Active Transfer Learning under Model Shift

Xuezhi Wang XUEZHIW@CS.CMU.EDU
Computer Science Department, Carnegie Mellon University

Tzu-Kuo Huang TZUKUOH@CS.CMU.EDU
Robotics Institute, Carnegie Mellon University

Jeff Schneider SCHNEIDE@CS.CMU.EDU
Robotics Institute, Carnegie Mellon University

Abstract

Transfer learning algorithms are used when one has sufficient training data for one supervised learning task (the source task) but only very limited training data for a second task (the target task) that is similar but not identical to the first. These algorithms use varying assumptions about the similarity between the tasks to carry information from the source to the target task. Common assumptions are that only certain specific marginal or conditional distributions have changed while all else remains the same. Alternatively, if one has only the target task, but also has the ability to choose a limited amount of additional training data to collect, then active learning algorithms are used to make choices which will most improve performance on the target task. These algorithms may be combined into active transfer learning, but previous efforts have had to apply the two methods in sequence or use restrictive transfer assumptions.

We propose two transfer learning algorithms that allow changes in all marginal and conditional distributions but assume the changes are smooth in order to achieve transfer between the tasks. We then propose an active learning algorithm for the second method that yields a combined active transfer learning algorithm. We demonstrate the algorithms on synthetic functions and a real-world task on estimating the yield of vineyards from images of the grapes.

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

1. Introduction

In a classical transfer learning setting, we have sufficient fully labeled data from the source domain (also denoted as the training domain), (X^{tr}, Y^{tr}), where the data points, X^{tr}, are fully observed and all corresponding labels, Y^{tr}, are also known. We are given data points, X^{te}, from the target domain (also denoted as the test domain), but few or none of the corresponding labels, Y^{te}, are given. The source domain and the target domain are related but not identical, thus the joint distributions, $P(X^{tr}, Y^{tr})$ and $P(X^{te}, Y^{te})$, are different across the two domains. Most statistical models learned from the source domain do not directly apply to the target domain. However, it may still be possible to avoid the cost of collecting an entire new labeled training data set. The goal of transfer learning is to reduce the amount of new labeled data needed in the target domain. It learns and transfers a model based on the labeled data from the source domain and the unlabeled data from the target domain. Some real-world applications of transfer learning include adapting a classification model that is trained on some products to help learn classification models for some other products (Pan & Yang, 2009), and learning a model on the medical data for one disease and transferring it to another disease.

We are motivated by an autonomous agriculture application where we want to manage the growth of grapes in a vineyard (Nuske et al., 2012). A robot can easily take images of the crop throughout the growing season. At the end of the season the yield will be known for every vine because the product is weighed after harvest. This data can be used to learn a model that predicts yield from images. However, decisions about selling the produce and nurturing the growth must be made mid-season. Acquiring training labels at that time is very expensive because it requires a human to go out and manually estimate yield. Ideally, a model learned from previous years and/or on other grape varieties can be used with a transfer learning algorithm to minimize

this manual yield estimation. Furthermore, we would like a simultaneously applied active learning algorithm to tell us which vines to assess manually. Finally, there are two different objectives of interest. A robot that needs to decide which vines to water needs an accurate estimate of the current yield of each vine. However, a farmer that wants to know how big his crop will be this fall so he can pre-sell an appropriate amount of it only needs an estimate of the sum of the vine yields. We call these problems active learning and active surveying respectively and they lead to different selection criteria.

In this paper, we focus our attention on real-valued regression problems. We propose two transfer algorithms that allow both $P(X)$ and $P(Y|X)$ to change across the source and target tasks. We assume only that the change is smooth as a function of X. The first approach builds on the kernel mean matching (KMM) idea (Huang et al., 2007; Gretton et al., 2007) to match the conditional distributions, $P(Y|X)$, between the tasks. The second approach uses a Gaussian Process to model the source task, the target task, and the offset between. The assumption here is that although the offset may be a nonlinear function over the input domain, there is some smoothness in that offset over the input domain. If that is not true, we suspect there is little hope for transferring between domains at all. The GP-based approach naturally lends itself to the active learning setting where we can sequentially choose query points from the target dataset. Its final predictive covariance, which combines the uncertainty in the transfer function and the uncertainty in the target label prediction, can be plugged into various GP based active query selection criteria. Specifically, we consider (1) active learning which tries to reduce total predictive variance (Ji & Han, 2012); and (2) active surveying which tries to predict $\sum_i Y_i^{te}$ (Garnett et al., 2012).

We evaluate our methods on synthetic data and real-world grape image data. The experimental results show that our transfer learning algorithms significantly outperform covariate-shift methods with few labeled target data points, and our combined active transfer learning algorithm transfers knowledge from the source data and makes target labeling requests that achieve better prediction performance on the target data than alternative methods.

2. Related Work

Traditional methods for transfer learning, including Markov logic networks (Mihalkova et al., 2007), parameter learning (Do & Ng, 2005; Raina et al., 2006), Bayesian Network structure learning (Niculescu-Mizil & Caruana, 2007) consider models where specific parts of the model can be carried over between tasks. Some transfer learning work has focused on the problem of covariate shift (Shimodaira, 2000; Huang et al., 2007; Gretton et al., 2007). They consider the case where the distributions on X are different across domains, i.e., $P(X^{tr})$ differs from $P(X^{te})$, while making the assumption that the conditional distributions $P(Y^{tr}|X^{tr})$ and $P(Y^{te}|X^{te})$ are the same. Following these assumptions they propose the kernel mean matching method to minimize $||\mu(P_{te}) - \mathbf{E}_{x\sim P_{tr}(x)}[\beta(x)\phi(x)]||$ over a re-weighting vector β on training data points such that distributions on X are matched with each other. They then incorporate the learned weights $\hat{\beta}$ into the training procedure, e.g., training an SVM with re-weighted source data points, to obtain a model that generalizes well on the target data. The advantage of using kernel mean matching is to avoid density estimation, which is difficult in high dimensions. It has been proved (Song et al., 2009) that even if we use the empirical version of mean embeddings we can still achieve a fast convergence rate of $O(m^{-1/2})$, where m is the sample size. The algorithms we propose in this paper will allow more than just the marginal on X to shift.

