to probe large cosmological distances. The classification goal here is to identify SNe Ia (positive class) among other types (SNe Ib and Ic, negative class).

When analyzing light from a supernova, one can either exploit spectroscopic measurements, to take advantage of the wealth of information that can be obtained from spectral data. Such approach, however, is laborious and cumbersome. Another more common approach is to exploit photometric measurements; these are easier to obtain, but limited to a summarization of light intensity in bands or filters. The domain adaptation framework fits into this scenario as follows [45]: the source dataset corresponds to spectroscopic measurements where class labels (SNe Ia, Ib and Ic) are known with high

[2]Please visit https://github.com/PAL-UH/transferAL

confidence (but data is scarce); whereas the target domain corresponds to photometric measurements where class labels are missing (but data abounds).

Our experiments use simulations to generate samples that resemble the type of measurements expected when using spectroscopic or photometric measurements. This brings the advantage of having samples with the same set of features (i.e., same input space $\mathcal{X}$). Specifically, we use data from the Supernova Photometric Classification Challenge [46], consisting of supernova light curves simulated according to Dark Energy Survey specifications, using SNANA light curve simulator [47]. The data comes from simulations that approximate the characteristics of the Dark Energy Survey (DES). Simulations include both spectroscopic (source) and photometric (target) samples; these are created with biases found in true datasets, where spectroscopic data are in general smaller, brighter, closer, and less noisy than photometric data.

Regarding the construction of simulated samples, we follow the data processing steps specified in [47]. We only take objects with a minimum of three observed epochs per filter; at least one of them occurs before -3 days and at least one after +24 days since maximum brightness. On each filter, we use Gaussian process regression to do light-curve model fitting [48]; the resulting function is sampled using a window of size one day. No quality cuts are imposed (SNR>0). At the end, the spectroscopic (source) sample has 718 SNe, while the photometric (target) sample has 11946 SNe. Both samples have 108 columns or features (27 epochs $\times$ 4 filters). We use Kernel Principal Component Analysis (KPCA) to reduce the original dataset from 108 features to solely 20 features. This form of dimensionality reduction is useful to avoid the curse of dimensionality [49]. A preliminary analysis shows no loss of information or accuracy performance after data transformation using KPCA.

B. Mars Landforms

The second area corresponds to the automatic geomorphic mapping of planetary surfaces. Specifically, the goal is to label segments on specific regions on planet Mars with their corresponding landforms [50]–[52]. The input data comes in the form of raster data or digital elevation models (DEMs) produced by orbiting satellites (Mars Orbiter Laser Altimeter instrument on board the Mars Global Surveyor spacecraft). Each DEM is first subdivided into meaningful segments (groups of adjacent pixels with similar terrain properties) that are subsequently classified into the following landforms: crater floors, convex crater walls, concave crater walls, convex ridges, concave ridges, and inter-crater plateau. Domain adaptation is important to attain accurate predictive models on new target sites that exhibit different distribution from the original source site.

Figure 2 illustrates the sequence of steps needed to classify landforms on Mars using DEMs. The original map (A) is first divided into small segments amenable to labeling (B). A model is then trained on a fraction of all segments and applied to the rest. Models of different complexity yield different classifications (C-E).

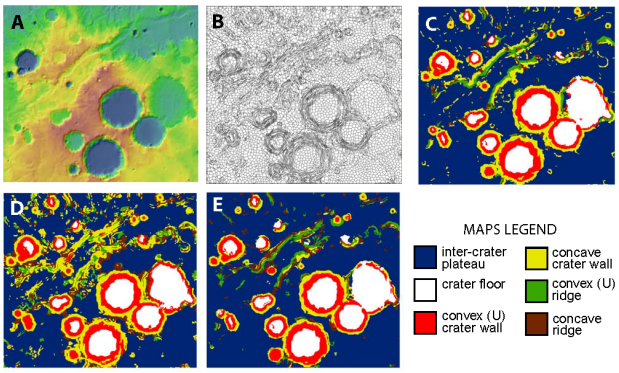

Fig. 2. DEMs on Mars are processed and classified into different landforms. The original DEM (A), corresponding to a site known as Tissia Valles, is segmented for labeling processes (red-to-blue gradient indicates high-to-low elevation). (B) The DEM is then classified using models of different complexity (C-E; color labels explained on bottom-right). This site, Tissia Valles, acts as the source domain.

