

---

# Flexible Transfer Learning under Support and Model Shift

---

**Xuezhi Wang**  
Computer Science Department  
Carnegie Mellon University  
xuezhiw@cs.cmu.edu

**Jeff Schneider**  
Robotics Institute  
Carnegie Mellon University  
schneide@cs.cmu.edu

## 1 Introduction

In a classical transfer learning setting, we have sufficient fully labeled data from the source domain (or the training domain) where we fully observe the data points  $X^{tr}$ , and all corresponding labels  $Y^{tr}$  are known. On the other hand, we are given data points,  $X^{te}$ , from the target domain (or the test domain), but few or none of the corresponding labels,  $Y^{te}$ , are given. The source and the target domains 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.

The real-world application we consider is an autonomous agriculture application where we want to manage the growth of grapes in a vineyard [3]. Recently, robots have been developed to take images of the crop throughout the growing season. The measured yield after each harvest season can be used to learn a model to predict yield from images. Farmers would like to know their yield early in the season so they can make better decisions on selling the produce or nurturing the growth. Acquiring training labels early in the season is very expensive because it requires a human to go out and manually estimate the yield. Ideally, we can apply a transfer-learning model which learns from previous years and/or on other grape varieties to minimize this manual yield estimation.

In this paper, we focus our attention on real-valued regression problems. We propose a transfer learning algorithm that allows both the support on  $X$  and  $Y$ , and the model  $P(Y|X)$  to change across the source and target domains. We assume only that the change is smooth as a function of  $X$ . In this way, more flexible transformations are allowed than mean-centering and variance-scaling.

As an illustration, we show a toy problem in Fig. 1, where neither the support of  $P(X)$  or the support of  $P(Y)$  overlap across the two domains. In Fig. 2, we show the labels (the yield) of two real-world grape image dataset (Fig. 3), along with the 3rd dimension of its feature space. We can see that the real-world problem is quite similar to the toy problem, which indicates that the algorithm we propose in this paper will be both useful and practical for real applications.

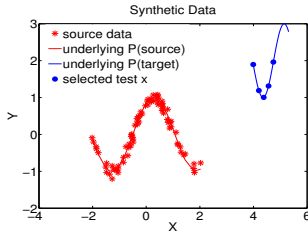


Figure 1: Toy problem

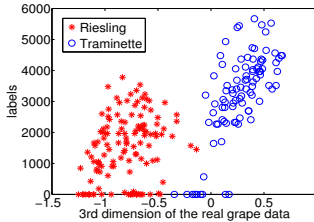


Figure 2: Real grape data



Figure 3: A part of one image from each grape dataset

We evaluate our methods on synthetic data and real-world grape image data. The experimental results show that our transfer learning algorithms significantly outperform existing methods with few labeled target data points. This work is included in our paper [1].

## 2 Related Work

Transfer learning is applied when joint distributions differ across source and target domains. Traditional methods for transfer learning use Markov logic networks [4], parameter learning [5, 6], and Bayesian Network structure learning [7], where specific parts of the model are considered to be carried over between tasks.

Recently, a large part of transfer learning work has focused on the problem of covariate shift [8, 9, 10]. However, this work suffers two major problems. First, the conditional distribution  $P(Y|X)$  is assumed to be the same, which might not be true under many real-world cases. Second, the KMM method requires that the support of  $P(X^{te})$  is contained in the support of  $P(X^{tr})$ , i.e., the training set is richer than the test set. If it is not true, one might mean-center (and possibly also variance-scale) the data to ensure that the support of  $P(X^{te})$  is contained in (or at least largely overlapped with)  $P(X^{tr})$ . More recent research [12] made a similar assumption on the support of  $P(Y)$ . In this paper, we provide an alternative way to solve the support shift problem that allows more flexible transformations than mean-centering and variance-scaling.

## 3 Approach

### 3.1 Problem Formulation

We are given a set of  $n$  labeled training data points,  $(X^{tr}, Y^{tr})$ , from the source domain where each  $X_i^{tr} \in \mathbb{R}^{d_x}$  and each  $Y_i^{tr} \in \mathbb{R}^{d_y}$ . 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  $X^{teU}$ .

### 3.2 Transfer Learning Approach (SMS)

Our strategy is to simultaneously learn a nonlinear mapping  $X^{te} \rightarrow X^{new}$  and  $Y^{te} \rightarrow Y^*$ . This allows flexible transformations on both  $X$  and  $Y$ , and our smoothness assumption using GP prior makes the estimation stable. We call this method Support and Model Shift (SMS).