Some recent research (Zhang et al., 2013) has focused on modeling target shift (different $P(Y)$) and conditional shift (different $P(X|Y)$). They assume that X depends causally on Y, thus they can re-weight $P(Y)$ (assuming support of $P(Y^{te}) \subseteq$ support of $P(Y^{tr})$, i.e., the training set is richer than the test set) to match the distributions $P(Y)$. They apply a location-scale transformation on X to match the distributions on $P(X|Y)$. More specifically, they transform X^{tr} to X^{new} by $X^{new} = X^{tr} \odot \mathbf{W} + \mathbf{B}$, then by minimizing the MMD $||\mu[P_X^{new}] - \mu[P_X^{te}]||$ they try to find the optimal transformation. However, they do not assume they can obtain additional labels, Y^{te}, from the target domain, and thus make no use of the labels Y^{te}, even if some are available.

There also have been a few papers dealing with differences in $P(Y|X)$. Jiang & Zhai. (2007) designed specific methods (change of representation, adaptation through prior, and instance pruning) to solve the label adaptation problem. Liao et al. (2005) relaxed the requirement that the training and testing examples be drawn from the same source distribution in the context of logistic regression. They also proposed an active learning approach using the Fisher information matrix, which is a lower bound of the exact covariance matrix. Sun et al. (2011) weighted the samples from the source domain to handle the domain adaptation. These settings are relatively restricted while we consider a more general case that there is a smooth transformation from the source domain to the target domain, hence all source data will be used with the advantage that the part of source data which do not help prediction in the target domain will automatically be corrected via an offset model.

The idea of combining transfer learning and active learning has also been studied recently. Shi et al. (2008) and

Rai et al. (2010) perform transfer and active learning in multiple stages. The first work uses the source data without any domain adaptation. The second work performs domain adaptation at the beginning without further refinement. Saha et al. (2011) and Chattopadhyay et al. (2013) consider active learning under covariate shift and still assume the conditional distributions $P(Y|X)$ are the same across the source and the target domain.

A research area we draw from is active learning with Gaussian Processes, for which many selection criteria have been proposed, such as choosing the test point with the highest variance (or entropy). We can also utilize mutual information (Guestrin et al., 2005), which the same authors further improve by considering both parameter (kernel width) uncertainty reduction (exploration) and model uncertainty reduction under current parameter setting (exploitation) (Krause & Guestrin, 2007). Another popular criterion is minimizing the total variance conditioned on the point to be selected (Seo et al., 2000; Ji & Han, 2012), which can be done using the trace of the covariance matrix, $\mathrm{Tr}\{\sigma^2_{y|A}\}$, where A is the set of labeled data points and the candidate query points. Active surveying (Garnett et al., 2012; Ma et al., 2013), uses an estimation objective that is the sum of all the labels in the test set. The corresponding myopic active selection criteria is minimizing the sum of all elements in the covariance matrix conditioned on the selected point, $\mathbf{1}^\top \sigma^2_{y|A}\mathbf{1}$. We adopt these last two selection criteria for our active transfer algorithms.

3. Approach

3.1. Problem Formulation

Assume we are given a set of n labeled training data points, (X^{tr}, Y^{tr}), from the source domain where each $X^{tr}_i \in \Re^{d_x}$ and each $Y^{tr}_i \in \Re^{d_y}$. Assume we are also given a set of m test data points, X^{te}, from the target domain. Some of these will have corresponding labels, Y^{teL}. When necessary we will separately denote the subset of X^{te} that has labels as X^{teL}, and the subset that does not as X^{teU}.

For *static transfer learning*, the goal is to learn a predictive model using all the given data that minimizes the squared prediction error on the test data, $\Sigma^m_{i=1}(\hat{Y}^{te}_i - Y^{te}_i)^2$ where $\hat{Y}_i$ and Y_i are the predicted and true labels for the ith test data point. We will evaluate the transfer learning algorithms by including a subset of labeled test data chosen uniformly at random.

For *active transfer learning* the performance metric is the same. The difference is that the active learning algorithm chooses the test points for labeling rather than being given a randomly chosen set.

The surveying metric is to minimize the error on the sum of the predictions: $(\Sigma^m_{i=1}\hat{Y}^{te}_i - \Sigma^m_{i=1}Y^{te}_i)^2$. Again, this is evaluated using a randomly chosen set of test labels for static transfer surveying or a set chosen by the algorithm for active transfer surveying.

To illustrate the problem, we show a toy example in Figure 1. The left figure shows data in the source domain, drawn from a sine function. The right figure shows data in the target domain, drawn from the same sin function adding a positive offset 1. The middle figure shows the offset. The goal is, given the data in the left figure, and a few data points to query, to recover the function in the right figure in the least number of queries.

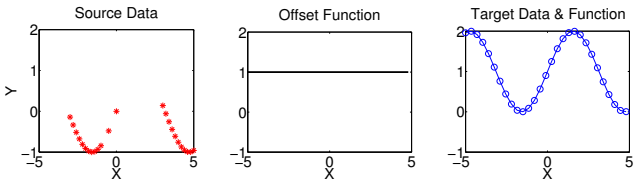

Figure 1. Toy example showing the transfer/active learning problem

3.2. Transfer Learning

3.2.1. Distribution Matching Approach

First we propose a distribution matching approach for transfer learning. The basic idea is, we want to

(1) **Match the conditional distributions** $P(Y^{new}|X^{tr})$ and $P(Y^{te}|X^{te})$, where Y^{new} is under location-scale transform of Y^{tr}: $Y^{new} = Y^{tr} \odot \mathbf{w}(X^{tr}) + \mathbf{b}(X^{tr})$. If the conditional distributions are matched with each other, and $P(X^{tr}) = P(X^{te})$ (which can be achieved by various methods dealing with covariate shift, hence it is not the focus of this paper), then a model learned from the source data will generalize well on the target data because the joint distribution is also matched with each other, i.e., $P(X^{tr}, Y^{tr}) = P(X^{te}, Y^{te})$.
(2) **The transform function is smooth**, i.e., $\mathbf{w}$ and $\mathbf{b}$ should be smooth w.r.t X.