The source domain is a region on Mars where we know the labels for all landforms; it is shown in Figure 2(A) and is known as Tissia Valles; it was chosen primarily because in a relatively small area, most landforms of interest are present. The region is heavily cratered and many different crater morphologies are present in a range of sizes. The goal here is to leverage experience during the model building process on Tissia Valles to find the right complexity for a model induced on a new site on Mars, corresponding to the target domain, where the landform distribution is different. In our experiments, the target site corresponds to a region known as Evos, shown in Figure 3. Notice the difference in distribution, where the shape of the craters and number of them differs significantly from Tissia Valles.

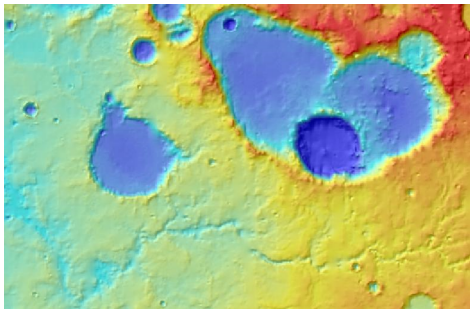

Fig. 3. A digital elevation map DEM of a region on Mars known as Evos, corresponding to our target domain (red-to-blue gradient indicates high-to-low elevation).

The input to the learning algorithm is not the DEM, but a training set made of feature vectors, one for each segment in the map. A segment is made of adjacent pixels with similar feature values. Each segments is characterized by three features, which are averages over the pixels embedded by the segment: slope, computed as the maximum rate of change

in elevation from a cell to its neighbors (indicative of the steepness of the terrain); curvature, computed as the second derivative of the surface elevation, useful to distinguish between convex, e.g., ridge, and concave, e.g., channel, surfaces; and flow, computed as the degree of flow accumulated on each cell (high values are indicative of stream or river channels).

C. Experimental Settings

To estimate the prior $P_s(\theta)$, we create $k = 100$ bootstrap samples of the source dataset T_s under uniform random sampling without replacement. We then record the θ^* that yields highest accuracy on each sample. In our experiments, θ corresponds to the number of hidden nodes in a neural network, $\theta \in [2, 50]$. For active learning, we divide dataset T_t randomly into two (equal) parts: a pool of target instances from which data can be queried T_t^1, and a pool of test instances that remains unknown during training T_t^2. We then randomly generate 10 pairs of training and testing pools. We first limit our active-learning budget to $b = 100$ queries, and subsequently study how accuracy varies with different budgets.

Regarding the posterior, we calculate the value of θ^* that maximizes the product of prior and likelihood. We then use θ^* to build an optimal model $f(\mathbf{x}|\theta^*)$ on the target pool T_t^1, and subsequently test it on the test pool T_t^2. In both domains, we have perfect knowledge of class labels (both source and target samples); class labels are hidden on the target sample to validate model performance.

Our hardware is made of a 3712-core computer cluster with 8 Tesla C2075 GPUs, and 22 GTK570 GPUs, 120TB Lustre Filesystem, and 127TB storage space.

D. Methods for Comparison

For comparison purposes, our experiments include other domain adaptation techniques. We list and describe such techniques next.

Subspace Alignment [22]. The goal is to find separate subspaces for source and target domains using Principal Component Analysis, followed by a linear transformation that maps the source domain into the target domain . The result is an alignment of both spaces through the basis vectors. The number of principal components is optimized based on a theoretical bound.

Joint Distribution Optimal Transportation (JDOT) for Domain Adaptation [53]. The technique assumes a map that aligns the joint distributions ($\mathcal{X} \times \mathcal{Y}$) of the source and target domains. The optimization function combines both the distance between samples and the discrepancy in the loss between class labels.

Adaptation Regularization based Transfer Learning (ARTL) [54]. The central idea is to combine different strategies for transfer learning within a single framework: it simultaneously optimizes the structural risk functional over the source domain, the joint distribution matching of both marginal and conditional distributions, and the consistency of the geometric manifold corresponding to the marginal distribution.