We apply the following steps ( $K$  in the following represents the Gaussian kernel, and  $K_{XY}$  represents the kernel between matrices  $X$  and  $Y$ ,  $\lambda$  ensures invertible kernel matrix):

1. Transform  $X^{teL}$  to  $X^{new(L)}$  by a location-scale shift:  $X^{new(L)} = \mathbf{W}^{teL} \odot X^{teL} + \mathbf{B}^{teL}$ , such that the support of  $P(X^{new(L)})$  is contained in the support of  $P(X^{tr})$ ;
2. Build a Gaussian Process on  $(X^{tr}, Y^{tr})$  and predict on  $X^{new(L)}$  to get  $Y^{new(L)}$ ;
3. Transform  $Y^{teL}$  to  $Y^*$  by a location-scale shift:  $Y^* = \mathbf{w}^{teL} \odot Y^{teL} + \mathbf{b}^{teL}$ , then we optimize the following empirical loss:

$$\arg \min_{\mathbf{W}^{teL}, \mathbf{B}^{teL}, \mathbf{w}^{teL}, \mathbf{b}^{teL}, \mathbf{w}^{te}} \|Y^* - Y^{new(L)}\|^2 + \lambda_{reg} \|\mathbf{w}^{te} - \mathbf{1}\|^2, \quad (1)$$

where  $\mathbf{W}^{teL}, \mathbf{B}^{teL}$  are matrices with the same size as  $X^{teL}$ .  $\mathbf{w}^{teL}, \mathbf{b}^{teL}$  are vectors with the same size as  $Y^{teL}$  ( $l$  by 1, where  $l$  is the number of labeled samples in the target domain), and  $\mathbf{w}^{te}$  is an  $m$  by 1 scale vector on all  $Y^{te}$ .  $\lambda_{reg}$  is a regularization parameter.

To make the transformation smooth w.r.t.  $X$ , we parameterize  $\mathbf{W}^{teL}, \mathbf{B}^{teL}, \mathbf{w}^{teL}, \mathbf{b}^{teL}$  using:  $\mathbf{W}^{teL} = R^{teL} \mathbf{G}, \mathbf{B}^{teL} = R^{teL} \mathbf{H}, \mathbf{w}^{teL} = R^{teL} \mathbf{g}, \mathbf{b}^{teL} = R^{teL} \mathbf{h}$ , where  $R^{teL} = L^{teL} (L^{teL} + \lambda I)^{-1}$ ,  $L^{teL} = K_{X^{teL} X^{teL}}$ . Following the same smoothness constraint we also have:  $\mathbf{w}^{te} = R^{te} \mathbf{g}$ , where  $R^{te} = K_{X^{te} X^{teL}} (L^{teL} + \lambda I)^{-1}$ . This parametrization results in the new objective:

$$\arg \min_{G, H, g, h} \|(R^{teL} \mathbf{g} \odot Y^{teL} + R^{teL} \mathbf{h}) - Y^{new(L)}\|^2 + \lambda_{reg} \|R^{te} \mathbf{g} - \mathbf{1}\|^2. \quad (2)$$

We use a Metropolis-Hasting algorithm to optimize the objective (Eq. 2) which is multi-modal due to the use of the Gaussian kernel. The proposal distribution is given by  $\theta^t \sim \mathcal{N}(\theta^{t-1}, \Sigma)$ , where  $\Sigma$  is a diagonal matrix with diagonal elements determined by the magnitude of  $\theta \in \{\mathbf{G}, \mathbf{H}, \mathbf{g}, \mathbf{h}\}$ . In addition, the transformation on  $X$  requires that the support of  $P(X^{new})$  is contained in the support

of  $P(X^{tr})$ , which might be hard to achieve on real data, especially when  $X$  has a high-dimensional feature space. To ensure that the training data can be better utilized, we relax the support-containing condition by enforcing an overlapping ratio between the transformed  $X^{new}$  and  $X^{tr}$ , i.e., we reject those proposal distributions which do not lead to a transformation that exceeds this ratio.

After obtaining  $\mathbf{G}, \mathbf{H}, \mathbf{g}, \mathbf{h}$ , we make predictions on  $X^{teU}$  by:

(1) Transform  $X^{teU}$  to  $X^{new(U)}$  with the optimized  $\mathbf{G}, \mathbf{H}$ :  $X^{new(U)} = \mathbf{W}^{teU} \odot X^{teU} + \mathbf{B}^{teU} = R^{teU} \mathbf{G} \odot X^{teU} + R^{teU} \mathbf{H}$ ; (2) Build a Gaussian Process on  $(X^{tr}, Y^{tr})$  and predict on  $X^{new(U)}$  to get  $Y^{new(U)}$ ; (3) Predict using optimized  $\mathbf{g}, \mathbf{h}$ :  $\hat{Y}^{teU} = (Y^{new(U)} - \mathbf{b}^{teU}) ./ \mathbf{w}^{teU} = (Y^{new(U)} - R^{teU} \mathbf{h}) ./ R^{teU} \mathbf{g}$ , where  $R^{teU} = K_{X^{teU} X^{teL}} (L^{teL} + \lambda I)^{-1}$ .