To achieve the first goal, similar to the kernel mean matching idea, we can directly minimize the discrepancy of the conditional embedding of the two distributions (K in the following equations stands for the Gaussian kernel, and K_{XY} represents the kernel between matrix X and Y) with a regularization term:

$$\begin{aligned} &\min_{\mathbf{w},\mathbf{b}} \; L + L_{reg}, \text{where} \\ &L = ||\mu[P_{Y^{new}|X^{tr}}] - \mu[P_{Y^{te}|X^{te}}]||^2_F \\ &L_{reg} = \lambda_{reg}(||\mathbf{w} - \mathbf{1}||^2 + ||\mathbf{b}||^2) \end{aligned} \tag{1}$$

L can be further simplified by

Algorithm 1 Conditional Distribution Matching

Input: $X^{tr}, Y^{tr}, \{X^{teL}, Y^{teL}\} \cup X^{teU}$
Initialize $\mathbf{w} = \mathbf{1}, \mathbf{b} = \mathbf{0}$
repeat
  Predict $\hat{Y}^{teU}$ using $\{X^{tr}, Y^{new}\} \cup \{X^{teL}, Y^{teL}\}$, where Y^{new} is transformed using current $\mathbf{w}, \mathbf{b}$
  Optimize the objective function in Equation 1
until $\mathbf{w}, \mathbf{b}$ converge
Output: Prediction $\hat{Y}^{teU}$

$$
\begin{aligned}
L =& ||\psi(Y^{new})(K_{X^{tr}X^{tr}} + \lambda I)^{-1}\phi^\top(X^{tr}) - \\
& \psi(Y^{te})(K_{X^{te}X^{te}} + \lambda I)^{-1}\phi^\top(X^{te})||_F^2 \\
=& C + \mathrm{Tr}\{\phi(X^{tr})(L^{tr} + \lambda I)^{-1}\tilde{K}(L^{tr} + \lambda I)^{-1}\phi^\top(X^{tr})\} \\
& - 2\,\mathrm{Tr}\{\phi(X^{tr})(L^{tr} + \lambda I)^{-1}\tilde{K}^c(L^{te} + \lambda I)^{-1}\phi^\top(X^{te})\} \\
=& C + \mathrm{Tr}\{(L^{tr} + \lambda I)^{-1}\tilde{K}(L^{tr} + \lambda I)^{-1}L^{tr}\} \\
& - 2\,\mathrm{Tr}\{(L^{tr} + \lambda I)^{-1}\tilde{K}^c(L^{te} + \lambda I)^{-1}K_{X^{te}X^{tr}}\},
\end{aligned}
$$

where $\tilde{K} = K_{Y^{new}Y^{new}}, \tilde{K}^c = K_{Y^{new}Y^{te}}, L^{tr} = K_{X^{tr}X^{tr}}, L^{te} = K_{X^{te}X^{te}}$. λ is the regularization parameter to ensure the kernel matrix is invertible.

To make the transformation smooth w.r.t. X, we parameterized $\mathbf{w}, \mathbf{b}$ in this way (Zhang et al., 2013): $\mathbf{w} = R\mathbf{g}, \mathbf{b} = R\mathbf{h}$, where $R = L^{tr}(L^{tr}+\lambda I)^{-1}$. We use scaled conjugate gradient to minimize the objective function. The derivation of the required derivatives is given in the supplementary materials.

When matching the conditional distributions, if we only use X^{teL}, Y^{teL} in the empirical version of the conditional operator $\mu[P_{Y^{te}|X^{te}}]$, it will be unstable due to the small size of the observed labeled test points, especially in the early stage of active learning. However, using both X^{teL}, Y^{teL} and X^{teU}, Y^{teU} would require knowing the values Y^{teU}, which are not obtained before querying. We replace Y^{teU} with the prediction $\hat{Y}^{teU}$ based on $\{X^{tr}, Y^{new}\} \cup \{X^{teL}, Y^{teL}\}$, where Y^{new} are under transformation using current $\mathbf{w}, \mathbf{b}$, while $\{X^{teL}, Y^{teL}\}$ are the labeled test data selected up to the present. After obtaining $\hat{Y}^{teU}$ we minimize the objective function Eq 1. We iterate over the two steps until convergence. The algorithm is described as in Algorithm 1.

3.2.2. Offset Approach

In the second proposed method, we use a Gaussian Process to model the source task, the target task, and the offset between, described as follows (K in the following equations stands for the Gaussian kernel, and λ is the regularization parameter to ensure the kernel matrix is invertible):

(1) We build a GP from the source domain and predict on X^{teL}, then compute the offset Z between the prediction and the true labels Y^{teL}: $\hat{Z}^{teL} = Y^{teL} - \hat{Y}^{teL}$. It follows: $P(\hat{Z}^{teL}|X^{tr}, Y^{tr}, X^{teL}, Y^{teL}) \sim \mathcal{N}(\mu_s, \Sigma_s)$, where $\mu_s = Y^{teL} - K_{X^{teL}X^{tr}}(K_{X^{tr}X^{tr}} + \lambda I)^{-1}Y^{tr}$, $\Sigma_s = K_{X^{teL}X^{teL}} - K_{X^{teL}X^{tr}}(K_{X^{tr}X^{tr}} + \lambda I)^{-1}K_{X^{tr}X^{teL}}$.