Transfer Joint Matching (TJM) [55]. The technique combines a shared representation between source and target domains with the concept of instance re-weighting, where source instances that fall within high density regions of the target domain, see their weight increased.

Transfer feature Learning with Joint Distribution Adaptation (JDA) [56]. The technique jointly adapts both marginal and class-conditional probabilities using Principal Component Analysis.

Domain-Adversarial Training of Neural Networks (DATNN) [57]. This is a neural network framework that implements domain adaptation by finding features that provide low error on the training set, while the features remain invariant across the two domains (i.e., across source and target domains). The architecture combines two learners that play in an adversarial manner: while one adjusts parameters to reduce training error, the other one adjusts parameters to discriminate (increase error) between source and target examples. The result is a regularized deep neural network that generates an informative abstract feature representation.

Geodesic Flow Kernel (GFK) for Unsupervised Domain Adaptation [58]. This technique integrates an infinite number of subspaces between source and target domains by paying attention to geometric and statistical properties of both domains. The technique focuses on those subspaces that are domain invariant.

E. Results

Prior. After estimating the sampling distribution for the optimal parameter θ^* on the source domain, our experiments show the following results. For Supernova: $\mu = 33.75$ and $\sigma = 9.35$, and for Mars landforms: $\mu = 23.19$ and $\sigma = 12.63$. The range of values for the prior are set to $\theta^* \in [24, 43]$ for Supernova and $\theta^* \in [10, 36]$ for Mars landforms.

As an illustration, Figures 4 and 5 show the histogram and corresponding univariate Gaussian approximation of optimal values of model complexity for the Supernova domain. Model complexity is measured in terms of the number of hidden nodes in a neural network. Approximating the prior distribution helps to narrow the search of values for the posterior, yielding substantial savings in computational cost (as shown in the following section).

Accuracy. Table I shows average accuracy comparing our approach with two blocks of techniques: one block labeled "Domain Adaptation" corresponding to the techniques described in Section IV-D (except for the first technique labeled "Source Model" that simply builds a model on the source domain and applies it directly to the target domain). The other block labeled "Source Model + Active Learning" contains results for methods using active learning, with the initial model built on the source dataset. Our technique (Bayesian DA or

TABLE I
CLASSIFICATION ACCURACY (NUMBERS ENCLOSED IN PARENTHESES REPRESENT STANDARD DEVIATIONS).

General Method	Learning technique	Datasets	
		Supernova Ia	Mars Landforms
Domain Adaptation	Source Model	69.13 (0.00)	74.36 (9.40)
	Subspace Alignment	62.56 (7.98)	85.16 (2.65)
	JDOT SVM	77.57 (0.13)	85.2 (0.59)
	JDOT NN	69.05 (0.08)	80.96 (0.06)
	DANN	80.4 (0.3)	88.61 (0.22)
	TJM	65.56 (0.01)	82.28 (0.03)
	JDA	70.64 (0.03)	80.40 (0.03)
	ARTL	66.21 (0.01)	88.12 (0.02)
	GFK	63.98 (0.02)	83.56 (0.04)
Source Model + Active Learning	NN + AL	85.75 (0.04)	80.41 (0.08)
	SVM + AL	69.33 (0.17)	85.90 (0.03)
	LR + AL	83.70 (0.03)	85.18 (0.02)
Bayesian DA	**NN-DA-AL**	**86.17 (0.35)**	**90.81 (1.49)**

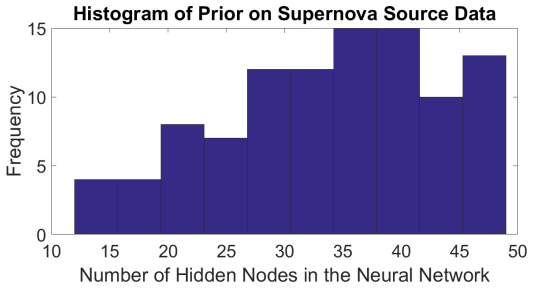

Fig. 4. Histogram of optimal values of model complexity for the Supernova domain.

