Scalable Domain Adaptation via Intelligent Sampling

Aaditya Landge[1], Samira Daruki[2] and Shashank Krishnaswamy[3]

***Abstract*— In the area of deploying machine learning systems, Domain Adaptation is an important task which we encounter in real world. Here the goal is to build our model based on some fixed source domain and then deploy it to one or more different target domains. In many applications, it is expensive and time consuming to collect labeled training samples. On the other side, classifiers trained with only a limited number of labeled patterns are usually not robust. In practice, the computational cost for domain adaptation will grow fast as the data sets become larger and more unlabelled data is cheaply available.**

In this paper, we consider a semi-supervised domain adaptation technique named DTMKL(Domain Transfer Multiple Kernel Learning) which can learn robust classifiers with only a limited number of labeled patterns from the target domain by leveraging a large amount of labeled training data from other auxiliary(which we call source) domains. Under the framework of DTMKL, we propose an approach based on intelligent sampling on the unlabled data which reduces the running time without any significant impact on accuracy.

I. Introduction

Data mining and machine learning technologies have already achieved significant success in many knowledge engineering areas including classification, regression, and clustering. However, many machine learning methods work well only under a common assumption: the training and test data are drawn from the same feature space and the same distribution. When the distribution changes, most statistical models need to be rebuilt from scratch using newly collected training data. In many real-world applications, it is expensive or impossible to recollect the needed training data and rebuild the models. It would be nice to reduce the need and effort to recollect the training data. In such cases, knowledge transfer or transfer learning between task domains would be desirable.

In recent years, there has been growing research interest in developing new domain adaptation and transfer learning methods. These techniques have been successfully applied and used in the many real world applications, in which the domain of interest (the target domain) contains very few or even no labeled samples, while an existing domain (the auxiliary/source domain) is often available with a large number of labeled examples.

In domain adaptation settings, we have three possible scenarios: *supervised* which we have labeled data in both source and target, *unsupervised* which we have labeled data in only source and *semi-supervised* which we have both labeled and unlabeled data in target. In this work, we focus on the semi-supervised domain adaptation.

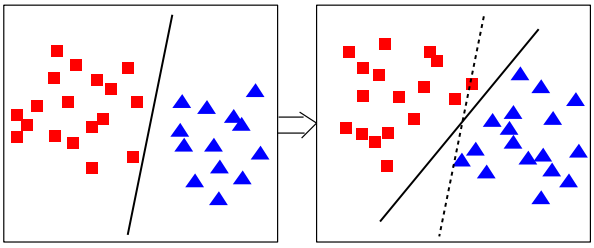

Fig. 1. Leveraging classifiers from one domain to obtain classifiers for the other domain

A motivating example Sentiment detection and classification has received considerable attention recently [15], [18], [9]. As a specific simple example, assume we have reviews for two types of products, for instance Books and DVDs, and the goal is to classify these reviews to two groups of positive and negative. These problems can be viewed as cross-domain adaptation which we leverage classifiers from one domain to obtain classifiers for the other one (See Figure 1).

While movie reviews have been the most studied domain, sentiment analysis has extended to a number of new domains, ranging from stock message boards to congressional floor debates. Research results have been deployed industrially in systems that study on market reaction and summarize opinion from web pages, discussion boards, and blogs. With such widely-varying domains, researchers and engineers who build sentiment classification systems need to collect and curate data for each new domain they encounter. The effort to annotate corpora for each domain may become prohibitive, especially since product features change over time. To deal with this problem, researchers came up with this idea to annotate corpora for a small number of domains, train classifiers on those corpora, and then apply them to other similar corpora. However there are two challenges with this approach: First, it is well known that trained classifiers lose accuracy when the test data distribution is significantly different from the training data distribution. Second, it is not clear which notion of domain similarity should be used to select domains to annotate that would be good proxies for many other domains.

Several domain adaptation techniques have been developed to address these problems efficiently. However, in this work we are not designing a new domain adaptation algorithm, instead we are seeking to improve the speed-up for one of the existing methods.

[1]Ph.D. student and researcher at the University of Utah, Scientific Computing and Imaging Institute. aaditya@sci.utah.edu

[2]Ph.D. student and researcher at the University of Utah, School of Computing, Theory and Algorithms Lab. daruki@cs.utah.edu

[3]Master student at the University of Utah, School of Computing, Joining Amazon as software developer. shash2k5@gmail.com

The contributions of this paper are mostly on the improving the speed up for training part in the domain adaptation process which is achieved by reducing the size of unlabeled points and taking the most informative points as the sample for algorithm input, while keeping the accuracy on the same level.