(2) We transform Y^{tr} to Y^{new} by $Y^{new} = Y^{tr} + \hat{Z}^{tr}$, where $\hat{Z}^{tr}$ is the predicted mean of the offset on X^{tr} using the GP built from $\{X^{teL}, \hat{Z}^{teL}\}$, i.e., $P(\hat{Z}^{tr}|\hat{Z}^{teL}, X^{tr}, X^{teL}) \sim \mathcal{N}(\mu_0, \Sigma_0)$, where $\mu_0 = K_{X^{tr}X^{teL}}(K_{X^{teL}X^{teL}} + \lambda I)^{-1}\hat{Z}^{teL}$, $\Sigma_0 = K_{X^{tr}X^{tr}} - K_{X^{tr}X^{teL}}(K_{X^{teL}X^{teL}} + \lambda I)^{-1}K_{X^{teL}X^{tr}}$.

(3) Train a model on $\{X^{tr}, Y^{new}\} \cup \{X^{teL}, Y^{teL}\}$, use the model to make predictions on X^{teU}.

3.3. Active Learning

We consider two active learning goals and apply a myopic selection criteria to each:
(1) **Active learning** which tries to reduce the total predictive variance (Ji & Han, 2012). An optimal myopic selection is achieved by choosing the point which minimizes the trace of the predictive covariance matrix conditioned on that selection: $\mathrm{Tr}\{\sigma^2_{y|A}\}$.
(2) **Active surveying** which tries to predict $\sum_i Y_i^{te}$. An optimal myopic selection is achieved by choosing the point which minimizes the sum over all elements of the covariance matrix conditioned on that selection (Garnett et al., 2012), which is also denoted Σ-optimality in (Ma et al., 2013): $\mathbf{1}^\top\sigma^2_{y|A}\mathbf{1}$.

Note that the predictive covariances for a Gaussian process are computed without using the observed labels. This means that conditioning on hypothetical point selections can be done quickly without needing to marginalize out the unknown label. All that is needed to create an integrated active transfer algorithm using the offset approach from the previous section is to determine the corresponding predictive covariance matrices so the active selection criteria can be applied. We now derive these.

3.3.1. Uncertainty for transforming the training labels

Given $P(\hat{Z}^{teL}|X^{tr}, Y^{tr}, X^{teL}, Y^{teL}) \sim \mathcal{N}(\mu_s, \Sigma_s)$, $P(\hat{Z}^{tr}|\hat{Z}^{teL}, X^{tr}, X^{teL}) \sim \mathcal{N}(\mu_0, \Sigma_0)$, to model the uncertainty for transforming the labels Y^{tr}, we need to integrate over $\hat{Z}^{teL}$, i.e.,

$$
\begin{aligned}
& P(\hat{Z}^{tr}|X^{tr}, Y^{tr}, X^{teL}, Y^{teL}) \\
=& \int_{\hat{Z}^{teL}} P(\hat{Z}^{tr}, \hat{Z}^{teL}|X^{tr}, Y^{tr}, X^{teL}, Y^{teL})d(\hat{Z}^{teL}) \\
=& \int_{\hat{Z}^{teL}} P(\hat{Z}^{tr}|\hat{Z}^{teL}, X^{tr}, X^{teL}) \\
& P(\hat{Z}^{teL}|X^{tr}, Y^{tr}, X^{teL}, Y^{teL})d(\hat{Z}^{teL}).
\end{aligned}
$$

Denote $K_1 = K_{X^{tr}X^{teL}}(K_{X^{teL}X^{teL}} + \lambda I)^{-1}$, we can derive that $P(\hat{Z}^{tr}|X^{tr}, Y^{tr}, X^{teL}, Y^{teL}) \sim \mathcal{N}(\mu_1, \Sigma_1)$, where $\mu_1 = \Sigma_1\Sigma_0^{-1}K_1(K_1^\top\Sigma_0^{-1}K_1 + \Sigma_s^{-1})^{-1}\Sigma_s^{-1}\mu_s$, $\Sigma_1 = \Sigma_0 + K_1\Sigma_s K_1^\top$.

3.3.2. Uncertainty for target label prediction

The prediction on X^{teU} is based on the Gaussian Process built from the merged data $\{X^{tr}, Y^{new}\} \cup \{X^{teL}, Y^{teL}\}$, hence it also follows a Gaussian distribution: $P(\hat{Y}^{teU}|X^{teU}, X^{tr}, Y^{new}, X^{teL}, Y^{teL}) \sim \mathcal{N}(\mu, \Sigma)$, where
$\mu = K_{X^{teU}X}(K_{XX} + \lambda I)^{-1}Y = [\Omega_1\ \Omega_2][Y^{new}\ Y^{teL}]^\top$,
$\Sigma = K_{X^{teU}X^{teU}} - K_{X^{teU}X}(K_{XX} + \lambda I)^{-1}K_{XX^{teU}}$.
Here X, Y represent the merged data, i.e., $X = X^{tr} \cup X^{teL}, Y = Y^{new} \cup Y^{teL}$. Ω_1 is the matrix consisting of the first n columns of $K_{X^{teU}X}(K_{XX} + \lambda I)^{-1}$, where n is the number of training data points. Ω_2 consists of the remaining l columns, where l is the size of labeled test points.