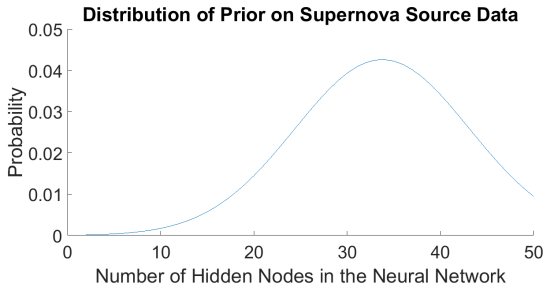

Fig. 5. Gaussian approximation to the (prior) distribution of optimal model complexity values for the Supernova domain.

NN-DA-AL) is shown as the last row of Table I. It combines domain adaptation with active learning using a neural network architecture (budget $b = 100$ queries and the initial labeled pool is of size $r = 10$).

For the first block, results show a significant increase in accuracy with our approach. For a statistical test, we use the Welch's Student Paired t-test distribution at the $p = 0.01$ level. We also perform a multiple-comparison test by adjusting the statistical test using a Bonferroni adjustment [59]; results are significant too at the $p = 0.01$ level after the adjustment; this is true on both astronomical problems. These results show that using a posterior distribution of model complexity yields better classification accuracy on the target dataset than using the best prior. Results also show a major limitation of domain adaptation techniques founded on the assumption of the existence of feature-invariant subspaces between source and target domains. Real-world applications either do not guarantee the existence of such subspaces, or exhibit a complex subspace landscape where finding common subspaces turns difficult. Under the general assumption where both marginal and posterior probabilities between source and target domains differ ($P_{\mathrm{s}}(\mathbf{x}) \neq P_{\mathrm{t}}(\mathbf{x})$ and $P_{\mathrm{s}}(y|\mathbf{x}) \neq P_{\mathrm{t}}(y|\mathbf{x})$), a better strategy is to sample directly from the target domain under a framework that limits the cost of class queries. If the posterior class probability on the target domain follows a smooth distribution, a limited number of queries should suffice to attain an accurate predictive model.

The second block shows average accuracy with techniques that use active learning, where the initial model is built on the source domain. We report on a neural network (NN) with logistic activation units and 25 hidden nodes, Support Vector Machines (SVM) with a radial basis function kernel, and Logistic Regression (LR); all with budget $b = 100$ and $r = 10$. We applied the same statistical test using Welch's Student Paired t-test distribution at the $p = 0.01$ and a Bonferroni adjustment. Our approach is significantly more accurate in all cases. This is true even with the use of a plain neural network, where there is no search for optimal complexity parameters (e.g., number of hidden units) and no source task to guide such search. Our method is also more efficient, as finding the best model-complexity from scratch on the target domain requires a new exhaustive search for an optimal value of model-complexity.

The next experiment tests the impact of active learning within our approach as the budget is increased. Figure 6 shows results for the Supernova task. Figure 7 shows results for the Mars landforms task. For the Supernova task, there is a significant increase in accuracy as the budget grows from 50 to $2,000$. This is expected as a large labeled sample on the target set provides enough evidence to generate accurate predictive models. Results tend to converge between $500 - 1000$ instances. For the Mars Landforms task, results tend to converge after only 100 instances. In practical real-world scenarios, such results can be used to set a trade-off between the size of

the budget and the cost of labeling new target instances. In the Supernova domain, for example, labeling new instances is extremely expensive as it involves running a full spectroscopic analysis; in such case, a lower budget may be preferred at the cost of some accuracy loss. The opposite is true on the Mars domain, where the cost of labeling segments with their correct landforms is relatively cheap, thus allowing for a budget increase.

