Inf Retrieval (2012) 15:373–390
DOI 10.1007/s10791-012-9196-x

INFORMATION RETRIEVAL FOR SOCIAL MEDIA

# Sentiment detection with auxiliary data

Dan Zhang · Luo Si · Vernon J. Rego

Received: 15 May 2011 / Accepted: 27 February 2012 / Published online: 21 March 2012

**Abstract** As an important application in text mining and social media, sentiment detection has aroused more and more research interests, due to the expanding volume of available online information such as microblogging messages and review comments. Many machine learning methods have been proposed for sentiment detection. As a branch of machine learning, transfer learning is an important technique that tries to transfer knowledge from one domain to another one. When applied to sentiment detection, existing transfer learning methods employ articles with human labeled sentiments from other domains to help the sentiment detection on a target domain. Although most existing transfer learning methods are devoted to handle the data distribution difference between different domains, they only resort to some approximation methods, which may introduce some unnecessary biases. Furthermore, the popular assumption of existing transfer learning techniques on conditional probability is often too strong for practical applications. In this paper, we propose a novel method to model the distribution difference between different domains in sentiment detection by directly modeling the underlying joint distributions for different domains. Some of the important properties of the proposed method, such as the convergence rate and time complexity, are analyzed. The experimental results on the product review dataset and the twitter dataset demonstrate the advantages of the proposed method over the state-of-the-art methods.

**Keywords** Sentiment detection · Social media · Twitter · Microblogging · Transfer learning

D. Zhang (✉) · L. Si · V. J. Rego
Department of Computer Science, Purdue University, West Lafayette, IN, USA
e-mail: danzhang2008@gmail.com

L. Si
e-mail: lsi@cs.purdue.edu

V. J. Rego
e-mail: rego@cs.purdue.edu

## 1 Introduction

Over the past decade, with the development of the Internet and the social media, enormous opportunities have been created for companies of all sizes to interact with customers, advertise products, make business transactions, as well as for individuals to get better knowledge on the product reviews (Blitzer et al. 2012; Das and Chen 2007; Thomas et al. 2006). As a result, sentiment detection methods are becoming more and more important in automatically analyzing and summarizing the sentiments online.

With widely-varying domains, researchers who build sentiment classification systems need to collect and curate data for the domains they work on. But laboring the collected data is normally very expensive and inefficient. To solve the label insufficiency problem, a feasible way is to utilize examples from other domains. According to the availability of labels/sentiments on different domains, there are mainly three scenarios in sentiment detection: (1) Some labeled (with known sentiments) examples from other domains (source domain) are available, while no labeled example is available on the new domain (target domain). In this case, the sentiments of the examples in the new domain can be predicted by using domain adaption methods (Blitzer et al. 2012; Huang et al. 2006; Pan et al. 2010). (2) Sufficient unlabeled examples in other domains are available, while a small number of labeled examples in the new domain can be obtained. Then, a classifier can be trained through self taught learning (Raina et al. 2007). (3) Abundant labeled examples in the source domain are available, and meanwhile several labeled examples can also be obtained in the target domain. In this scenario, the classifier can be trained by treating the source domain examples as auxiliary data, and training a classifier that is consistent on both the source domain and the target domain.

In this paper, our focus is on the third case. In this case, it is clear that if the data distributions for both the source and the target domains are the same, then a classifier can be directly trained based on the labeled examples from both the source and target domains by using methods, such as support vector machines (SVM) (Scholkopf and Smola 2002). However, in practice, the distributions of these two domains are normally very different and directly applying the classifiers from other domains on the target domain normally leads to a poor classification performance (Blum and Chawla 2001; Lafferty et al. 2001; Nigam et al. 2000).

In twitter sentiment classification problem, normally some labeled tweets are available in the target domain, while in the source domain a rich set of labeled “complete“ documents are available. Since each tweet contains only a limited number of characters, directly designing classifiers based on the labeled tweets will not be accurate due to the extreme sparseness of these tweet feature vectors. So, we consider to incorporate the examples in the source domain. But the distribution for the source domain examples is usually deviates a lot from that of tweets, since they may probably cover different topics and the huge difference of the feature sparseness on these two domains also poses a big challenge. So, if we simply merge the labeled source and target domain examples together, and design a classifier based on them, the sentiment classification results will be badly affected.

As another example, in the sentiment detection of product review comments, suppose we have some labeled review comments in the target domain, as well as a lot of labeled ones in the source domain. A natural question is whether we can use the labeled review comments for some other products to help us to understand the review comments on the target product. Normally, different products have different characteristics. For example, “sharpness” is a good descriptor for a good knife. But it is not a good evaluation for laptops. So, if we design a sentiment classifier based on the labeled “knife” products, we cannot directly use it to classify the sentiments for the laptop review comments.

To deal with the domain difference problem, a lot of methods have already been developed. One of the most successful methods is the structural correspondence learning (SCL) (Ando and Bartlett 2005; Blitzer et al. 2006). In this work, the authors first find a set of pivot features which frequently appear in both the source domain and the target domain. Then, the correlations between the pivot features and the non-pivot features are modeled through a set of linear classifiers. Some hidden patterns underlying these classifiers are then considered as the correlations between different kinds of features. Based on these hidden patterns, another set of features are designed and appended to the original feature space before the supervised training process. In Huang et al. (2006), the authors assume that the posterior probability for the two domains are the same, and the difference only lies on the data distribution without considering the labels. Based on this assumption, they modeled data distribution difference between different domains through kernel mean matching.

Their methods are reasonable and handle the distribution difference problem from different perspectives. However, the distribution difference problem still exists in these works. For example, in SCL, even if the new features are appended to the original feature space, it is clear that distribution difference problem still exists on the original features. Therefore, designing a classifier based on the new feature space is still not good for target domain examples. In Huang et al. (2006), the assumption that the conditional probability of $P_S(y|\mathbf{x})$ and $P_T(y|\mathbf{x})$, where $\mathbf{x}$ represents the instance and $y$ is referred to as the label, are the same is too strong. Instead, in this paper, we propose a novel formulation—sentiment detection with auxiliary data (SDAD), which solves this problem by modeling the joint distribution difference between different domains through Kernel Density Estimation (KDE) (Bishop 2007) and incorporates the source domain examples more naturally into the objective function through reweighting the source domain examples. The proposed formulation is then solved by the bundle method (Smola et al. 2008; Teo et al. 2010). Some important properties of the proposed method, such as the convergence rate and the time complexity, are analyzed in "Appendix". The experimental results clearly demonstrate the advantages of the proposed method.

The rest of this paper is organized as follows: Sect. 2 introduces the related works. Section 3 gives the problem statement and puts forward the proposed method. An extensive set of experiments are given in Sect. 4 At the end of this paper, a conclusion will be drawn.

## 2 Related works

### 2.1 Sentiment detection

With more than 10 years' development, sentiment detection (Argamon et al. 1998; Kessler et al. 1997; Spertus 1997) has become one of the major subfields in information management (Dimitrova et al. 2002; Hillard et al. 2003; Wilson et al. 2005), especially after the year 2001. This is mainly due to three reasons (Pang and Lee 2008): (1) the increase of machine learning techniques in natural langauge processing; (2) the availability of the datasets due to the popularity of the Internet, especially the development of social media; (3) the rising interest in commercial and business intelligence applications in this area. As a result, a lot of approaches (Cardie et al. 2003; Das and Chen 2001; Morinaga et al. 2002; Pang and Lee 2004) have been developed to solve this problem.