3.3.3. The combined uncertainty for final prediction

Due to the uncertainty for the transformed labels Y^{new}, to model the uncertainty for the final prediction again we need to integrate over Y^{new} , i.e.:

$$
\begin{aligned}
&P(\hat{Y}^{teU}|X^{teU}, X^{tr}, Y^{tr}, X^{teL}, Y^{teL}) \\
=&\int_{Y^{new}} P(\hat{Y}^{teU}, Y^{new}|X^{teU}, X^{tr}, Y^{tr}, X^{teL}, Y^{teL})dY^{new} \\
=&\int_{Y^{new}} P(\hat{Y}^{teU}|X^{teU}, X^{tr}, Y^{new}, X^{teL}, Y^{teL}) \\
&P(Y^{new}|X^{tr}, Y^{tr}, X^{teL}, Y^{teL})dY^{new} \\
=&C\int_{Y^{new}} \exp\{-\frac{1}{2}(\hat{Y}^{teU} - \mu)^\top\Sigma^{-1}(\hat{Y}^{teU} - \mu)\} \\
&\exp\{-\frac{1}{2}(Y^{new} - Y^{tr} - \mu_1)^\top\Sigma_1^{-1}(Y^{new} - Y^{tr} - \mu_1)\}dY^{new} \\
=&C'\int_{Y^{new}} \exp\{-\frac{1}{2}(Y_* - \Omega_1 Y^{new})^\top\Sigma^{-1}(Y_* - \Omega_1 Y^{new})\} \\
&\exp\{-\frac{1}{2}(Y^{new} - \mu_1)^\top\Sigma_1^{-1}(Y^{new} - \mu_1)\}dY^{new},
\end{aligned}
$$

where $Y_* = \hat{Y}^{teU} - \Omega_2 Y^{teL}$.

After some derivation we can get $P(\hat{Y}^{teU}|X^{teU}, X^{tr}, Y^{tr}, X^{teL}, Y^{teL}) \sim \mathcal{N}(\mu_2, \Sigma_2)$, where
$\mu_2 = \Sigma_2\Sigma^{-1}\Omega_1(\Omega_1^\top\Sigma^{-1}\Omega_1 + \Sigma_1^{-1})^{-1}\Sigma_1^{-1}(\mu_1 + Y^{tr})$,
$\Sigma_2 = \Sigma + \Omega_1\Sigma_1\Omega_1^\top = \Sigma + \Omega_1(\Sigma_0 + K_1\Sigma_s K_1^\top)\Omega_1^\top$.

Hence we get $\mu(\hat{Y}^{teU}) =$

$$\Omega_2 Y^{teL} + \Sigma_2\Sigma^{-1}\Omega_1(\Omega_1^\top\Sigma^{-1}\Omega_1 + \Sigma_1^{-1})^{-1}\Sigma_1^{-1}(\mu_1 + Y^{tr}).$$

For more detailed derivation please refer to the supplementary materials.

4. Experiments

4.1. Synthetic Dataset

4.1.1. Data Description

We generate two synthetic datasets. The first one has a constant shift between the labels Y^{tr} and Y^{te}. The second one has a shift in both the data points X^{tr}, X^{te} and their labels Y^{tr} and Y^{te}.
(1) Synthetic Dataset 1 (using matlab notation):
Source: X^{tr} = [-3:0.2:-1 -0.5:0.5:0 3:0.2:5]; $Y^{tr} = \sin(X^{tr})$; **Target:** X^{te} = [-5:0.35:5]; $Y^{te} = \sin(X^{te}) + 1$.
(2) Synthetic Dataset 2 (using matlab notation):
Source: X^{tr} = [-5:0.2:-1 -0.5:0.5:0.5 1:0.2:5]; $Y^{tr} = \sin(X^{tr})$; **Target:** X^{te} = [-5:0.35:5]; $Y^{te} = \sin(X^{te} + 1)$.
Illustrations for the two datasets are shown as in Figure 2.

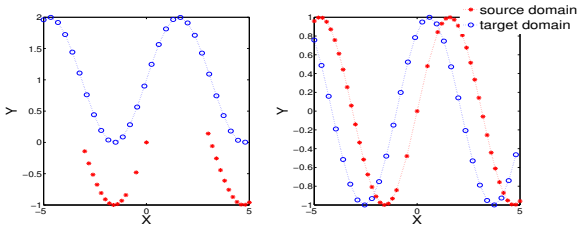

Figure 2. Illustration of two synthetic datasets

4.1.2. Transfer Learning on Synthetic Dataset

We compare the following methods:
(1) **distribution approach**, described in section 3.2.1.
(2) **offset approach**, described in section 3.2.2.
(3) **use only test x**. GP Prediction using only labeled test points (i.e. no transfer learning).
(4) **use both x**. GP Prediction using both training points and labeled test points, without any transfer learning.
(5) **KMM** for covariate shift (Huang et al., 2007).
(6) **Target/Conditional shift**, proposed by (Zhang et al., 2013), code is from http://people.tuebingen.mpg.de/kzhang/Code-TarS.zip.

The evaluation metric is the mean squared error of predictions on the unlabeled test points with different numbers of observed test points with labels, and averaged over 10 experiments. Parameters (kernel width, regularization term, etc.) are set using cross validation. In the test domain initially there is not much data for tuning parameters using cross validation, we assume the same smoothness constraint (same kernel width and λ) as in the source domain. The selection of which test points to label is done uniformly at random. Results for Synthetic Datasets 1 and 2 are shown in Figures 3 and 4, respectively. From the results we can see that for observed test points with labels fewer than 10, our proposed methods can greatly reduce the prediction error by transferring the model learned from

the source domain. With more points the errors tend to converge to using only X^{teL}, Y^{teL} because the number of labeled points in the test domain is large enough for learning a good model by itself. KMM and Target/Conditional shift methods do not utilize the possible label information Y^{teL}, hence the error is much larger compared to other methods which use a few Y^{te}'s.