The rest of the paper is organized as follows: Section 2 gives a brief review about some related and previous work on different techniques for domain adaptation and speed-up approaches. Section 3, overview the basic notations, framework and state the problem definition. Then, section 4 introduces the main approach and techniques which we are applying. In particular, we present the two intelligent sampling methods based on the cluster entropy and base-classifier. Section 5 describes a series of experiments that validate the proposed approach on domain adaptation problem with real-world data. Finally, conclusive remarks are presented in Section 6.

II. Related Work

We consider and review the related work in two following categories: the prior work on different cross-domain adaptations techniques(which shows our specific interest in exploring DTMKL algorithm [8] among all the existing algorithms) and the prior work on how to speed up these techniques to scale up for massive data sets.

There is a significant amount of prior work on the speeding up the semi-supervised learning algorithms(mostly for only one domain), but these publications achieve their speed-ups through novel optimization methods, which are explicitly designed for some specific learning algorithms. In contrast, our work is mostly independent of the learning algorithm and it can be easily adapted to any learning technique.

In practice, cross-domain learning methods have been successfully used in many real-world applications, such as sentiment classification [2], natural language processing [7], text categorization [6], [13], information extraction [6], WiFi localization [13], and visual concept classification [11], [12], [20]. Recall that the feature distributions of training samples from different domains change tremendously, and the training samples from multiple sources also have very different statistical properties(such as mean, intraclass, and interclass variance). Though a large number of training data are available in the auxiliary domain, the classifiers trained from those data or the combined data from both the auxiliary and target domains may perform poorly on the test data from the target domain [11], [20].

To take advantage of all labeled patterns from both auxiliary and target domains, Daume III [7] proposed a so-called Feature Replication(FR) method to augment features for cross-domain learning. The augmented features are then used to construct a kernel function for Support Vector Machine(SVM) training. Yang et al. [20] proposed Adaptive SVM (A-SVM) for visual concept classification, in which the new SVM classifier $f^T(\mathbf{x})$ is adapted from an existing classifier $f^S(\mathbf{x})$ (referred to as source classifier) trained from the source domain. Cross-domain SVM (CD-SVM) proposed by Jiang et al. [11] used k-nearest neighbours from the target domain to define a weight for each auxiliary pattern, and then the SVM classifier was trained with the reweighted auxiliary patterns. More recently, Jiang et al. [12] proposed mining the relationship among different visual concepts for video concept detection. They first built a semantic graph and the graph can then be adapted in an online fashion to fit the new knowledge mined from the test data. However, all these methods [7], [11], [12], [19], [20] did not utilize unlabeled patterns from the target domain. Such unlabeled patterns can also be used to improve the classification performance [3], [21].

When there are only a few or even no labeled patterns available in the target domain, the auxiliary patterns or the unlabeled target patterns can be used to train the target classifier. Several cross-domain learning methods [10], [17] were proposed to cope with the inconsistency of data distributions(such as covariate shift [17] or sampling selection bias [10]). These methods reweighted the training samples from the auxiliary domain by using unlabeled data from the target domain such that the statistics of samples from both domains are matched. Very recently, Bruzzone and Marconcini [5] proposed Domain Adaptation Support Vector Machine(DASVM), which extended Transductive SVM(T-SVM) to label unlabeled target patterns progressively and simultaneously remove some auxiliary labeled patterns. Interested readers may refer to [14] for the more complete survey of cross-domain learning methods.

III. Preliminaries and Problem Definition

Throughout this paper, we denote the labeled and unlabeled data from the target domain respectively as D_ℓ^T and D_u^T, where $D^T = D_\ell^T \cup D_u^T$ shows the total data set in target domain. Similarly, we have $D^S = D_\ell^S \cup D_u^S$ for source domain.

In the following, we give a brief overview of kernel trick (which is the basic for many learning algorithm), DTMKL algorithm (which is introduced in detail in [8]) and soft clustering.