In machine learning, sentiment detection can be viewed as a classification or regression problem, which mainly deals with two subproblems, i.e., sentiment polarity/classification and degrees of positivity. Depending on the domains on which training examples are available, the concrete methods can be categorized into two groups, i.e., the group that deals with only one single domain and the group with multiple domains. As for the first group (Taboada et al. 2006; Whitelaw et al. 2005; Wiebe and Riloff 2005), to train the classifiers, the training examples are normally available on the target domain. Some machine learning methods that have been used to train the classifiers and have shown state-of-the-art performances in these tasks include naive bayes, maximum entropy, support vector machine (SVM) (Pang et al. 2002) etc.

The second group, which is also the focus of this paper, considers training examples from several different domains. However, sentiment detection is a very domain specific problem, i.e., the classifiers trained in one domain do not perform well in others (Blum and Chawla 2001; Lafferty et al. 2001; Nigam et al. 2000). This is mainly because that the data distributions on different domains are usually different. For example, "sharpness" is a good word feature to describe knives, but is not a good one to evaluate computer products. To deal with this problem, one common way is to use the transfer learning (Pan and Yang 2010), which transfers the knowledge from the training examples in other domains (source domain) to the target domain.

However, previous transfer learning methods that have been applied to sentiment detection only deal with the data distributions problem implicitly. For example, in Blitzer et al. (2012), the authors picked some pivot features which appear frequently in both the source domain and the target domain, and then models the correspondences between these pivot features and all the other features. These correlations are considered as some new features in the training process (Ando and Bartlett 2005; Blitzer et al. 2006). Their method is reasonable. However, although being alleviated, the problem of distribution difference still exist, since they only append some additional features into the original feature space. And the performances are affected by the choices of the pivot features. In this paper, we model the joint distribution difference between the target domain and the source domain by kernel density estimation, so that the training examples on the source domain can be better utilized and the problem of picking the pivot features can also be avoided.

### 2.2 Transfer learning

In traditional machine learning, such as supervised learning (Duda et al. 2001) and semi-supervised learning (Zhu 2006), one of the common assumptions is that both the labeled and unlabeled data are sampled from the same distribution or lie on the same manifold. But when the distribution changes, a new model would need to be built. It would be useful if the previously trained models can be reused to guide the construction of the new model. This gives rise to the concept of transfer learning, a technique that transfers knowledge across domains, tasks and distributions that are similar but not the same.

An important problem in transfer learning is what kind of knowledge can actually be transferred from the source domain to the target domain. Roughly speaking, the assumptions introduced in previous transfer learning work can be grouped into four categories: (Pan and Yang 2010):

- Feature Representation Transfer. In Argyriou et al. (2007), Dai et al. (2008), Duan et al. (2009), Pan et al. (2012), Raina et al. (2007), and Zhang and Si (2009), the authors assume that there exist some common feature space shared by both the source domain examples and the target domain examples, and this common feature space can be used as a bridge to transfer knowledge from the source domain to the target domain.

- Parameter Transfer. By assuming the shared parameters/hyper-parameters, such as in Gaussian Process (GP)models, in Bonilla et al. (2008) and Lawrence and Platt (2004), the authors try to justify and estimate the shared parameters for the models in the source domain and the target domain.
- Instance Transfer. The examples in the source domain examples are selected or reweighted for use in the target domain (Dai et al. 2007; Huang et al. 2006).
- Relation Transfer. In Davis and Domingos (2009), Mihalkova et al. (2007), and Mihalkova and Mooney (2008), the authors build the relational map between the source and target domains, and relax the i.i.d. assumptions in these two domains.

In this paper, the proposed method is based on the instance transfer, which considers modeling the distribution difference between the examples on the source domain and target domain together through reweighting the importances of the labeled examples on the source domain. It is true that, some previous works, such as Huang et al. (2006), are also devoted to model the difference between different domains. However, they only achieve this goal indirectly by some approximation methods, while the proposed method directly models the distribution difference through kernel density estimation. Furthermore, to simplify the proposed formulation, the previous works assume that the conditional probability $P_S(y|\mathbf{x})$ and $P_T(y|\mathbf{x})$, where $\mathbf{x}$ are the same for both the source domain and the target domain. Instead, in this work, by taking advantage of kernel density estimation, we can avoid this assumption elegantly.

### 2.3 Training with auxiliary data

As a machine learning technology, SVM has enjoyed its popularity for more than ten years. One question related to SVM, as well as some other supervised learning methods, is how we can utilize the training examples from some other sources to improve the classification accuracy on the target domain. In Wu and Dietterich (2004), the authors proposed a novel formulation to incorporate the source domain examples into the training process as follows:

$$\min_h \sum_i^{N^p} L(h(\mathbf{x}_i^p), y_i^p) + \gamma \sum_i^{N^a} L(h(\mathbf{x}_i^a), y_i^a) + \lambda D(h), \tag{1}$$

where $h$ is the classification. $(\mathbf{x}_i^p, y_i^p)$ denotes the $i$th training example on the target domain, and $(\mathbf{x}_i^a, y_i^a)$ refers to the $i$th auxiliary example. $L(\bullet, \bullet)$ is a predefined loss function, such as the hinge loss. $D(h)$ is a complexity penalty to prevent overfitting. $\gamma$ and $\lambda$ are two trade-off parameters.

The problem with this method is that it incorporates the auxiliary data into the objective function without considering the distribution difference between the different domains. As suggested by Huang et al. (2006), by modeling the training data distribution difference between different domains, the model can be much more accurate. Out of the same motivation, in this paper, we model the distribution difference between different sentiment detection domains and combine them in a more natural way.

### 2.4 Bundle method

The proposed formulation is a convex optimization problem. In this paper, we proposed an optimization algorithm based on the bundle method (Smola et al. 2008; Teo et al. 2010), which has shown its superior performances in both efficiency and effectiveness over state-of-the-art methods, to solve this proposed formulation. The basic motivation of the bundle method is to approximate the objective function $J(\mathbf{w})$ through a set of linear functions, where $\mathbf{w}$ is the model parameter. In particular, this objective function is lower bounded as follows:

$$J(\mathbf{w}) \geq \max_{1 \leq i \leq t} \{J(\mathbf{w}_{i-1}) + \langle \mathbf{w} - \mathbf{w}_{i-1}, \mathbf{a}_i \rangle\},$$

where $\mathbf{w}_i$ is a set of points picked by the bundle method, and $\mathbf{a}_i$ is the gradient/sub-gradient at point $\mathbf{w}_i$. The bundle method monotonically decreases the gap between $J(\mathbf{w})$ and $\max_{1 \leq i \leq t}\{J(\mathbf{w}_{i-1}) + \langle \mathbf{w} - \mathbf{w}_{i-1}, \mathbf{a}_i \rangle\}$ such that the minimal point of $J(\mathbf{w})$ can be approximated by the minimum of the line segments $\max_{1 \leq i \leq t}\{J(\mathbf{w}_{i-1}) + \langle \mathbf{w} - \mathbf{w}_{i-1}, \mathbf{a}_i \rangle\}$.