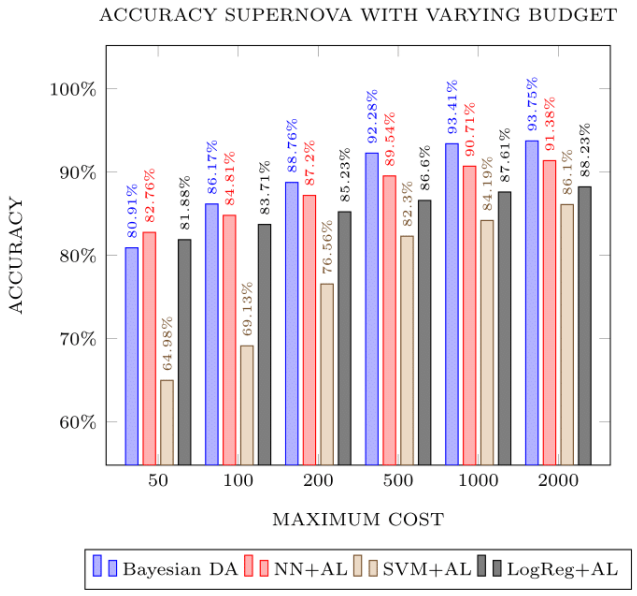

Fig. 6. Accuracy on Supernova improves significantly with increasing budget, and tends to converge after about 2,000 queries.

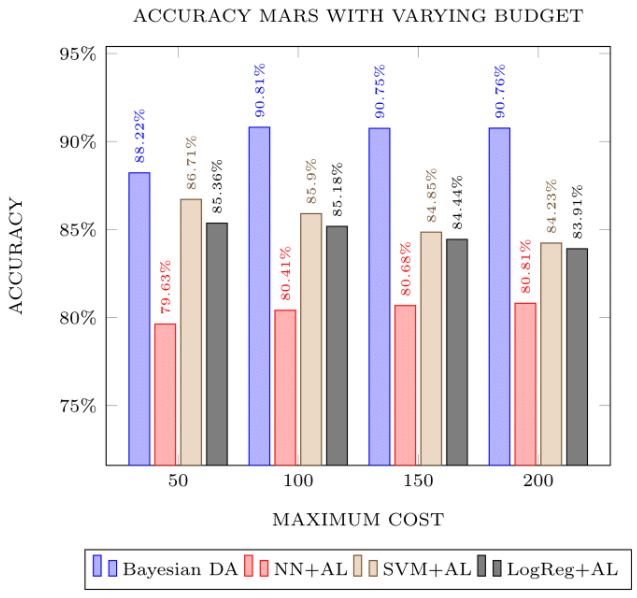

Fig. 7. Accuracy on Mars Landforms improves significantly with a budget of about 100 queries, and tends to converge after that threshold.

Execution Time. A final experiment assesses the benefits gained in computational complexity between our approach and the common approach that finds a common subspace to match source and target distributions. Experiments follow the assumption that domain adaptation generates a prior distribution in the past, and hence does not incur any additional time during the current target task; in addition the search for an optimal value of model complexity on the target is limited to one standard deviation around the optimal value found on the source prior. The model built on the source domain incurs on additional computational cost by searching for a common space over the two domains. We invoke Subspace Alignment as a representative case of feature matching techniques. For the Supernova task, execution time is reduced from about 90 hrs to under 2 hours. For the Mars landforms task, execution time goes from about 4 hrs to about 4 minutes. Results show the advantage of generating a prior distribution of model complexity on the source domain that is readily available on a new target task: it obviates an exhaustive search for an optimal parameter value.

V. Summary and Conclusions

We propose a new direction in domain-adaptation using a Maximum a Posteriori approach where the prior distribution is obtained from a source task (previous experience), whereas the likelihood is obtained from the target (or current) task. Our methodology invokes active learning to compensate for the lack of (target) class labels, leaving the budget size as an experimental parameter. Our study leads to a new formulation of the likelihood as a function of empirical error and a term that depends on model complexity as estimated by the Vapnik-Chervonenkis dimension. Overall, our technique broadens the general applicability of domain adaptation by relaxing the stringent requirement of close proximity between source and target distributions.

Empirical results on two astronomical problems show a significant advantage in computational cost as the range of complexity values on the target domain is limited to a small window; this is the result of using a prior distribution over the complexity parameter derived from the source domain. In terms of accuracy, results show a significant increase in performance with our approach; this holds for both astronomical domains. Our experiments also show a trade-off between budget size and the cost of labeling; in cases where labeling is relatively cheap, one can increase the budget to achieve an increase in accuracy performance.

