Journal of Computers Vol. 30 No. 6, 2019, pp. 276-285
doi:10.3966/199115992019123006023

Cross-domain Text Sentiment Classification Based on Wasserstein Distance

Guoyong Cai[1*], Qiang Lin[1], Nannan Chen[1]

[1] Guangxi Key Lab of Trusted Software, Guilin University of Electronic Technology, 541004 Guilin, China
ccgycai@gmail.com, firejohnny@outlook.com, 1728792152@qq.com

Received 10 May 2019; Revised 10 June 2019; Accepted 2 July 2019

Abstract. Text sentiment analysis is mainly to detect the sentiment polarity implicit in text data. Most existing supervised learning algorithms are difficult to solve the domain adaptation problem in text sentiment analysis. The key of cross-domain text sentiment analysis is how to extract the domain shared features of different domains in the deep feature space. The proposed method uses denosing autoencoder to extract the deeper shared features with better robustness. In addition, Wasserstein distance-based domain adversarial and orthogonal constraints are combined for better extracting the deep shared features of the different domain. Finally, the deep shared features are used for cross domain sentiment classification. The experimental results on the real data sets show that the proposed method can better adapt to domain differences and achieve higher accuracy.

Keywords: cross-domain, domain adversarial, text sentiment analysis, wasserstein distance

1 Introduction

Traditional text sentiment analysis researches usually assume that the training data and the target data are from the same space and with the same distribution. With such data, supervised learning algorithm can train an appropriate sentiment polarity classifier and get qualified classification results. However, it requires a large amount of high-quality manual annotation data to train the classifier. In practical applications, manual annotation data requires experts to understand the data so that it is expensive and labor-intensive to obtain a large amount of annotation data. Text sentiment analysis is a task which is sensitive to domains cause that the sentiment expression features of different domains are different. The classification model trained in one domain usually cannot be directly applied to other domains [1]. In order to reasonably and effectively apply the sentiment classification model trained in one domain to other domains, namely maximizing the domain adaptation ability of the sentiment classification model and reducing the annotation cost of the new domain, cross-domain sentiment analysis [2] has emerged and received increasing attention from the academic community. At present, there are two main methods in cross-domain text sentiment analysis: instance-based cross-domain sentiment analysis and feature-based cross-domain sentiment analysis [3].

The instance-based cross-domain sentiment analysis realizes the knowledge transfer from the source domain to the target domain by finding the connection between the source domain and the target domain, and then applies the transfer knowledge to the target domain for classification task. Pan et al. [2] proposed a spectral feature alignment algorithm to construct the relationship between different domains. Through the spectral clustering algorithm, the domain independent words with high correlation are clustered under one cluster, and the domain co-occurrence words are also clustered in one for the purpose of the transfer learning of domain knowledge. However, the defect of instance-based method is that it has neither extracted the text feature nor learned the language features of text (such as the context information in the text). When the spatial distributions of the source and target domains are different, the model is less robust.

[*] Corresponding Author

Feature-based cross-domain sentiment analysis constructs a unified feature representation of cross-domain data by searching the correlation or the co-occurrence features between the source and the target domains. Glorot et al. [4] applied Stacked Denoising Autoencoder (SDA) to cross-domain sentiment analysis. They first extracted words or phrases with higher frequency as their respective feature set from the source domain and target domain, and then reconstruct features of the source domain and the target domain through SDA, so that the features of the source domain and the target domain can show the same distribution in the potential space. Finally, they trained the classifier with the source domain feature and applied it to the target domain to realize the cross-domain text sentiment analysis task. Inspired by Goodfellow et al.'s generative adversarial nets [5], Ajakan et al. [6] integrated the source domain data with the target domain data, and then labeled the source domain data and the target domain data with domain labels. Through maximizing the cross entropy loss, the weights of network were updated and achieved the purpose of domain adaptation. Next, training the feature extractor with the domain adversarial of the source domain and the target domain, and training sentiment classifier with the deep features of the source domain and the target domain adversarial simultaneously. Finally, with the deep features extracted from target domain by adversarial, they can predict the sentiment polarity of the text of the target domain and thereby realizing cross-domain text sentiment analysis. However, Arjovsky and Bottou [7] mathematically proved that the method based on adversarial which proposed by Goodfellow et al. proposed would cause instability in the generative adversarial nets training, and the generated samples are also in the lack of lacked diversity. In order to alleviate these problems, Arjovsky et al. [8] proposed another function Wasserstein distance instead of the original cross entropy function. Linear regression combined with weight cropping is used to alleviate the problem that the generative and adversarial process is not easy to converge and the samples are diversity lacked. In the cross-domain classification task of images, Konstantinos et al. [9] proposed Domain Separation Networks (DSN) to separate the domain's own private features and the shared features between domains, and thus achieved domain feature separation, and then, they optimized the similarity loss function to train the feature extraction network, finally trained the classifier through the labeled image data, and achieved good results. This paper improves DSN and applies it to cross-domain text sentiment analysis.