Some recent development in bundle method (Teo et al. 2010) shows that if $J(\mathbf{w})$ contains some regularizers by itself, the bundle method is guaranteed to converge to the precision $\epsilon$ in $O(1/\epsilon)$ steps. In this paper, we adapt the bundle method to solve the proposed problem, which can also be proven to have an efficient convergence rate.

## 3 Sentiment detection with auxiliary data

In this section, we first introduce the problem of SDAD. Then, an optimization formulation is proposed, which integrates the source domain examples (i.e., auxiliary data) into the objective function in a principled way. This problem is later solved by bundle method (Smola et al. 2008; Teo et al. 2010). In "Appendix", we analyze some important properties of the proposed method, such as the convergence rate.

### 3.1 Problem statement

In the proposed problem, we have labeled data from both the source domain and the target domain, where the source domain examples are denoted as: $(\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ and the target domain examples are referred to as $(\mathbf{Z}, \mathbf{Y}^*) = \{(\mathbf{z}_1, y_1^*), (\mathbf{z}_2, y_2^*), \ldots, (\mathbf{z}_m, y_m^*)\}$, where $y_i \in \{1, -1\}$ and $y_i^* \in \{1, -1\}$ represent the positive and negative attitudes on source and target domains respectively. Without loss of generality, our objective is to train a linear sentiment classifier $\mathbf{w}$ based on the labeled examples from both the source domain and the target domain.

### 3.2 Methodology

#### *3.2.1 Formulation*

In this subsection, we propose the formulation of SDAD, which incorporates examples from different domains by modeling the data distribution difference. In particular, the optimization problem of SDAD can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{w}, \xi_i \geq 0, \xi_j^* \geq 0} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C_1}{n}\sum_{i=1}^{n} \xi_i + \frac{C_2}{m}\sum_{j=1}^{m} \beta_j \xi_j^* \\ s.t. \quad & \forall i \in \{1, 2, \ldots, n\}, \quad y_i \mathbf{w}^T \mathbf{x}_i \geq 1 - \xi_i, \\ & \forall j \in \{1, 2, \ldots, m\}, \quad y_j^* \mathbf{w}^T \mathbf{z}_j \geq 1 - \xi_j^*. \end{aligned} \tag{2}$$

where $\beta_j$ refers to $\frac{Pr_T(\mathbf{x}_j, y_j^*)}{Pr_S(\mathbf{x}_j, y_j^*)}$, and represents the ratio between the joint data distribution on the target domain $Pr_T(\mathbf{x}_j, y_j^*)$ and that on the source domain $Pr_S(\mathbf{x}_j, y_j^*)$ for the source domain example $\mathbf{x}_j$.

There are various ways to estimate $\beta_j$, such as Gaussian Mixture Model (GMM) (Liu et al. 2002), kernel density estimation (Sheather and Jones 1991), kernel mean matching (Huang et al. 2006), etc. In our approach, without loss of generality, we adopt kernel density estimation. In particular, we have:

$$\beta_j = \frac{Pr_T(\mathbf{x}_j, y_j^*)}{Pr_S(\mathbf{x}_j, y_j^*)} = \frac{Pr_T(\mathbf{x}_j|y_j^*) \times P_T(y_j^*)}{Pr_S(\mathbf{x}_j|y_j^*) \times P_S(y_j^*)}. \tag{3}$$

It is clear that $\frac{P_T(y_j^*)}{P_S(y_j^*)}$ represents the label ratios on the two different domains, which can be estimated from the labeled examples on both domains. As for $\frac{Pr_T(\mathbf{x}_j|y_j^*)}{Pr_S(\mathbf{x}_j|y_j^*)}$, by using kernel density estimation with the gaussian kernel, it can be estimated as follows:

$$\frac{Pr_T(\mathbf{x}_j|y_j^*)}{Pr_S(\mathbf{x}_j|y_j^*)} \propto \frac{\sum_{i=1}^{m} I_{ij}^* exp(-\frac{\|\mathbf{x}_j - \mathbf{z}_i\|}{\sigma^2})}{\sum_{k=1}^{n} I_{kj} exp(-\frac{\|\mathbf{x}_j - \mathbf{x}_k\|}{\sigma^2}) - 1}, \tag{4}$$

where $\sigma$ is the bandwidth parameter for the gaussian kernel. $I_{ij}^*$ is an indication function, which equals 1 if $y_i^*$ equals $y_j$, and otherwise zero. Similarly, $I_{kj}$ is an indication function, which equals 1 if $y_k$ equals $y_j$, and otherwise zero. It is clear that if a source domain example is close enough to the target domain examples, then its importance is higher. Otherwise, it will be down weighted. Through this way, the data distribution of the training examples on the source domain is adjusted to follow the data distribution on the target domain as close as possible.

Some of the previous transfer learning works also share similar motivations as the one we are using here. However, they only do this indirectly, by calculating the probability ratios in some other ways, such as kernel mean matching. Furthermore, in these works, to ease the complexity of the formulations, one common assumption for these previous works is that the conditional probability $P_T(y_j^*|\mathbf{x}_j)$ and $P_S(y_j^*|\mathbf{x}_j)$ are the same and the difference between different domains only lies on their data distributions without considering the labels, which is too strong in most cases. In this paper, instead, we model the joint probability ratio directly through kernel density estimation. It effectively avoids the strong assumption on the conditional probability and directly models the distributions on the two domains, rather than approximate them in an implicit way.

### *3.2.2 Efficient optimization*

There are several alternatives to solve problem (2) efficiently. Here, an efficient way, which is an adaption of the bundle method, is used to solve it. The concrete procedure is described in Table 1. Here, $R_{emp}(\mathbf{w}) = \frac{C_1}{n}\sum_{i=1}^{n} \max\{0, 1 - y_i\mathbf{w}^T\mathbf{x}_i\} + \frac{C_2}{m}\sum_{j=1}^{m} \beta_j \max\{0, 1 - y_j^*\mathbf{w}^T\mathbf{z}_j\}$, $J_t(\mathbf{w}_t) = \frac{1}{2}\|\mathbf{w}\|^2 + \max_{1 \le i \le t}\langle \mathbf{w}, \mathbf{a}_i\rangle + b_i$. Since $R_{emp}(\mathbf{w})$ is non-smooth, when calculating its gradient, we use the subgradient instead, which can be calculated as:

$$\partial_{\mathbf{w}} R_{emp}(\mathbf{w}) = -\frac{C_1}{n}\sum_{i=1}^{n} I_i^S y_i \mathbf{x}_i - \frac{C_2}{m}\sum_{j=1}^{m} I_j^T \beta_j y_j^* \mathbf{z}_i, \tag{5}$$