As future work, we will investigate how to extend our work when multiple source domains are available. One possibility is to simply choose the best prior based on domain knowledge, or through a ranking system that orders all source tasks based on spatial or temporal proximity to the astronomical event of interest. Another direction is to combine all priors by assigning a degree of relevance to each source task. The posterior distribution can then be defined as a weighted combination of all available priors.

Additionally we hope to stimulate the astronomical community to consider domain adaptation as a useful resource when analyzing different surveys on similar objects. For example, while providing class labels for transient objects or events contained in one single survey is still feasible –even though costly– the ability of labeling variable sources across the large number of available surveys is almost non-existent. The goal of acquiring predictive models from many surveys is a daunting task. This can be tackled by creating predictive models that adapt across datasets under analysis using domain adaptation techniques. The need for domain adaptation lies in the distributional discrepancy between source and target domains.

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. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation." in *NIPS*, B. Schlkopf, J. Platt, and T. Hofmann, Eds. MIT Press, 2006, pp. 137–144.

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, and F. Pereira, "A theory of learning from different domains," *Machine Learning*, no. 79, pp. 151–175, 2010.

[3] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the 2006 conference on empirical methods in natural language processing, ACL*, 2006, pp. 120–128.

[4] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Advances in Neural Information Processing Systems, NIPS*, 2007, pp. 129–136.

[5] H. Daume and D. Marcu, "Domain adaptation for statistical classifiers," *Journal of Machine Learning Research*, no. 26, pp. 101–126, 2006.

[6] Y. Mansour, M. Mohri., and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," in *Proceedings of the 22nd Conference on Learning Theory, COLT*, 2009.

[7] D. Hal, "Frustratingly easy domain adaptation," *arXiv preprint arXiv:0907.1815*, 2009.

[8] L. Bruzzone and M. Marconcini, "Domain adaptation problems: a dasvm classification technique and a circular validation strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 770–787, 2010.

[9] M. Brescia and G. Longo, "Astroinformatics, data mining and the future of astronomical research," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 720, pp. 92 – 94, 2013.

[10] E. Feigelson, "The changing landscape of astrostatistics and astroinformatics," in *Astroinformatics: Proceedings of the International Astronomical Union, Symposium No. 325*, 2017.

[11] B. Settles, *Active Learning*. Morgan & Claypool, 2012.

[12] 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, 2010.

[13] 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.

[14] J. Xu, S. Ramos, D. Vzquez, and A. M. Lpez, "Hierarchical adaptive structural svm for domain adaptation." *CoRR*, vol. abs/1408.5400, 2014.

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

[16] 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.

[17] 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.

[18] 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.

[19] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning under covariate shift," *J. Mach. Learn. Res.*, vol. 10, pp. 2137–2155, Dec. 2009.

[20] 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, NIPS*, 2008, pp. 1433–1440.

[21] 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.

[22] F. Basura, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, 2013, pp. 2960–2967.

[23] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *Journal of Machine Learning Research*, vol. 6, pp. 1817–1853, 2005.

[24] 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)*, 2011, pp. 513–520.

[25] 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.

[26] M.-F. Balcan, A. Beygelzimer, and J. Langford, "Agnostic active learning," *Journal of Computer and System Sciences*, vol. 75, no. 1, pp. 78 – 89, 2009.

[27] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, no. 1, pp. 129–145, 1996.

[28] W. Bolstad, *Introduction to Bayesian Statistics*. Wiley-Interscience, 2nd Edition, 2007.

[29] S. Goodman, "Introduction to bayesian methods i: measuring the strength of evidence," *Clinical Trials*, vol. 2, pp. 281–290, 2005.

[30] T. Louis, "Introduction to bayesian methods ii: fundamental concepts," *Clinical Trials*, vol. 2, pp. 291–294, 2005.

[31] 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.

[32] 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*, 1994, pp. 3–12.

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

[34] D. Haussler, M. Kearns, and R. Schapire, "Bounds on the sample complexity of bayesian learning using information theory and the vc dimension," in *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, ser. COLT '91. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991, pp. 61–74.