The above research can achieve shared feature extraction to a certain extent by means of the shared network structure. But Salzmann et al. [10] found that the features extracted by the shared feature extraction network contain a large number of private domain features since the sentiment classifier is trained by the shared features of the source domain and the corresponding sentiment labels. In order to obtain better domain shared features and improve the accuracy of cross-domain text sentiment classification, by means of DSN network, this paper proposes a domain separation model based on Wasserstein distance (W-DSN) which combining denosing autoencoder and domain adversarial method based on Wasserstein distance together.

2 Wasserstein Distance Based Domain Adversarial Model

2.1 Task Description

The task of cross-domain text sentiment analysis in this paper refers to training the sentiment classifier with the sentiment -labeled data in the source domain, and achieving sentiment classification in the target domain data. Suppose that there are a source domain data set D_S and a target domain data set D_T, in which x^S is the text data with label and x^t is the text data without label. N_S and N_i denote the number of x^S and x^t respectively. That is $x^S = \{(x_i^S)_{i=1}^{N_S}(x^S \sim D_S)\}$ and $x^t = \{(x_i^t)_{i=1}^{N_1}(x^t \sim D_T)\}$. Obviously, it is easy to use the deep features of the source domain to train the source domain sentiment classifier via the sentiment labels in the source domain. But for the target domain, it is difficult to obtain a classifier since there is no label in the target domain. In order to apply the source domain private deep features and labels to the target domain for the text sentiment classification task, it is necessary to firstly assume that the sentiment classifier is shared by the source domain and the target domain, that is, the distributions over the source domain private features and the target domain private features are same in the feature space. The work of this paper is to train a sentiment classifier with the texts and labels in the source domain data set and then combined with the target domain data set to do the adversarial training. Finally, the sentiment classifier can effectively classify the text data in the target domain.

2.2 W-DSN Description

This section will introduce the cross-domain text sentiment classification network (W-DSN) based on Wasserstein distance. It consists of six parts and the overall structure is show in Fig. 1. Specially, it includes: (1) $E_p^s(x^S)$, the private encoder of the source domain as Fig. 1(a) shows. (2) $E_p^t(x^t)$, the private encoder of the target domain as Fig. 1(c) shows. (3) shared encoder $E(x)$ as Fig. 1(b) shows. (4) sentiment classifier $G(h_c^s)$ as Fig. 1(d) shows. (5) domain adversarial net $D_a(h_c^{s+t})$ as Fig. 1(e) shows. (6) shared decoder $D_c(h_p + h_c)$ as Fig. 1(f) shows. (7) $L_{senti}, L_{domain}, L_{rcom}$ denote the loss functions.

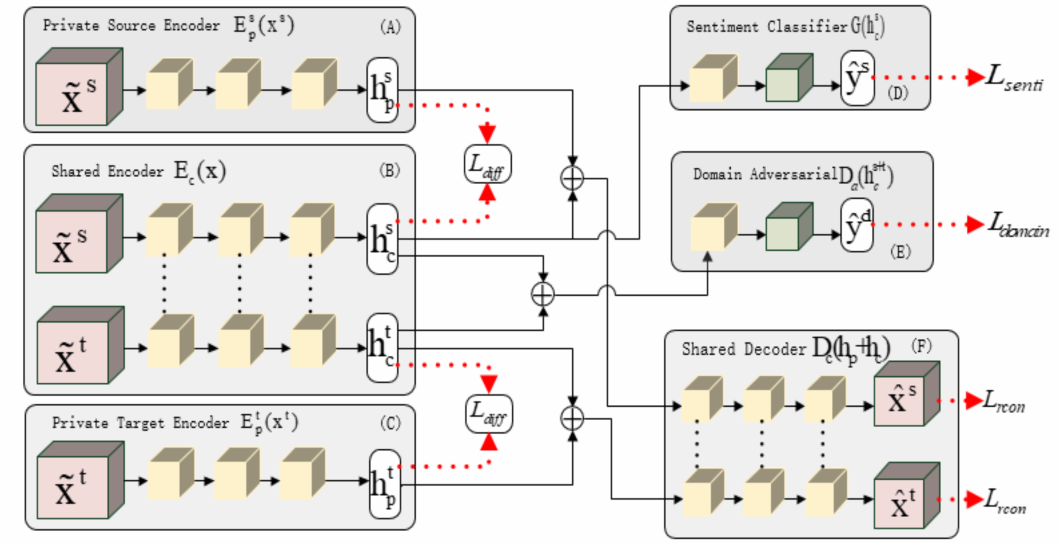


Fig. 1. Cross-domain Text Sentiment Classification method based on domain separation