**Table 1** Algorithm Description: SDAD

**Algorithm**: Sentiment Detection with Auxiliary Data (SDAD)

```
Input:
1. Kernel density parameter: $\sigma$ in Eq.(3).
2. Optimization parameters: trade-off parameters $C_1$, and $C_2$in Eq.(2), optimization precision $\epsilon = 0.01$.
3. Source Domain Examples: $(\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$.
4. Target Domain Examples: $(\mathbf{Z}, \mathbf{Y}^*) = \{(\mathbf{z}_1, y_1^*), (\mathbf{z}_2, y_2^*), \ldots, (\mathbf{z}_m, y_m^*)\}$.
Output: classifier $\mathbf{w}$
1. Calculate $\beta_j$ according to Eq.(3).
2. Initialization $t = 0$, randomly initialize $\mathbf{w}_0$.
3. repeat
4.     $t = t + 1$
5.     Compute the gradient for the empirical loss: $\mathbf{a}_t = \partial_{\mathbf{w}} R_{emp}(\mathbf{w}_{t\text{-}1})$, and $b_t = R_{emp}(\mathbf{w}_{t\text{-}1}) - <\mathbf{w}_{t\text{-}1}, \mathbf{a}_t>$ .
6.     Derive the optimization problem: $R_t^{CP} = \max_{1 \le i \le t} < \mathbf{w}, \mathbf{a}_i > + b_i$
7.     $\mathbf{w}_t = \arg\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + R_t^{CP}$
8.     $\epsilon_t = \min_{0 \le i \le t} J(\mathbf{w}_i) - J_t(\mathbf{w}_t)$
9. until $\epsilon_t \le \epsilon$
10. Output: $\mathbf{w} = \mathbf{w}_t$. For a test example $\mathbf{z}$ in the target domain, if $\mathbf{w}^T\mathbf{z} > 0$, then it is labeled as a positive sentiment. Otherwise, it is labeled as negative.
```

where $I_i^S$ equals 1 if $y_i \mathbf{w}^T \mathbf{x}_i \le 1$, and otherwise 0. Similarly, $I_j^T$ equals 1 if $y_j^* \mathbf{w}^T \mathbf{z}_j \le 1$, and 0 otherwise. Some important properties of the proposed method will be elaborated in "Appendix".

## 4 Experiments

In this section, we present an extensive set of experimental results to demonstrate the advantages of the proposed method.

### 4.1 Datasets

We use two datasets in our experiments, i.e., the product review dataset, and the twitter dataset.

*Product Review Dataset* This is a benchmark dataset for sentiment detection (Blitzer et al. 2012), which is selected from the Amazon product reviews for four different product types: books, DVDs, electronics, and kitchen appliances. Since each review consists of a 0–5 stars rating, reviews with ratings higher than 3 points are considered as positive sentiment, and otherwise are considered as negative. Each dataset contains 2,000 reviews, among which 1,000 are positive reviews and the remaining 1,000 reviews are negative reviews. The detailed description of this dataset can be found in Table 2.

*Twitter Dataset* This dataset contains tweets downloaded and labeled throughout the whole October, 2010. The tweets with keywords "software" and "education" are used in this dataset. When extracting features, we use the same feature space as we use for the product review dataset. For more details, please refer to Table 2.

**Table 2** Dataset description

| Dataset | Sub-dataset | # Positive Inst | # Negative Inst | # Dim |
|---|---|---|---|---|
| Product Review | DVDs | 1,000 | 1,000 | 6,844 |
| | Kitchen | 1,000 | 1,000 | 6,844 |
| | Electronics | 1,000 | 1,000 | 6,844 |
| | Books | 1,000 | 1,000 | 6,844 |
| Twitter | Software | 2,353 | 673 | 6,844 |
| | Education | 969 | 619 | 6,844 |

The feature spaces of both the product review dataset and the twitter dataset are the same

**Table 3** Task description

| Task # | Source domain | Target domain | Task # | Source domain | Target domain |
|---|---|---|---|---|---|
| 1 | Electronics | Kitchen | 8 | DVD | Kitchen |
| 2 | Electronics | DVD | 9 | DVD | Electronics |
| 3 | Kitchen | Electronics | 10 | Book | Electronics |
| 4 | DVD | Book | 11 | Book | DVD |
| 5 | Electronics | Book | 12 | Book | Kitchen |
| 6 | Kitchen | Book | 13 | Electronics | Software |
| 7 | Kitchen | DVD | 14 | Book | Education |

The first twelve tasks are conducted on the product review dataset, while the remaining two tasks are conducted on both the product review dataset and the twitter dataset. Since twitter is not a reliable information source, limited by the length of each post, they could not be used as source domain datasets

For these datasets, the tf-idf (normalized term frequency and log inverse document frequency) (Manning et al. 2008) features are extracted, and the stop words are removed. We use porter as the stemmer. Given these two datasets, 14 sentiment detection tasks are created by specifying different combinations of source domain and task domain subdatasets. The detailed descriptions of these 14 tasks are specified in Table 3.

## 4.2 Methods

We compare the proposed method with the following competitors: SVM on the Target domain (SVMT); SVM on both the Source domain and the Target domain (SVMST); SCL on the Target Domain (SCLT)[1]; SCL on both the Source domain and the Target domain (SCLST). In our experiments, we show that the proposed method can also be combined with SCL naturally by considering the whole procedure except the final training step in SCL as a feature construction process. In particular, the pivot features are chosen on both the source domain and target domain, and then the correlations between the pivot features and non-pivot features are learned. This correlation is then converted as a set of features that are then appended to the original feature space as is done in SCL. In the final training step, we apply SDAD to these newly represented examples. We name this method SCL-SDAD.

[1] SCL is a transfer learning method. Here, SCLT is implemented by using SCL to append the correlation features into the original feature space, and train a linear classifier based on the target domain examples by using SVM.

For both the proposed method and the baseline methods, their parameters are all set by five fold cross validations. For each experiment, we use all the examples from the source domain and a specified ratio of target domain examples as the training examples, while the rest of the target domain examples are used for testing. The averaged results of 10 independent runs are reported.

### 4.3 Results and analysis

The sentiment detection results on the product review datasets are reported in Figs.1 and 2. The results with product review datasets as the source domains, and the twitter datasets as the target domains are reported in Fig.3. The sentiment detection results with 90% target domain training examples and all of the source domain examples are further reported in Table 4.

As can be seen from these results, the proposed methods, i.e., SDAD and SCL-SDAD show the best performances in most cases. This is because through modeling examples on both the source and the target domains, examples on the source domain can be better incorporated to train a good classifier on the target domain. SVMST and SCLST can be considered as two special cases of SDAD and SCL-SDAD respectively, with $\beta_j$ being set to be 1. It is clear that by calculating appropriate $\beta_j$, the data distribution on the source domain can be tuned to fit that on the target domain.

From Fig. 3, it can be seen that the sentiment detection accuracies on the twitter dataset can be helped by the incorporation of some “complete” (on contrary to the short text on twitter dataset) examples from other domains. These results are also consistent with the results on Zhang et al. (2010, 2011), in which the authors improve the twitter classification accuracies by transferring the knowledge from some labeled webpages.