With the use of  $\mathbf{W} = R\mathbf{G}, \mathbf{B} = R\mathbf{H}, \mathbf{w} = R\mathbf{g}, \mathbf{b} = R\mathbf{h}$ , we allow more flexible transformations than mean-centering and variance-scaling while assuming that the transformations are smooth w.r.t  $X$ . We will illustrate the advantage of the proposed method in the experimental section.

### 3.3 A Kernel Mean Embedding Point of View

Under the kernel mean embedding point of view, it is easy to see that step (2) in the SMS approach is equivalent to estimating  $\hat{\mu}[P_{Y^{new(L)}}]$  using conditional embeddings [11] with a linear kernel on  $Y$ :  $\hat{\mu}[P_{Y^{new(L)}}] = \hat{U}[P_{Y^{tr}|X^{tr}}] \hat{\mu}[P_{X^{new(L)}}] = \psi(\mathbf{y}^{tr})(\phi(\mathbf{x}^{tr})^\top \phi(\mathbf{x}^{tr}) + \lambda I)^{-1} \phi^\top(\mathbf{x}^{tr}) \phi(\mathbf{x}^{new(L)}) = (K_{X^{new(L)} X^{tr}} (K_{X^{tr} X^{tr}} + \lambda I)^{-1} Y^{tr})^\top$ . In step (3) we want to find the optimal  $\mathbf{G}, \mathbf{H}, \mathbf{g}, \mathbf{h}$  such that the distributions on  $Y$  are matched across domains, i.e.,  $P_{Y^*} = P_{Y^{new(L)}}$ . The objective function Eq. 2 is effectively minimizing the maximum mean discrepancy:  $\|\hat{\mu}[P_{Y^*}] - \hat{\mu}[P_{Y^{new(L)}}]\|^2 = \|\hat{\mu}[P_{Y^*}] - \hat{U}[P_{Y^{tr}|X^{tr}}] \hat{\mu}[P_{X^{new(L)}}]\|^2$ , with a Gaussian kernel on  $X$  and a linear kernel on  $Y$ .

The transformation  $\{\mathbf{W}, \mathbf{B}, \mathbf{w}, \mathbf{b}\}$  are smooth w.r.t  $X$ . Take  $\mathbf{w}$  for example,  $\hat{\mu}[P_{\mathbf{w}}] = \hat{U}[P_{\mathbf{w}|X^{teL}}] \hat{\mu}[P_{X^{teL}}] = \varphi(\mathbf{g})(\phi^\top(\mathbf{x}^{teL}) \phi(\mathbf{x}^{teL}) + \lambda I)^{-1} \phi^\top(\mathbf{x}^{teL}) \phi(\mathbf{x}^{teL}) = \varphi(\mathbf{g})(L^{teL} + \lambda I)^{-1} L^{teL} = (R^{teL} \mathbf{g})^\top$ .

## 4 Experiments

**Synthetic Dataset.** We generate the synthetic data with (using matlab notation):  $X^{tr} = \text{randn}(80, 1)$ ,  $Y^{tr} = \sin(2X^{tr} + 1) + 0.1 * \text{randn}(80, 1)$ ;  $X^{te} = [w * \min(X^{tr}) + b : 0.03 : w * \max(X^{tr})/3 + b]$ ,  $Y^{te} = \sin(2(\text{rev}_w * X^{te} + \text{rev}_b)) + 1 + 2$ . The synthetic dataset used is with  $w = 0.5; b = 5; \text{rev}_w = 2; \text{rev}_b = -10$ , as shown in Fig. 1. We compare the SMS approach with the following approaches:

(1) **Only test x**: prediction using labeled test data only; (2) **Both x**: prediction using both the training data and labeled test data without transformation; (3) **Offset**: the offset approach [16]; (4) **DM**: the distribution matching approach [16]; (5) **KMM**: Kernel mean matching [9]; (6) **T/C shift**: Target/Conditional shift [12], code is from <http://people.tuebingen.mpg.de/kzhang/Code-TarS.zip>.

To ensure the fairness of comparison, we apply (3) to (6) using: the **original** data, the **mean-centered** data, and the mean-centered+variance-scaled (**mean-var-centered**) data.