[35] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data*. AMLBook Singapore, 2012, vol. 4.

[36] W. Maass, "Vapnik-chervonenkis dimension of neural nets," *The handbook of brain theory and neural networks*, pp. 1000–1003, 1995.

[37] M. Gönen and A. A. Margolin, "Kernelized bayesian transfer learning," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, ser. AAAI'14. AAAI Press, 2014, pp. 1831–1839.

[38] J. R. Finkel and C. D. Manning, "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*, ser. NAACL '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 602–610.

[39] D. M. Roy and L. P. Kaelbling, "Efficient bayesian task-level transfer learning," in *Proceedings of the 20th International Joint Conference on*

Artifical Intelligence, ser. IJCAI'07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, pp. 2599–2604.

[40] R. Raina, A. Y. Ng, and D. Koller, "Constructing informative priors using transfer learning," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 713–720.

[41] P. Germain, A. Habrard, F. Laviolette, and E. Morvant, "A new pac-bayesian perspective on domain adaptation," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, pp. 859–868.

[42] P. Germain, F. Bach, A. Lacoste, and S. Lacoste-Julien, "PAC-Bayesian Theory Meets Bayesian Inference," *ArXiv e-prints*, May 2016.

[43] S. Blondin, T. Matheson, R. P. Kirshner, K. S. Mandel, P. Berlind, M. Calkins, P. Challis, P. M. Garnavich, S. W. Jha, M. Modjaz, A. G. Riess, and B. P. Schmidt, "The spectroscopic diversity of type ia supernovae," *The Astronomical Journal*, vol. 143, no. 5, p. 126, 2012.

[44] M. Sasdelli, E. E. O. Ishida, R. Vilalta, M. Aguena, V. C. Busti, H. Camacho, A. M. M. Trindade, F. Gieseke, R. S. de Souza, Y. T. Fantaye, and P. A. Mazzali, "Exploring the spectroscopic diversity of Type Ia supernovae with DRACULA: a machine learning approach," *Monthly Notices of the Royal Astronomical Society*, vol. 461, pp. 2044–2059, Sep. 2016.

[45] D. G. K., P. R., V. R., I. E. E. O., and de Souza R. S., "Automated supernova ia classification using adaptive learning techniques," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, ser. CIDM '16. Morgan Kaufmann Publishers Inc., 2016, pp. 61–74.

[46] R. Kessler, A. Conley, S. Jha, and S. Kuhlmann, "Supernova photometric classification challenge," *arXiv:1001.5210*, 2010.

[47] E. Ishida and R. S. de Souza, "Kernel pca for type ia supernovae photometric classification," *Monthly Notices of the Royal Astronomical Society*, vol. 430, no. 1, pp. 509–532, 2013.

[48] M. Chilenski, M. Greenwald, Y. Marzouk, N. Howard, A. White, J. Rice, and J. 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, 2015. [Online]. Available: http://stacks.iop.org/0029-5515/55/i=2/a=023012

[49] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[50] B. Bue and T. Stepinski, "Automated classification of landforms on mars," *Computers & Geosciences*, vol. 32, no. 5, pp. 604 – 614, 2006.

[51] T. F. Stepinski, S. Ghosh, and R. Vilalta, "Automatic recognition of landforms on mars using terrain segmentation and classification," in *Proceedings of the International Conference on Discovery Science, LNAI 4265*, 2006, pp. 255–266.

[52] T. Stepinski and R. Vilalta, "Digital Topography Models for Martian Surfaces," *IEEE Geoscience and Remote Sensing Letters*, vol. 2, pp. 260–264, Jul. 2005.

[53] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 3730–3739.

[54] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Transactions on Knowledge & Data Engineering*, vol. 26, no. 5, pp. 1076–1089, May 2014.

[55] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 1410–1417.

[56] ——, "Transfer feature learning with joint distribution adaptation," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[57] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, Jan. 2016.

[58] K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR '12. IEEE Computer Society, 2012, pp. 2066–2073.

[59] D. D. Jensen and P. R. Cohen, "Multiple comparisons in induction algorithms," *Machine Learning*, vol. 38, no. 3, pp. 309–338, Mar. 2000.