A. Kernel Functions

Kernel methods are a class of algorithms for pattern analysis, whose best known element is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations in general types of data. Kernel methods map the data into higher dimensional spaces in the hope that in this higher-dimensional space the data could become more easily separated or better structured. There are also no constraints on the form of this mapping, which could even lead to infinite-dimensional spaces. This mapping function, however, hardly needs to be computed because of a tool called the kernel trick. The kernel trick is a mathematical tool which can be applied to any algorithm which solely depends on the dot product between two vectors. Wherever a dot product is used, it is replaced by a kernel function. When properly applied, those

candidate linear algorithms are transformed into a non-linear algorithms.

In DTMKL method [8], base kernels are predetremined. We consider two types of base kernels: linear kernel ($k(x_i, x_j) = x_i' x_j$) and polynomial kernel ($k(x_i, x_j) = (x_i' x_j + 1)^a$), where we use different values for parameter $a = 1.5, 1.6, ..., 2.0$. Thus, we have in total seven base kernels in learning part, which results in seven base classifier.

B. Domain Transfer Multiple Kernel Learning(DTMKL)

DTMKL framework [8] is based on the SVM and prelearned classifiers. This method makes use of the labled target training data as well as the decision values from the existing base classifiers on the unlabeled data from the target domain. These base classifiers can be learned by using any method like SVM. DTMKL, as stated in [8], is the first semi-supervised cross-domain kernel learning framework for the single source domain problem which can incorporate many existing kernel methods. In fact, DTMKL is different from other traditional kernel learning methods since it does not assume that the training and test data are drawn from the same domain.

The main task of DTMKL is to learn the decision function for the target domain:

$$f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b = \Sigma_{i=1}^{n} \alpha_i k(\mathbf{x_i}, \mathbf{x}) + b \qquad (1)$$

as well as the kernel function k simultaneously, where $\mathbf{w}$ is the weight vector in the feature space and b is the bias term. Notice that α_is are the coefficients of the kernel expansion for the decision function $f(x)$ using Representer Theorem [16].

In practice, DTMKL follows two main objectives:

- Minimizing the distance between the data distributions of the source and target domains (Data distribution mismatch)
- Minimizing the structural risk functional of any kernel method

The learning framework of DTMKL is then formulated as

$$[k, f] = \arg\min_{k,f} \Omega(DIST_k^2(D^S, D^T)) + \theta R(k, f, D) \quad (2)$$

where $\Omega(.)$ is any monotonic increasing function and $\theta > 0$ is a tradeoff parameter to balance the mismatch between data distributions of two domains and the structural risk functional $R(k, f, D)$ defined on the labeled patterns.

Instead of learning a non-parametric kernel matrix K in (2) for cross-domain learning, DTMKL assumes the kernel k is a linear combination of a set of base kernels k_ms, namely,

$$k = \Sigma_{m=1}^{M} d_m k_m \qquad (3)$$

In this framework, the optimal kernel is learned by explicitly minimizing the distribution mismatch between the source and target domains by using both labeled and unlabeled patterns and utilizing the patterns from both source and target which results in a better classification performance.

More details about different versions of DTMKL and full algorithm can be found in [8] and we do not go through all of that again here. We just highlight main features and ideas about formulation of the algorithm. The sketch of the DTMKL framework is illustrated in Figure 2:

- To deal with the considerable change between feature distributions of different domains, DTMKL minimizes the structural risk functional and Maximum Mean Discrepancy(MMD)[4], a criterion to evaluate the distribution mismatch between the source and target domains. In practice, DTMKL provides a unified framework to simultaneously learn an optimal kernel function as well as a robust classifier. Moreover, it proposes a reduced gradient descent procedure to efficiently and effectively learn the linear combination coefficients of multiple base kernels as well as the target classifier.
- Many kernel learning methods such as SVM can be readily embedded into DTMKL framework to solve cross-domain learning problems.
- Kernel matrix is learned in an semi-supervised manner and as a result, by using the label information will be a more effective method.
- DTMKL simultaneously learns a kernel function and SVM classifier.
- In contrast to other cross-domain learning approaches, which are nonparametric and cannot be applied to unseen data, DTMKL can handle any new test data.
- The complexity of DTMKL algorithm is less than other cross-domain learning algorithms and it makes it more convenient and effective to be used in medium or large-scale real-world applications.