Firstly, the deep feature representation of text data in different domains is extracted by DAE, and then the domain adaptation is realized by adversarial and orthogonal constraint. Finally, the source domain features are used to train the sentiment classifier and the text sentiment of the target domain can be predicted by the classifier trained in the source domain. The details of the proposed model will be described in sections 2.3-2.7.

2.3 Deep Features Extraction with Denosing Autoencoder

Denosing autoencoder (DAE) firstly adds noise to the raw input and then extracts the main feature by encoder and decode the main feature by the decoder. Then, it can reconstruct the raw input. Because of the noise added to the raw input, DAE will own better generation ability than the standard autoencoder. W-DSN utilizes the DAE to extract the deep feature of the source domain text data and the target domain text data so that they can be projected into the same feature space.

Specifically, W-DSN uses a three-layer fully connected network to construct a private encoder of the source domain (as shown in Fig. 1(a)), a private encoder of the target domain(as shown in Fig. 1(c)) and a shared encoder (as shown in Fig. 1(b)) to extract shared features from source domain and target domain text data. Firstly, source domain text data x^S and target domain data x^t are artificially noised as $\tilde{x}^S$ and $\tilde{x}^t$. And then, take $\tilde{x}^S$ into private encoder $E_p^s(x^S)$ of the source domain and then the deep feature representation $h_p^s(\tilde{x}^S)$ of the source domain is obtained. The deep feature representation $h_c^t = E_p^t(\tilde{x}^t)$ of the target domain is obtained in the same way. In addition, W-DSN takes $\tilde{x}^S$ and $\tilde{x}^t$ into the shared encoder to get the shared features $h_c^s = E_c(\tilde{x}^S)$ and $h_c^s = E_c(\tilde{x}^t)$.

After that, a shared decoder (Fig. 1(f)) which is also consisted of a three-layer fully connected network is used to reconstruct the source domain deep feature representation, the target domain deep features representation and the domain shared feature representation. By minimizing the reconstruction loss function L_{rcom}, deep features of different domain texts could be captured. W-DSN takes mean square error as the loss function which is formulated as follows.

$$L_{rcon} = \sum_{\mathrm{i}=1}^{N_s+N_t} \left\| \mathrm{x}_i - \hat{\mathrm{x}}_i \right\|^2 . \quad \textbf{(1)}$$

Where x_i denote the input data and $\hat{x}_i$ denote the reconstructed data.

2.4 The Domain Adversarial Based on Wasserstein Distance

In order to make feature distribution of the source domain and the target domain more consistent in the deep feature space, W-DSN draws on the idea of the domain confrontation [5, 8] process and uses domain adversarial net (Fig. 1(e)) to make the deep features of the source domain and the target domain with the same distribution in the feature space. In order to better learn the sentiment-related features, W-DSN takes the Wasserstein distance as the adversarial-loss instead of cross entropy. The Wasserstein distance can be used to evaluate the distance between two fields. Even if two domains are absolutely different, Wasserstein distance can offer gradient for extracting features. But for cross entropy, the gradient equals to zero under such a condition. The Wasserstein distance between the two domains needs to be calculated as:

$$W_p(P,Q) = \inf_{\gamma \sim \Pi(P,Q)} \mathbb{E}_{(x,y)\sim\gamma} (\|x-y\|) . \quad \textbf{(2)}$$

Where inf denote the low bound of the function, $\Pi(P,Q)$ denote all the possible joint distribution of P and Q, thus the marginal distributions of each one in $\Pi(P,Q)$ are P and Q. For each possible joint distribution γ, the distance of samples is calculated as $\|x-y\|$. Since Wasserstein can not be solved directly, Arjovsky and Bottou [7] convert formula (2) to formula (3) according to the duality of Kantorovich-Rubinstein.

$$W_p(P,Q) = \sup_{\|f\|_L \le K} \mathbb{E}_{x\sim P}\left[f(x)\right] - \mathbb{E}_{y\sim Q}\left[f(y)\right] . \quad \textbf{(3)}$$

Where $\|f\|$ is the half-norm of Lipschitz, *sup* means the upper bound, and $\|f\|$ satisfies with the Lipschitz continuity [5]. Under such a condition, W-DSN takes formula (3) as the domain adversarial loss function, in other words, it takes Wasserstein distance as the domain adversarial loss function. So in Fig. 1, equation (4) is obtained.