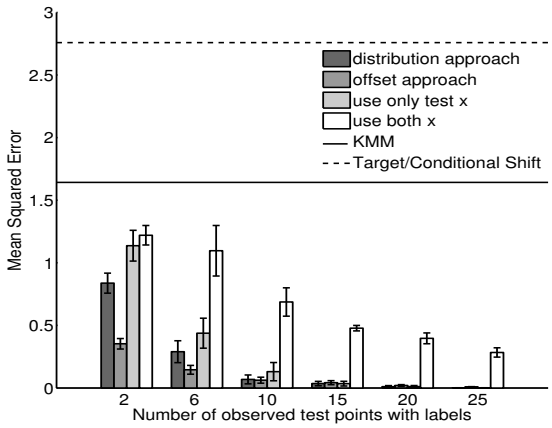

Figure 3. MSE for transfer learning on synthetic dataset 1

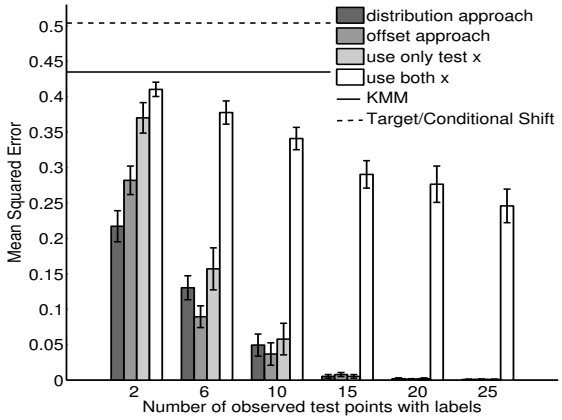

Figure 4. MSE for transfer learning on synthetic dataset 2

4.1.3. Active Learning/Surveying on Synthetic Dataset

We consider two active learning goals: (1) *Active Learning to reduce the total predictive variance* (shortened to *Active Learning*, or AL in the following description) and (2) *Active Surveying* (AS). We compare the following uncertainty measures for each goal:
(1) **combined**. AL/AS using the combined covariance matrix (Σ_2 in section 3.3).
(2) **source**. AL/AS using the covariance matrix based only on the source domain, i.e., $K_{X^{teU}X^{teU}} - K_{X^{teU}X^{tr}}(K_{X^{tr}X^{tr}} + \lambda I)^{-1}K_{X^{tr}X^{teU}}$.
(3) **target**. AL/AS using the covariance matrix based only on the target domain, i.e., $K_{X^{teU}X^{teU}} - K_{X^{teU}X^{teL}}(K_{X^{teL}X^{teL}} + \lambda I)^{-1}K_{X^{teL}X^{teU}}$.
(4) **both**. AL/AS using the covariance matrix based on both source and target domain, i.e., $K_{X^{teU}X^{teU}} - K_{X^{teU}\tilde{X}}(K_{\tilde{X}\tilde{X}}+\lambda I)^{-1}K_{\tilde{X}X^{teU}}$, where $\tilde{X} = X^{tr} \cup X^{teL}$.
(5) **random**. Points selected uniformly at random.

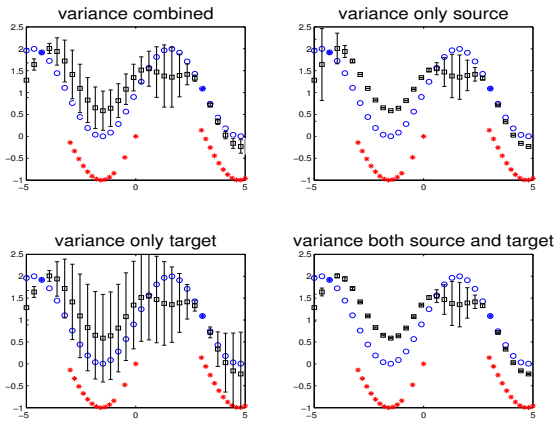

Figure 5. The comparison of different covariance matrices. Red stars show the data from the source domain, and blue circles show the data from the target domain. The black bars show the error bar/uncertainty (diagonal elements of the covariance matrix) on the prediction of unlabeled test points. The two labeled test points are shown in filled blue circles ($x_1 = -4.3, x_2 = 3.05$).

To better illustrate how the combined covariance matrix compares to other covariance matrices, we show a comparison by plotting the diagonal elements of each covariance matrix, as the uncertainty for prediction on the unlabeled points (with two points labeled) in the test domain, as shown in Figure 5. Based on what covariance matrix is used for active learning, the most likely selection for the unlabled test points are: (a) **source**: points far away from the source data; (b) **target**: points far away from the labeled test points; (c) **both**: points far away from both the source data and the labeled test points; (d) **combined**: the uncertainty of unlabeled test points will be approximately ranked as (from highest to lowest), (1) points far away from both the source data and the labeled test points, (2) points far away from the labeled test points but close to the source data, and points far away from the source data but close to the labeled test points, (3) points close to both the source data and the labeled test points.

We consider the mean squared error (for *Active Learning*) and absolute error (for *Active Surveying*) with respect to

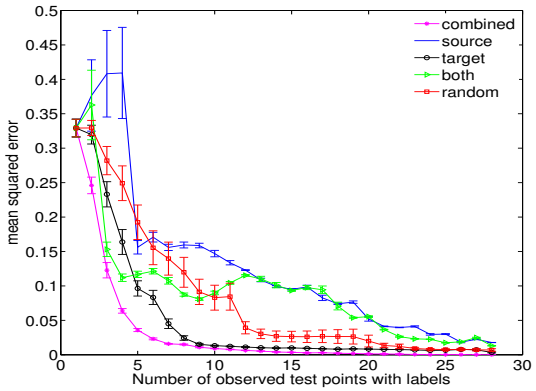

Figure 6. MSE for *Active Learning* on Synthetic Dataset 1

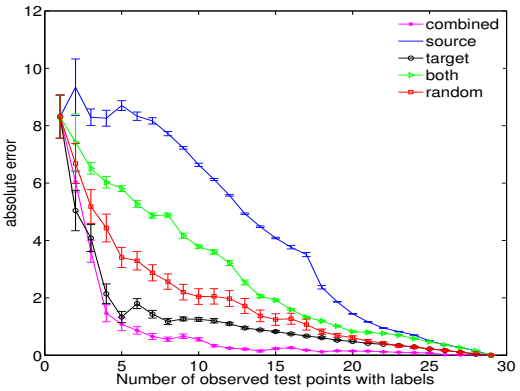

Figure 7. Absolute Error for *Active Surveying* on Synthetic Dataset 1

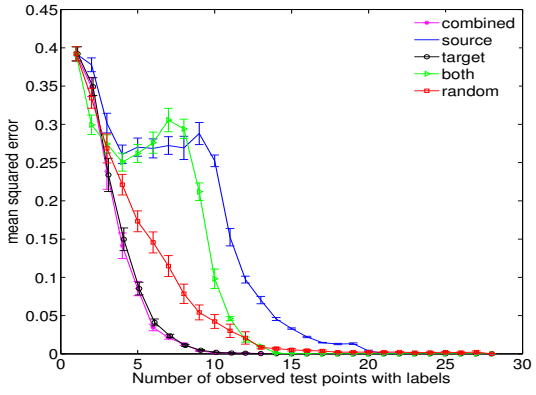

Figure 8. MSE for *Active Learning* on Synthetic Dataset 2

different number of observed test points with labels (in the order being selected by the corresponding active selection criteria). We averaged the results over 29 experiments, each one initiated with a test point chosen uniformly at random.