A detailed comparison with different number of observed test points are shown in Fig. 4, averaged over 10 experiments. The selection of which test points to label is done uniformly at random for each experiment. The parameters are chosen by cross-validation. As we can see from the results, our proposed approach performs better than all other approaches.

As an example, the results for transfer learning with 5 labeled test points on the synthetic dataset are shown in Fig. 5. The 5 labeled test points are shown as filled blue circles. First, our proposed model, SMS, can successfully learn both the transformation on  $X$  and the transformation on  $Y$ , thus resulting in almost a perfect fit on unlabeled test points. Using either only labeled test points, or training+labeled test points, results in a poor fit towards the right part of the function because there are no observed test labels in that part. The DM/offset approach also results in a poor fit because simple variance-scaling does not yield a good match on  $P(Y|X)$ . The KMM approach, as mentioned before, applies the same conditional model  $P(Y|X)$  across domains, hence it does not perform well. The Target/Conditional Shift approach does not perform well either since it does not utilize any of the labeled test points. Its predicted support of  $P(Y^{te})$ , is constrained in the support of  $P(Y^{tr})$ , which results in a poor prediction of  $Y^{te}$  once there exists an offset between the  $Y$ 's.

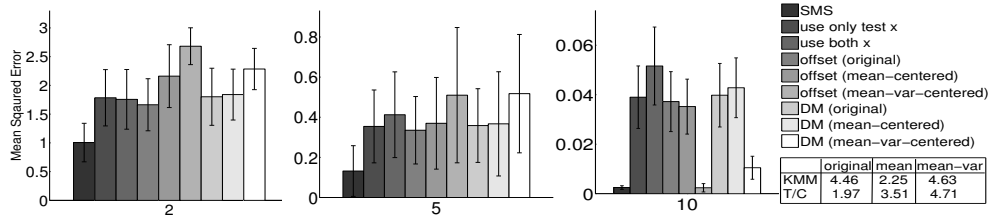


Figure 4: Comparison of MSE on the synthetic dataset with  $\{2, 5, 10\}$  labeled test points

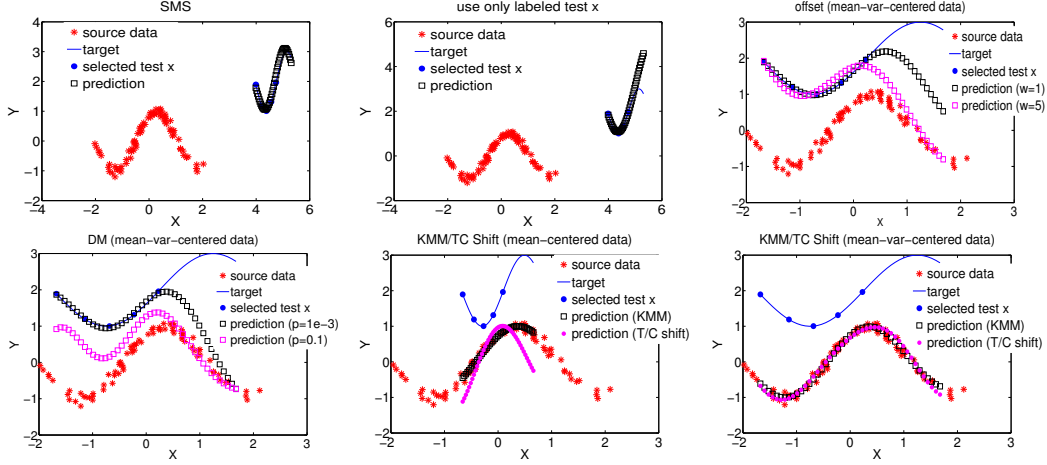


Figure 5: Comparison of results on the synthetic dataset: An example

**Real-world Dataset.** The two grape datasets we use are riesling (128 labeled images) and traminette (96 labeled images), as shown in Fig. 3. The goal is to transfer the model learned from one grape dataset to another. The results are shown in Table 1. In each row the result in bold indicates the result with the best RMSE (\* means statistically significant at a  $p = 0.05$  level with unpaired t-tests). We can see that our proposed algorithm yields better results under most cases, especially when the number of labeled test points is small.