SVMT and SCLT are two methods that only consider training classifiers on the target domain. From the experimental results, it is clear that these two methods perform worse than SVMST and SCLST on the previous twelve tasks in most cases, while they are very competitive on the latter two tasks. This is because the similarities between different domains in the product review dataset are much higher than those between the product review dataset and the twitter dataset. Therefore, on the first twelve tasks, even if we don’t consider the distribution difference between different domains, the source domain examples can still be directly used to help to improve the performance on the target domain. But on the latter two tasks, since the domain difference is relatively high, incorporating source domain without considering these difference will sometimes degrade the classification performance. It is clear that on these two tasks, the proposed methods can still use the source domain examples, and show the best performances in most cases.

As for SCLT, SCLST and SCL-SDAD, we can conclude that SCL-SDAD performs better than SCLT and SCLST. This is because although SCLT and SCLST are transfer learning methods, they do not model the distribution difference directly. Even after the pivot features are picked and the correlations between pivot and non-pivot features are appended to the original feature space, this distribution difference still exists and deteriorates the the performances of the classifier. Different from these methods, after the pivot choosing and correlation learning steps, SCL-SDAD integrates the distribution difference into the objective function and reweighting the examples on the source domain in a reasonable way. Therefore, SCL-SDAD is superior to SCLT and SCLST.

There are in total three parameters in the proposed method, i.e., $\sigma$, $C_1$ and $C_2$. To study the robustness of the proposed method (SDAD), some parameter sensitivity experiments are also conducted by each time fixing two parameters and varying the other one.

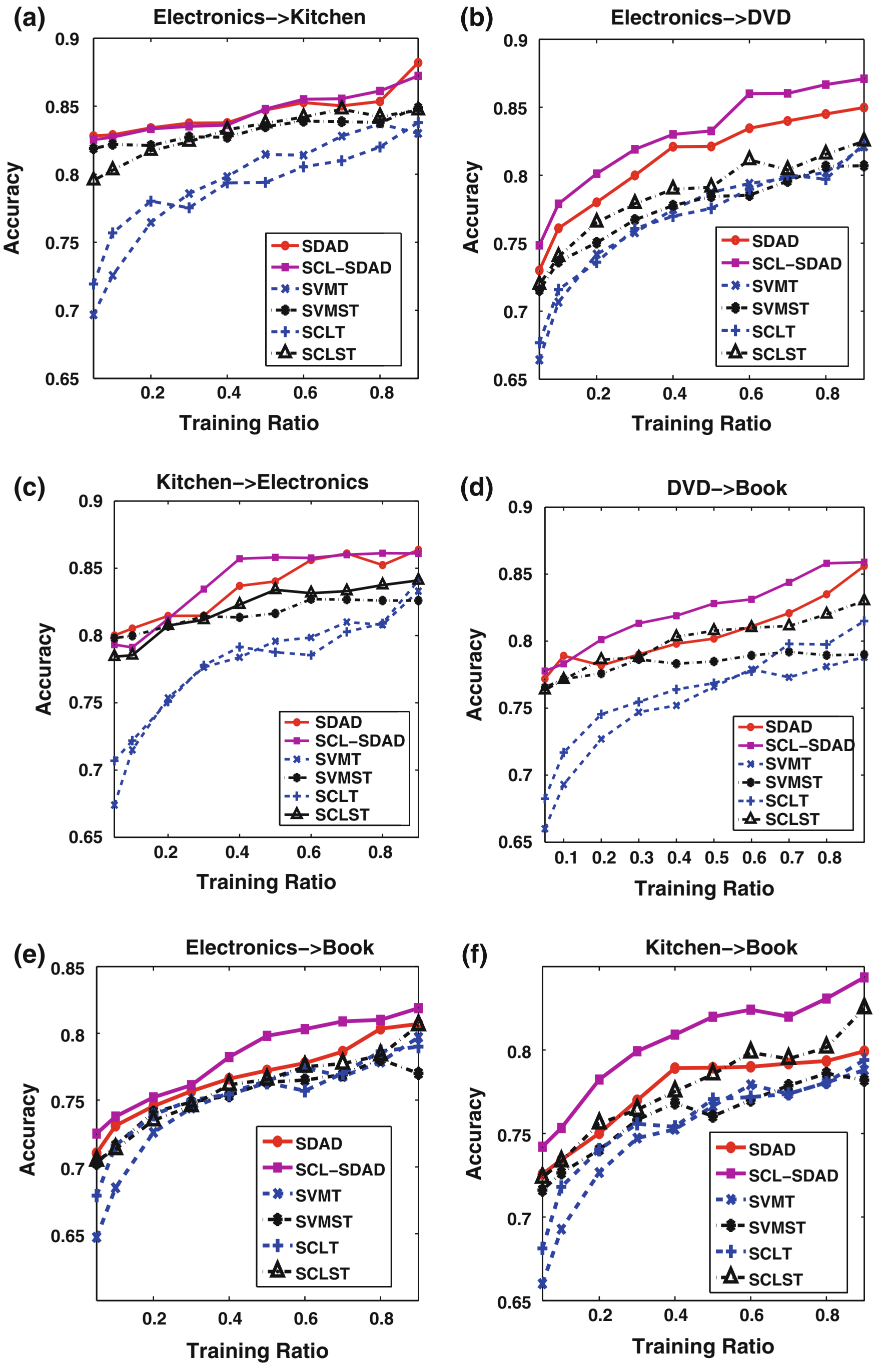

**Fig. 1** Sentiment detection accuracy, with different training ratios on the target domain and all of the examples on the source domain. The x-axis represents the different training ratios on the target domain, while the y-axis demonstrates the corresponding classification accuracy