C. Soft Clustering

A clustering algorithm takes a set D of input data points and partitions them into k groups $C_1, ..., C_k$ of similar objects by minimizing some specific cost function based on the problem. We can interpret the clustering in two different ways: In *hard* clustering, the data is divided into distinct partitions and the output from clustering algorithm is an assignment function $f : D \to [1...k]$ which maps each point to *exactly* one of the groups C_i. In *soft* clustering the data points can belong to multiple groups and associated with each point is a vector of membership probabilities sum to one, which stands for the weights of assignment to each cluster. We can represent a soft clustering as a function $f : D \to \Delta^k$ which in $\Delta^k = \{(p_1, ..., p_k) | p_i \geq 0; \Sigma_i p_i = 1\}$.

IV. Approach

In this section we discuss about our main contribution in this work, which includes using the intelligent sampling ideas and adapt it to DTMKL algorithm to improve its efficiency for domain adaptation task.

To find the relevant unlabeled points from target domain for our sampling purpose, we apply three different sampling strategies as follows:

- Uniform sampling: Here all the points, regardless of position, will be picked with the same probability.

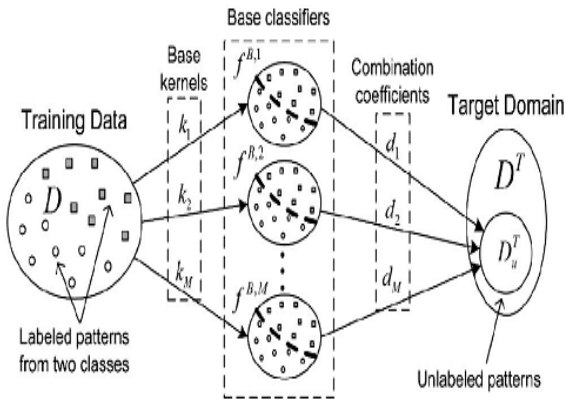

Fig. 2. [8] Illustration of virtual labels in DTMKL algorithm. The base classifier $f^{B,m}$ is learned with base kernel function k_m and the labeled training data from D, where $m = 1, ..., M$. For each of the unlabeled target pattern x from D_u^T, we can obtain its decision value $f^{B,m}(x)$ from each base classifier. Then, the virtual label $\hat{y}$ is defined as the linear combination of its decision values $f^{B,m}(x)$s weighted by the coefficients d_ms, i.e., $\hat{y} = \Sigma_{m=1}^{M} d_m f^{B,m}(x)$.

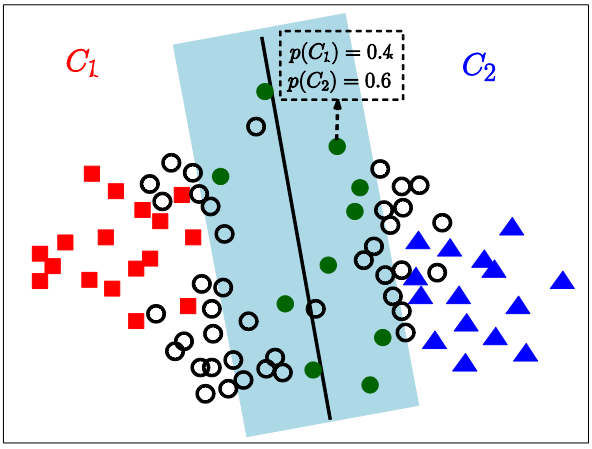

Fig. 3. Unstable points have maximum entropy score $H(x)$ and more likely are near decision boundary.

- Entropy-based sampling: Here the points will be picked with probability proportional to the cluster entropy score.
- Base-classifier based sampling: Here the points with decision probability values in some specific range will be picked.

In the following, we describe in detail the sampling approaches which perform in a more intelligent manner:

A. Entropy-based Sampling

In this approach, we are interested in identifying the unstable points which do not obtain a sharp cluster assignment and typically these points are the ones which lies close around the cluster boundary. We can find these points by making use of a soft clustering algorithms. Among all the several algorithms which exist for this purpose, we use the fuzzy k-means algorithm developed by Bezdek [1], which scales well with the size of unlabeled point set. For our purpose which is a binary classification task, we set the number of clusters $k = 2$.

The output of fuzzy k-mean algorithm is the vectors $\mathbf{p}(x_i) = [p_{i1}, p_{i2}]$ of assigned probabilities for every point x_i. Given these vectors, we can measure the uncertainty score of each point by computing the entropy of the resulted clustering distribution:

$$H(\mathbf{p}(x_i)) = -\Sigma_{j=1}^{k} p_{ij} \log(p_{ij})$$

Note that the points with higher uncertainty(which we call them *unstable* points) have the maximum values of entropy score. Some examples of these points are illustrated in Figure 3. Now we make use of these scores as weights and apply a standard acceptance/rejection sampling algorithm to pick a subset of the data points.