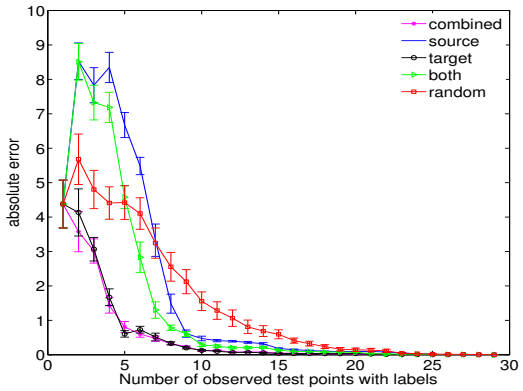

Figure 9. Absolute Error for *Active Surveying* on Synthetic Dataset 2

On Synthetic Dataset 1, *Active Learning* results are shown in Figure 6, and *Active Surveying* Results are shown in Figure 7. On Synthetic Dataset 2, *Active Learning* results are shown in Figure 8, and *Active Surveying* Results are shown in Figure 9. From the results we can see that, on Synthetic Dataset 1, for both *Active Learning* and *Active Surveying* our proposed combined covariance matrix (Σ_2 in section 3.3) clearly outperforms all other baselines. On Synthetic Dataset 2, our gain of using combined covariance matrix is smaller because Y^{te} differs from Y^{tr} at almost every location of X. Hence choosing a point corresponding to a larger transfer learning gain becomes very similar to choosing the point uniformly, which is the selection strategy of using covariance matrix merely based on the target domain.

4.2. Real-world Dataset

4.2.1. Transfer Learning on Real-world Dataset

We have two datasets with grape images taken from vineyards and the number of grapes on them as labels, one is riesling (128 labeled images), another is traminette (96 labeled images), as shown in Figure 10. The goal is to transfer the model learned from one kind of grape dataset to another kind of grape dataset. The total number of grapes for these two datasets are $19,253$ and $30,360$, respectively.

We extract raw-pixel features from the images, and use Random Kitchen Sinks (Rahimi & Recht, 2007) to get the coefficients as feature vectors (Oliva et al., 2014). We use Gaussian Process for prediction. On the traminette dataset we have achieved R-squared correlation 0.754 (95% for training and 5% for test). People have been using specifically designed image processing methods to detect grapes and achieved R-squared correlation 0.73 (Nuske et al., 2012). Grape-detection method takes lots of manual la-

Figure 10. A part of one image from each grape dataset

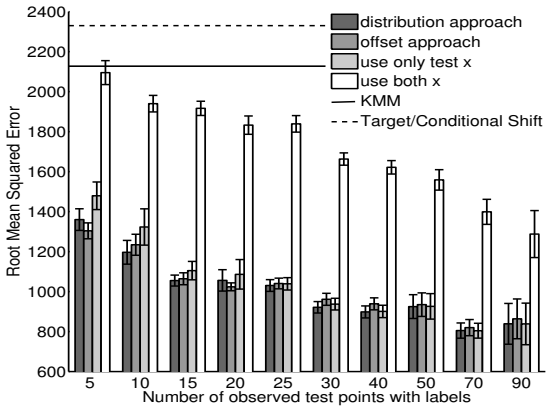

Figure 11. RMSE for transfer learning on the real grape data

beling work and cannot be directly applied across different kinds of grapes (due to difference in size and color). Our proposed approach for transfer learning, however, can be directly used for different kinds of grapes or even different kinds of crops.

We compare to the same baselines for both transfer learning and active learning goals as in the synthetic experiments. For transfer learning the results are shown in Figure 11, averaged over 10 experiments. We can see with labeled test points fewer than 25, our proposed approaches (both distribution matching approach and the offset approach) can reduce the error by transferring the model learned from the source domain. The *Active Learning* result is shown in Figure 12, and the *Active Surveying* result is shown in Figure 13. From the results we can see that our proposed method can well achieve both goals.

5. Conclusion and Discussions

In this paper, we propose two transfer learning algorithms that allow changes in all marginal and conditional distributions with the additional assumption that the changes are smooth as a function of X. The first approach is based on conditional distribution matching, and the second is based on modeling the source/target task and the offset between using Gaussian Processes. We then propose an active learning method which yields a combined active transfer algorithm. Results on both synthetic datasets and a real-world dataset demonstrate the effectiveness of our proposed methods.