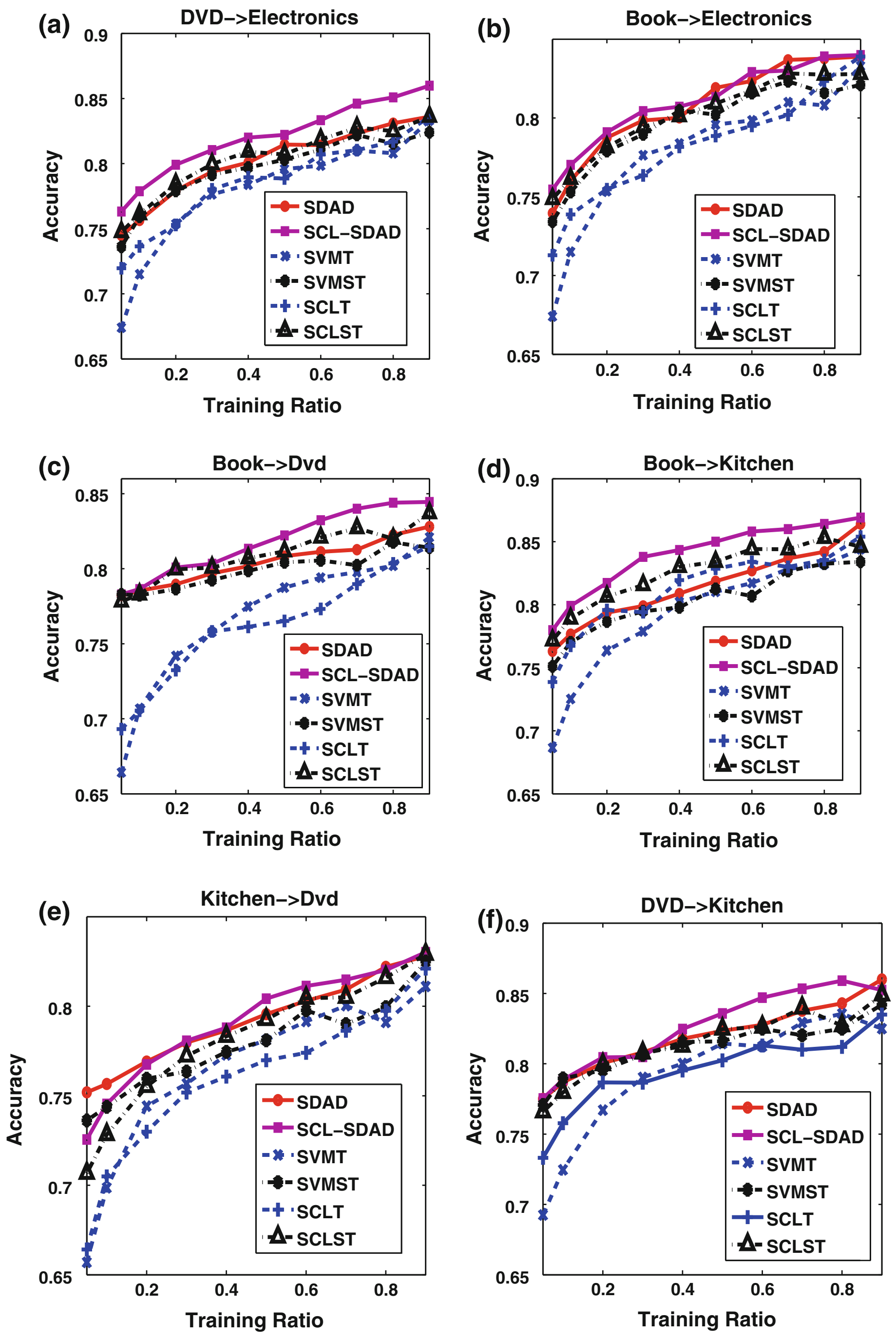

**Fig. 2** Sentiment detection accuracy, with different training ratios on the target domain and all of the examples on the source domain. The x-axis represents the different training ratios on the target domain, while the y-axis demonstrates the corresponding classification accuracy

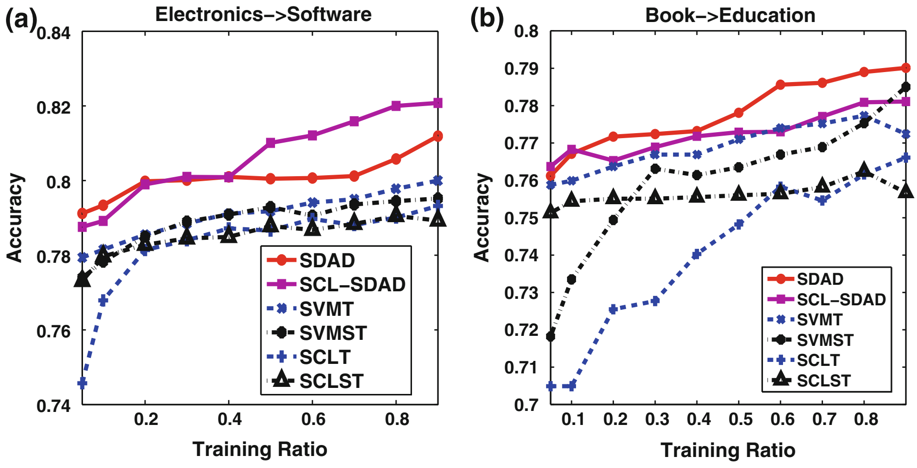

**Fig. 3** Sentiment detection accuracy, with different training ratios on the target domain and all of the examples on the source domain. The x-axis represents the different training ratios on the target domain, while the y-axis demonstrates the corresponding classification accuracy

**Table 4** Sentiment detection accuracies with 90% examples on the target domain, and all the labeled examples in the source domain as the training set, and the remaining examples in the target domain as the testing set

| Task # | SDAD | SVMT | SVMST | SCL-SDAD | SCLT | SCLST |
|---|---|---|---|---|---|---|
| 1 | **0.882** ± 0.003 | 0.830 ± 0.004 | 0.849 ± 0.003 | **0.872** ± 0.003 | 0.838 ± 0.002 | 0.847 ± 0.004 |
| 2 | **0.850** ± 0.002 | 0.821 ± 0.004 | 0.807 ± 0.005 | **0.871** ± 0.005 | 0.823 ± 0.003 | 0.825 ± 0.005 |
| 3 | **0.864** ± 0.003 | 0.833 ± 0.006 | 0.826 ± 0.005 | **0.861** ± 0.004 | 0.841 ± 0.005 | 0.841 ± 0.003 |
| 4 | **0.856** ± 0.011 | 0.788 ± 0.009 | 0.790 ± 0.007 | **0.859** ± 0.010 | 0.815 ± 0.006 | 0.830 ± 0.007 |
| 5 | **0.807** ± 0.002 | 0.797 ± 0.003 | 0.770 ± 0.004 | **0.819** ± 0.005 | 0.790 ± 0.003 | 0.806 ± 0.004 |
| 6 | **0.799** ± 0.003 | 0.788 ± 0.005 | 0.782 ± 0.003 | **0.844** ± 0.002 | 0.794 ± 0.001 | 0.825 ± 0.002 |
| 7 | **0.828** ± 0.012 | 0.811 ± 0.009 | 0.825 ± 0.008 | **0.830** ± 0.008 | 0.821 ± 0.005 | 0.829 ± 0.006 |
| 8 | **0.860** ± 0.004 | 0.825 ± 0.004 | 0.842 ± 0.003 | **0.852** ± 0.005 | 0.835 ± 0.004 | 0.849 ± 0.001 |
| 9 | **0.836** ± 0.003 | 0.833 ± 0.002 | 0.824 ± 0.003 | **0.860** ± 0.002 | 0.837 ± 0.004 | 0.836 ± 0.003 |
| 10 | **0.839** ± 0.013 | 0.833 ± 0.009 | 0.821 ± 0.010 | **0.840** ± 0.011 | 0.839 ± 0.007 | 0.828 ± 0.009 |
| 11 | **0.828** ± 0.007 | 0.821 ± 0.004 | 0.814 ± 0.005 | **0.845** ± 0.004 | 0.815 ± 0.006 | 0.837 ± 0.002 |
| 12 | **0.864** ± 0.009 | 0.844 ± 0.011 | 0.834 ± 0.009 | **0.869** ± 0.005 | 0.855 ± 0.007 | 0.846 ± 0.003 |
| 13 | **0.812** ± 0.010 | 0.800 ± 0.009 | 0.795 ± 0.010 | **0.821** ± 0.008 | 0.793 ± 0.006 | 0.789 ± 0.007 |
| 14 | **0.790** ± 0.015 | 0.772 ± 0.008 | 0.785 ± 0.010 | **0.781** ± 0.012 | 0.766 ± 0.010 | 0.757 ± 0.010 |

The best performances are marked in bold. It is clear that the proposed methods show the best performances on these tasks

The experimental results on Task 5 and Task 7 are reported in Fig. 4. It can be seen from these experiments that the proposed method is relatively robust with different parameter values.