B. Base-classifier based Sampling

As explained before, each base classifier is learned with some base kernel function and the labeled training data from D. Each base classifier outputs the decision probability values for every unlabled point from the target domain. We pick all the points with decision values within some specific range $[0.5 - R, 0.5 + R]$ which identifies the unstable points with respect to each base classifier. Then we take the union of these samples for the final sample from target unlabeled set as illustrated in Figure 4.

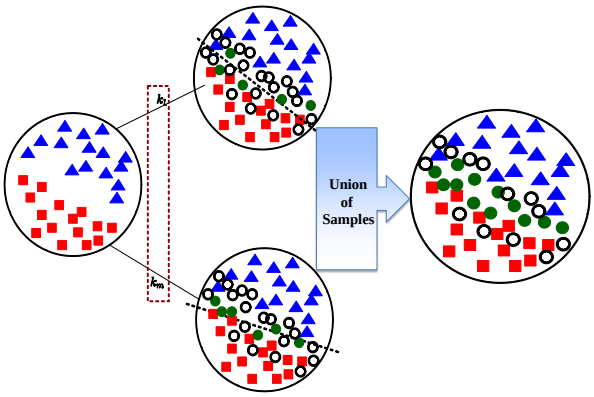

Fig. 4. Illustration of base-classifier based sampling.

V. Experimental Results

A. Datasets and Methodology

The data sets which we use include three Email spam datasets which are differentiated by three different users (denoted by User1, User2 and User3) from the email data set (available at: http://www.ecmlpkdd2006.org/challenge.html). Each of these subsets contains 2500 emails, in which half of them are *spam* and the other half are *nonspam*. Here our task is to apply a binary classification on the set of emails to find the group of spams and nonspams. Each email is represented

as word-frequency tokens. The data sets are in the format of 200000 dimensional but still sparse in nature. Note that since these sets of emails are annotated as spam and nonspam by different users, the distributions of three subsets will be related in some sense but the same time different and this makes it a good setting for applying the domain adaptation task.

For our experiments, we consider three settings between three users as source and target domains, which are illustrated in table 5. In each setting, we take all the labeled points from the source domain along with five negative and five positive random samples from the target domain to serve as the training data set. The remaining points in target domain will be used as the unlabeled training data and test data as well.

Experimental setting of the Email spam dataset used

Setting	Source Domain	Target Domain
1	User1 (U00)	User2 (U01)
2	User2 (U01)	User3 (U02)
3	User3 (U02)	User1 (U00)

Fig. 5. Experimental settings of the Email spam data set for cross-domain adaptation algorithm.

B. Performance

We implemented the sampling algorithms in Matlab and used the available Matlab code for DTMKL algorithm. The performance results in terms of running time and accuracy of classification is presented in Figures 6 and 7.

For entropy-based sampling, we take 10 percent of the unlabled data in target domain as the input sample for DTMKL and for base-classifier based sampling we take $R = 0.2$ and pick the unlabled points with probability decision values within range $[0.3, 0.7]$ as the sample.

As you can see we achieve over 3x speed-ups with little impact on accuracy and we believe that with more efficient implementation of algorithms we can even get better results in speed. However, we observe that uniform sampling also gives some close result to other sampling methods. At this moment, we do not have a strong reason about this, but we claim that probably it is related to the inherent distribution of current data set which we are using in our experiments. For future work, we are planing to apply our algorithm on different data sets and explore the effect of our approach.

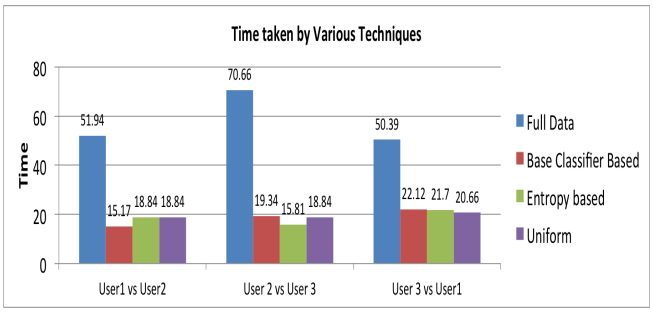

Fig. 6. Effect of sampling techniques on speed-up for DTMKL algorithm.