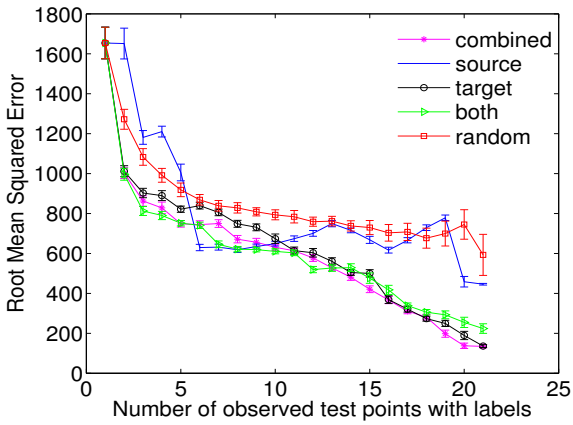

Figure 12. RMSE for *Active Learning* on the real data

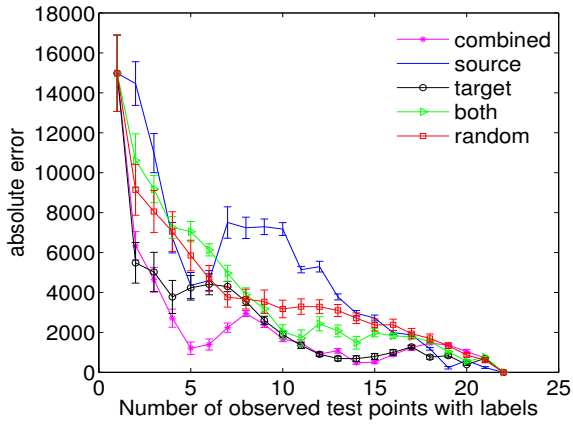

Figure 13. Absolute error for *Active Surveying* on the real data

About the convergence guarantee, for the distribution matching approach, one deficiency of using the Gaussian kernel is it results in a non-convex optimization objective. This problem could potentially be resolved by using a linear kernel. However, using a linear kernel would make the results less general. For the offset approach, the labels we get are biased by our selection scheme. However, just like most other Bayesian sequential learning methods, the offset approach empirically often converges to a good estimate as we get more labels.

Acknowledgments

This work is supported in part by the US Department of Agriculture under grant number 20126702119958.

References

Chattopadhyay, Rita, Fan, Wei, Davidson, Ian, Panchanathan, Sethuraman, and Ye, Jieping. Joint transfer and batch-mode active learning. In *ICML 2013*, 2013.

Do, Cuong B and Ng, Andrew Y. Transfer learning for text classification. In *Neural Information Processing Systems Foundation, NIPS 2005*, 2005.

Garnett, Roman, Krishnamurthy, Yamuna, Xiong, Xuehan, Schneider, Jeff, and Mann, Richard. Bayesian optimal active search and surveying. In *ICML 2012*, 2012.

Gretton, Arthur, Borgwardt, Karsten M., Rasch, Malte, Scholkopf, Bernhard, and Smola, Alex. A kernel method for the two-sample-problem. In *NIPS 2007*, 2007.

Guestrin, Carlos, Krause, Andreas, and Singh, Ajit Paul. Near-optimal sensor placements in gaussian processes. In *ICML 2005*, 2005.

Huang, Jiayuan, Smola, Alex, Gretton, Arthur, Borgwardt, Karsten, and Schlkopf, Bernhard. Correcting sample selection bias by unlabeled data. In *NIPS 2007*, 2007.

Ji, Ming and Han, Jiawei. A variance minimization criterion to active learning on graphs. In *AISTATS 2012*, 2012.

Jiang, J. and Zhai., C. Instance weighting for domain adaptation in nlp. In *Proc. 45th Ann. Meeting of the Assoc. Computational Linguistics, pp. 264-271*, 2007.

Krause, Andreas and Guestrin, Carlos. Nonmyopic active learning of gaussian processes: An explorationexploitation approach. In *ICML 2007*, 2007.

Liao, X., Xue, Y., and Carin, L. Logistic regression with an auxiliary data source. In *Proc. 21st Intl Conf. Machine Learning*, 2005.

Ma, Yifei, Garnett, Roman, and Schneider, Jeff. Sigma-optimality for active learning on gaussian random fields. In *NIPS 2013*, 2013.

Mihalkova, Lilyana, Huynh, Tuyen, and Mooney., Raymond J. Mapping and revising markov logic networks for transfer learning. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-2007)*, 2007.

Niculescu-Mizil, Alexandru and Caruana, Rich. Inductive transfer for bayesian network structure learning. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007)*, 2007.

Nuske, S., Gupta, K., Narasihman, S., and Singh., S. Modeling and calibration visual yield estimates in vineyards. In *International Conference on Field and Service Robotics*, 2012.

Oliva, Junier B., Neiswanger, Willie, Poczos, Barnabas, Schneider, Jeff, and Xing, Eric. Fast distribution to real regression. In *AISTATS*, 2014.

Pan, Sinno Jialin and Yang, Qiang. A survey on transfer learning. In *TKDE 2009*, 2009.

Rahimi, A. and Recht, B. Random features for large- scale kernel machines. In *NIPS*, 2007.

Rai, Piyush, Saha, Avishek, III, Hal Daume, and Venkatasubramanian, Suresh. Domain adaptation meets active learning. In *Active Learning for NLP (ALNLP), Workshop at NAACL-HLT 2010*, 2010.

Raina, Rajat, Ng, Andrew Y., and Koller, Daphne. Constructing informative priors using transfer learning. In *Proceedings of the Twenty-third International Conference on Machine Learning*, 2006.

Saha, Avishek, Rai, Piyush, III, Hal Daume, Venkatasubramanian, Suresh, and DuVall, Scott L. Active supervised domain adaptation. In *ECML*, 2011.

Seo, Sambu, Wallat, Marko, Graepel, Thore, and Obermayer, Klaus. Gaussian process regression: Active data selection and test point rejection. In *IJCNN 2000*, 2000.

Shi, Xiaoxiao, Fan, Wei, and Ren, Jiangtao. Actively transfer domain knowledge. In *ECML*, 2008.

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

Song, Le, Huang, Jonathan, Smola, Alex, and Fukumizu, Kenji. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *ICML 2009*, 2009.

Sun, Qian, Chattopadhyay, Rita, Panchanathan, Sethuraman, and Ye, Jieping. A two-stage weighting framework for multi-source domain adaptation. In *NIPS*, 2011.

Zhang, Kun, Schlkopf, Bernhard, Muandet, Krikamol, and Wang, Zhikun. Domian adaptation under target and conditional shift. In *ICML 2013*, 2013.