## 5 Conclusions

Sentiment detection is an important technique for investigating what people think in opinion rich resources such as online review sites, microblogging sites and personal blogs.

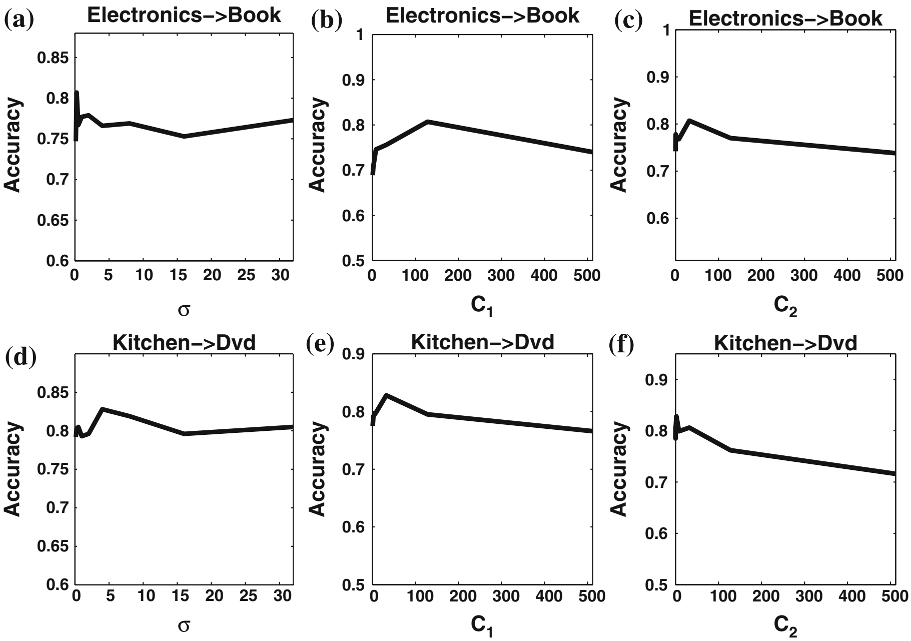

**Fig. 4** The parameter sensitivity of the proposed method. The experiments are conducted by using all of the source domain examples and 90% of the target domain examples as the training set, while fixing the following examples as the testing set. For each experiment, we fix two parameters and tune the other one. The average performances of 10 independent runs are reported

Transfer learning has been utilized in sentiment detection to transfer knowledge from one source domain with rich labeled information to another target domain. However, most existing transfer learning techniques for sentiment detection simply append some correlation features to the original feature space and the problem of distribution difference still exists. Moreover, the commonly used assumption on the conditional probability is too strong for practical applications. To address these problems, this paper presents a new method that directly models the joint distribution difference on different domains, and an efficient method is proposed to optimize the proposed formulations. The proposed method is guaranteed to converge within a finite number of steps. An extensive set of examples clearly demonstrate the advantages of the proposed method over the state-of-the-art methods. In the future, we plan to extend the proposed method to the the setting of multitask learning, as well as the mood classification, in which more than two classes exist.

**Acknowledgments** This work is partially supported by NSF research grants IIS-0746830, CNS-1012208 and IIS-1017837. This work also partially supported by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

## Appendix: Theoretical analysis

In this subsection, we deduct the convergence rate, and the time complexity of the proposed method.

Convergence rate

**Theorem 1** *For the algorithm developed in*Table 1, *suppose* $\beta_j \leq B$. *The proposed method converges to the precision* $\epsilon in O(1/\epsilon)$ *steps. In particular,*

- If $\epsilon > 8(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)$, the proposed method converges to precision $\epsilon$ after at most $log_2 \frac{C1+BC_2}{2(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2+B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)}$ steps.
- If $\epsilon \leq 8(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)$, the proposed algorithm converges to precision $\epsilon$ after at most

$$\begin{aligned} &log_2 \frac{C1+BC_2}{2(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)} \\ &+ \frac{16(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)}{\epsilon} - 1 \end{aligned} \tag{6}$$

steps.

*Proof* It is clear that:

$$\begin{aligned} \|\partial_{\mathbf{w}} R_{emp}(\mathbf{w})\|^2 &= \| \frac{C_1}{n}\sum_{i=1}^{n} I_i^S y_i \mathbf{x}_i + \frac{C_2}{m}\sum_{j=1}^{n} I_j^T \beta_j y_j^* \mathbf{z}_i \|^2 \\ &\leq 2(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2). \end{aligned} \tag{7}$$

and

$$J(\mathbf{0}) = C_1 + \frac{C_2}{m}\sum_{j=1}^{m} \beta_j \leq C1 + BC_2 \tag{8}$$

By integrating these inequations into Theorem 4 of Teo et al. (2010), we can get

$$\epsilon_t - \epsilon_{t+1} \geq \frac{\epsilon_t}{2}\min\{1, \epsilon_t/8(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)\}, \tag{9}$$

where $\epsilon_t = \min_{0\leq i\leq t} J(\mathbf{w}_i) - J_t(\mathbf{w}_t)$. The algorithm will terminate if $\epsilon_t \leq \epsilon$. So, if $\epsilon > 8(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2), \epsilon_t - \epsilon_{t+1} \geq \frac{\epsilon_t}{2}$, and the algorithm terminate after at most:

$$\begin{aligned} &log_2 \frac{J(\mathbf{0})}{2(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)} \\ &\leq log_2 \frac{C1+BC_2}{2(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)} \end{aligned} \tag{10}$$

steps.

If $\epsilon \leq 8(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)$, then, this algorithm needs the above steps to converge to $8(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)$, then, we should have $\epsilon_t - \epsilon_{t+1} \geq \frac{\epsilon_t^2}{16(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2+B^2C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)}$. It is clear that it needs another

$\frac{16(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2 C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)}{\epsilon} - 1$ steps to converge to the precision $\epsilon$. So, in total, the algorithm converges in

$$log_2 \frac{C1 + BC_2}{2(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2 C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)} + \frac{16(C_1^2 \max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2 + B^2 C_2^2 \max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2)}{\epsilon} - 1 \tag{11}$$

steps. □

In summary, the algorithm converges in $O(1/\epsilon)$ steps. It is clear that the convergence rate is highly influenced by $C_1$ and $B$ $C_2$, as well as $\max_{i\in\{1,\ldots,n\}} \mathbf{x}_i^2$ and $\max_{j\in\{1,\ldots,m\}} \mathbf{z}_j^2$. So, given a dataset, smaller $C_1$ and $BC_2$ normally lead to faster convergence rates.