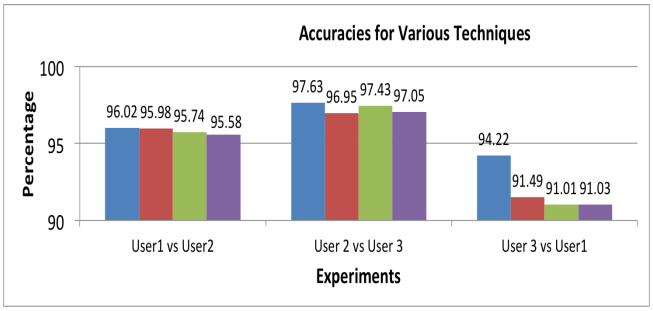

Fig. 7. Effect of sampling techniques on accuracy for DTMKL algorithm.

This work is an extension to some similar work which is done in Theory and Algorithm lab under supervision of Dr. Suresh Venkatasubramanian. They have proposed some adaptive subsampling methods and studied the effect on the Transductive SVM (which is a semi-supervised learning algorithm which is used for classification task on only one domain) and is submitted to ICML and currently is under review. We hope that with extending our experiments and results to other Domain Adaptation algorithms and considering more parameters in scoring function for sampling process, we can claim a strong statement about the effectiveness of intelligent sampling for large scale semi-supervised learning algorithms.

VI. Discussion and Conclusion

In this work, we have proposed a scalable domain adaptation approach based on intelligent sampling. We apply our sampling technique to some existing algorithm, Domain Transfer Multiple Kernel Learning (DTMKL), which is introduced in [8] as the first semi-supervised cross-domain kernel learning framework. Our approaches identify the most relevant and informative unlabled points to pick as an effective input sample for domain adaptation algorithm. This is achieved by utilizing cluster entropy and also uncertainty information resulted from the base-kernel classifiers. These two parameters help us to predict the regions of importance to the DTMKL algorithm and reduce the size of the unlabeled data by ignoring the other regions, while maintaining the accuracy.

Although our approach is based on DTMKL as a specific algorithm, it can be generalized and applied to other domain adaptation algorithms as well.

VII. Acknowledgement

We thank Parasaran Raman at Theory and Algorithms lab for proposing this research problem and helpful discussions throughout the course of this work, and Dustin Web for useful advice at the experimental stage for improving the performance results. Finally we are thankful to Dr. Van der berg for giving us this wonderful opportunity to work on some open-ended research project by taking Machine Learning course.

References

[1] James C Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, 1981.

[2] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007.

[3] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[4] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[5] Lorenzo Bruzzone and Mattia Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):770–787, 2010.

[6] Bo Chen, Wai Lam, Ivor Tsang, and Tak-Lam Wong. Extracting discriminative concepts for domain adaptation in text mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 179–188. ACM, 2009.

[7] Hal Daumé III. Frustratingly easy domain adaptation. In *ACL*, volume 1785, page 1787, 2007.

[8] Lixin Duan, Ivor W Tsang, and Dong Xu. Domain transfer multiple kernel learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):465–479, 2012.

[9] Andrew B Goldberg and Xiaojin Zhu. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pages 45–52. Association for Computational Linguistics, 2006.

[10] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2006.

[11] Wei Jiang, Eric Zavesky, Shih-Fu Chang, and Alex Loui. Cross-domain learning methods for high-level visual concept classification. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 161–164. IEEE, 2008.

[12] Yu-Gang Jiang, Jun Wang, Shih-Fu Chang, and Chong-Wah Ngo. Domain adaptive semantic diffusion for large scale context-based video annotation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1420–1427. IEEE, 2009.

[13] Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008.

[14] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[15] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[16] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels*. The MIT Press, 2002.

[17] Masashi Sugiyama and Amos J Storkey. Mixture regression for covariate shift. In *Advances in Neural Information Processing Systems*, pages 1337–1344, 2006.

[18] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[19] Pengcheng Wu and Thomas G Dietterich. Improving svm accuracy by training on auxiliary data sources. In *Proceedings of the twenty-first international conference on Machine learning*, page 110. ACM, 2004.

[20] Jun Yang, Rong Yan, and Alexander G Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th international conference on Multimedia*, pages 188–197. ACM, 2007.

[21] Xiaojin Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2:3, 2006.