Table 1: RMSE for transfer learning on real data

# $X^{teL}$	SMS	DM	Offset	Only test x	Both x	KMM	T/C Shift
5	<b>1197±23*</b>	1359±54	1303±39	1479±69	2094±60	2127	2330
10	<b>1046±35*</b>	1196±59	1234±53	1323±91	1939±41	2127	2330
15	<b>993±28</b>	1055±27	1063±30	1104±46	1916±36	2127	2330
20	<b>985±13</b>	1056±54	1024±20	1086±74	1832±46	2127	2330
30	960±19	<b>921±29</b>	961±30	937±29	1663±31	2127	2330
50	<b>893±16</b>	925±59	935±59	926±64	1558±51	2127	2330
70	860±40	805±38	819±40	<b>804±37</b>	1399±63	2127	2330
90	<b>791±98</b>	838±102	863±99	838±104	1288±117	2127	2330

## 5 Conclusion

In this paper, we proposed a transfer learning algorithm that handles both support and model shift. The algorithm transforms both  $X$  and  $Y$  by a location-scale shift, then the labels across domains are matched to learn both transformations. Since we allow more flexible transformations than mean-centering and variance-scaling, the proposed method yields better results than traditional methods.

## References

- [1] Wang, Xuezhi, and Schneider, Jeff. Flexible transfer learning under support and model shift. *NIPS*, 2014.
- [2] Oliva, Junier B., Neiswanger, Willie, Poczos, Barnabas, Schneider, Jeff, and Xing, Eric. Fast distribution to real regression. *AISTATS*, 2014.
- [3] Nuske, S., Gupta, K., Narasimhan, S., and Singh., S. Modeling and calibration visual yield estimates in vineyards. *International Conference on Field and Service Robotics*, 2012.
- [4] Mihalkova, Lilyana, Huynh, Tuyen, and Mooney., Raymond J. Mapping and revising markov logic networks for transfer learning. *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-2007)*, 2007.
- [5] Do, Cuong B and Ng, Andrew Y. Transfer learning for text classification. *Neural Information Processing Systems Foundation*, 2005.
- [6] Raina, Rajat, Ng, Andrew Y., and Koller, Daphne. Constructing informative priors using transfer learning. *Proceedings of the Twenty-third International Conference on Machine Learning*, 2006.
- [7] Niculescu-Mizil, Alexandru and Caruana, Rich. Inductive transfer for bayesian network structure learning. *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.
- [8] Shimodaira, Hidetoshi. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90 (2): 227-244, 2000.
- [9] Huang, Jiayuan, Smola, Alex, Gretton, Arthur, Borgwardt, Karsten, and Scholkopf, Bernhard. Correcting sample selection bias by unlabeled data. *NIPS 2007*, 2007.
- [10] Gretton, Arthur, Borgwardt, Karsten M., Rasch, Malte, Scholkopf, Bernhard, and Smola, Alex. A kernel method for the two-sample-problem. *NIPS 2007*, 2007.
- [11] Song, Le, Huang, Jonathan, Smola, Alex, and Fukumizu, Kenji. Hilbert space embeddings of conditional distributions with applications to dynamical systems. *ICML 2009*, 2009.
- [12] Zhang, Kun, Scholkopf, Bernhard, Muandet, Krikamol, and Wang, Zhikun. Domain adaptation under target and conditional shift. *ICML 2013*, 2013.
- [13] Jiang, J. and Zhai., C. Instance weighting for domain adaptation in nlp. *Proc. 45th Ann. Meeting of the Assoc. Computational Linguistics*, pp. 264-271, 2007.
- [14] Liao, X., Xue, Y., and Carin, L. Logistic regression with an auxiliary data source. *Proc. 21st Intl Conf. Machine Learning*, 2005.
- [15] Sun, Qian, Chattopadhyay, Rita, Panchanathan, Sethuraman, and Ye, Jieping. A two-stage weighting framework for multi-source domain adaptation. *NIPS*, 2011.
- [16] Wang, Xuezhi, Huang, Tzu-Kuo, and Schneider, Jeff. Active transfer learning under model shift. *ICML*, 2014.
- [17] Pan, Sinno Jialin and Yang, Qiang. A survey on transfer learning. *TKDE 2009*, 2009.
- [18] Seo, Sambu, Wallat, Marko, Graepel, Thore, and Obermayer, Klaus. Gaussian process regression: Active data selection and test point rejection. *IJCNN*, 2000.
- [19] Ji, Ming and Han, Jiawei. A variance minimization criterion to active learning on graphs. *AISTATS*, 2012.
- [20] Garnett, Roman, Krishnamurthy, Yamuna, Xiong, Xuehan, Schneider, Jeff, and Mann, Richard. Bayesian optimal active search and surveying. *ICML*, 2012.
- [21] Ma, Yifei, Garnett, Roman, and Schneider, Jeff. Sigma-optimality for active learning on gaussian random fields. *NIPS*, 2013.