Time complexity

**Theorem 2** *For each iteration of the proposed method, it takes time* $O(s(m + n))$, *where* $s$ *is the average feature sparsity on both the source domain and the target domain.*

*Proof* The gradient computation in step 5 takes time $O((m + n)s)$. Instead of solving the primal quadratic programming problem in step 7, one can instread solve it in the dual form. Setting up the dual requires computing $O(t^2)$ elements of the Hessian, which can be done in $O(t^2\ s)$ steps. Since $t^2$ is normally much less than $(m + n)$, the overall time complexity is dominated by $O(s(m + n))$ per iteration. □

This result is actually similar to that in Joachims (2006). However, the total number of iterations in Joachims (2006) can be as worse as $O(1/\epsilon^2)$, as given by the Lemma 2 of Joachims (2006). The proposed method is guaranteed to converge within $O(1/\epsilon)$ steps. So, solving the proposed formulation by bundle method is much faster than using the Cutting Plane method(Kelley 1960).

## Reference

Ando R. K., Zhang, T., & Bartlett, P. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR, 6*, 1817–1853.

Argamon, S., Koppel, M., & Avneri, G. (1998). Routing documents according to style. First International workshop on innovative information systems.

Argyriou, A., Evgeniou, T., Pontil, M. (2007). Multi-task feature learning. In *Advances in neural information processing systems: Proceedings of the 2006 conference*. Cambridge: MIT Press.

Bishop, C.M. (2007). *Pattern recognition and machine learning (information science and statistics)* (1st ed.). Berlin: Springer.

Blitzer, J., McDonald, R., & Pereira, F.(2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 120–128). Association for Computational Linguistics.

Blitzer, J., Dredze, M., & Pereira, F. (2012). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Association for computational linguistics*. Prague, Czech Republic.

Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th international conference on machine learning* (pp. 19–26). San Francisco, CA.

Bonilla, E., Chai, K., & Williams, C. (2008). Multi-task Gaussian process prediction. *Advances in Neural Information Processing Systems, 20*, 153–160.

Cardie, C., Wiebe, J., Wilson, T., & Litman, D. J. (2003). Combining low-level and summary representations of opinions for multi-perspective question answering. In *New directions in question answering* (pp. 20–27).

Dai, W., Yang, Q., Xue, G., & Yu, Y. (2007). Boosting for transfer learning. In *Proceedings of the 24th international conference on machine learning*. New York: ACM.

Dai, W., Yang, Q., Xue, G., & Yu, Y. (2008). Self-taught clustering. In *Proceedings of the 25th international conference on machine learning* (pp. 200–207). New York: ACM.

Das, S., & Chen, M. (2007) Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science*, *53*(9), 1375–1388.

Das, S., & Chen, M. (2001). Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference (APFA)*.

Davis, J., & Domingos, P. (2009). Deep transfer via second-order Markov logic. In *Proceedings of the 26th annual international conference on machine learning* (pp. 217–224). New York: ACM.

Dimitrova, M., Finn, A., Kushmerick, N., & Smyth, B. (2002). Web genre visualisation. In *Conference on human factors in computing systems*.

Duan, L., Tsang, I., Xu, D., & Maybank, S. (2009). Domain transfer svm for video concept detection. *Computer vision and pattern recognition, IEEE computer society conference* (pp. 1375–1381).

Duda, R., Hart, P., & Stork, D. (2001). Pattern classification.

Hillard, D., Ostendorf, M., & Shriberg, E. (2003). Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *HLT-NAACL*.

Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., & Schölkopf, B. (2006). Correcting sample selection bias by unlabeled data. In *NIPS* (pp. 601–608).

Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 217–226). New York, NY, USA: ACM.

Kelley, J. (1960). The cutting plane method for solving convex programs. *Journal of the SIAM, 8*(4), 703–712.

Kessler, B., Nunberg, G., & Schütze H. (1997). Automatic detection of text genre. In *ACL* (pp. 32–38).

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML* (pp. 282–289).

Lawrence, N., & Platt, J. (2004). Learning to learn with the informative vector machine. In *Proceedings of the twenty-first international conference on machine learning*. New York: ACM.

Liu, X., Gong, Y., Xu, W., & Zhu, S. (2002). Document clustering with cluster refinement and model selection capabilities. In *SIGIR* (pp. 191–198).

Manning, C. D., Raghavan, P., & Schtze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.

Mihalkova, L., Huynh, T., & Mooney, R. (2007). Mapping and revising Markov logic networks for transfer learning. In *Proceedings of the national conference on artificial intelligence*.

Mihalkova, L., & Mooney, R. (2008). Transfer learning by mapping with minimal target data. In *Proceedings of the AAAI-08 workshop on transfer learning for complex tasks*.

Morinaga, S., Yamanishi, K., Tateishi, K., & Fukushima T. (2002). Mining product reputations on the web. In *KDD* (pp. 341–349).

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using em. *Machine Learning, 39*(2/3), 103–134.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10): 1345–1359.

Pan, S., Kwok, J., & Yang, Q. (2012) Transfer learning via dimensionality reduction. In *Proceedings of AAAI* (pp. 677–682).

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval, 2*(1-2), 1–135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. CoRR, cs.CL/0205070.

Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL* (pp. 271–278).

Raina, R., Battle, A., Lee, H., Packer, B., & Ng, A. (2007). Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*. New York: ACM.

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., Chen, Z. (2010). Cross-domain sentiment classification via spectral feature alignment. In *WWW* (pp. 751–760).

Scholkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge, Mass: MIT press.

Sheather, S., & Jones, M. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological),53*(3), 683–690.

Smola, A., Vishwanathan, S., & Le, Q. (2008). Bundle methods for machine learning. NIPS.

Spertus, E. (1997). Smokey: Automatic recognition of hostile messages. In *AAAI/IAAI* (pp. 1058–1065).

Taboada, M., Gillies, M. A., McFetridge, P. (2006). Sentiment classification techniques for tracking literary reputation. In *LREC workshop: towards computational models of literary analysis* (pp. 36–43).

Teo, C., Vishwanthan, S., Smola, A., & Le, Q. (2010). Bundle methods for regularized risk minimization. *The Journal of Machine Learning Research, 11*, 311–365.

Thomas, M., Pang, B., & Lee, L. (2006). Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 327–335). Association for Computational Linguistics.

Whitelaw, C., Garg, N., & Argamon, S. (2005). Using appraisal groups for sentiment analysis. In *CIKM* (pp. 625–631).

Wiebe, J., & Riloff, E. (2005). Creating subjective and objective sentence classifiers from unannotated texts. In *CICLing* (pp. 486–497).

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP*.

Wu, P., & Dietterich, T. G. (2004). Improving svm accuracy by training on auxiliary data sources. In *ICML*.

Zhang, D., & Si, L. (2009). Multiple instance transfer learning. In *ICDM Workshops* (pp. 406–411).

Zhang, D., Liu, Y., Lawrence, R. D., & Chenthamarakshan, V. (2010). Alpos: A machine learning approach for analyzing microblogging data. In *ICDM workshops* (pp. 1265–1272).

Zhang, D., Liu, Y., Lawrence, R. D., & Chenthamarakshan, V. (2011). Transfer latent semantic learning: Microblog mining with less supervision. AAAI.

Zhu, X. (2006). *Semi-supervised learning literature survey*. Computer Science, University of Wisconsin-Madison.