$$L_{w-d} = \max\left(\frac{1}{N_s}\sum_{i=1}^{N_s} f(\mathrm{h}_i^s) - \frac{1}{N_t}\sum_{i=1}^{N_t} f(\mathrm{h}_i^t)\right) \quad \textbf{(4)}$$

Where $f(\cdot)$ denote linear function. In order to satisfy with the Lipschitz continuity [11] condition, we cut the gradient to the scale of [c, -c] as Arjovsky et al. [8] does. But Gulrajani [12] point out that such cut will result in Gradient disappearance or gradient explosion. So they use gradient regularization for the condition of Lipschitz continuity. Gradient regularization is to regularize the gradient value of the network before updating the network weight, that is, to regularize all the gradient values and then update the W-DSN model, and this process is called GP-DSN. The regularization of W-SDN is formulated as equation (5):

$$L_{gp} = \left(\left\|\nabla_{\mathrm{x}} f(\mathrm{x})\right\|_p - c\right)^2 . \quad \textbf{(5)}$$

Where c is a hyper parameter. So the loss function L_{domain} of GP-DSN equals to L_{gp-d}.

$$L_{gp\text{-}d} = \frac{1}{N_s}\sum_{i=1}^{N_s} f(\mathrm{x}_i^s) - \frac{1}{N_t}\sum_{i=1}^{N_t} f(\mathrm{x}_i^t) + \alpha\left(\left\|\nabla_{\mathrm{x}} f(\mathrm{x})\right\|_p - c\right)^2 . \quad \textbf{(6)}$$

Where α is the balance coefficient.

2.5 Domain Separation Based on Orthogonal Constraint

Although the shared deep features are extracted by the encoder of the domain adversarial training are already similar, the deep features still contain a large number of private domain features. In order to eliminate the private domain features as much as possible, we take advantage of the orthogonal constraint [10] to separate the private features and the shared feature as much as possible for the purpose of domain separation, and combine with the domain adversarial process to make the shared features' distribution more similar. Orthogonal constraint push the product of two eigenvectors as close as possible to zero in order to make as much as possible differences between the private and shared features. The Orthogonal loss of W-DSN is formulated as follows.

$$L_{diff} = \left\| h_c^{sT} h_P^s \right\|_F^2 + \left\| h_c^{tT} h_P^t \right\|_F^2 . \tag{7}$$

Where h_c^s and h_c^t denote the domain shared deep feature representation, h_p^s denote the private deep feature representation of the source domain, h_p^t denote the private deep feature representation of the target domain and $\| \cdot \|_F^2$ is the square of Frobenius norm. Minimizing the loss could make the domain shared deep feature representation h_c^s and h_c^t more similar by optimizing the adversarial loss and the orthogonal constraint loss.

2.6 Sentiment Classifier

After the process of domain adversarial net and orthogonal constraint, the more similar deep feature representation h_c^s and h_c^t are obtained in the feature space. W-DSN training the sentiment classifier with the source domain shared deep features as shown in Fig. 1(d) and applying the trained sentiment classifier to the target domain for text sentiment classification task. After the domain confrontation network and orthogonal constraints are processed, the deep shared feature representation the more consistent distribution in the feature space are obtained. W-DSN performs sentiment classification by logistic regression. The loss function of the classifier is show in formula (6).

$$L_{senti} = -\sum_{i=0}^{N_s} y^s \log(\hat{y}^s) . \tag{8}$$

Where y^s denote the real sentiment label and $\hat{y}^s$ denote the predicted sentiment label. The weights are updated by minimizing the sentiment classification loss L_{senti}.

2.7 Object Function

All loss functions in the domain separation model are introduced in sections 2.3-2.7, where the loss functions L_{senti}, L_{domain} and L_{rcon} need to be minimized, and the loss function L_{domain} needs to be maximized. Suppose that the weights of the network need to be updated as $\{\theta\} = \{\theta_{p-enc}, \theta_{s-enc}, \theta_{dec}, \theta_s, \theta_d\}$, of which the element respectively denote the private feature encoder, domain shared feature encoder, decoder, sentiment classifier and the weight of adversarial net. In order to unify the training process, W-DSN applies the gradient reversal layer(GRL) which is proposed by Ganin and Lempitsky [13] the W-DSN, no operation is performed by GRL. But in the back propagation, L_{domain} is multiplied by a hyperparameter $-\lambda$ to minimize the overall loss function. Therefore, the final objective function of W-DSN is shown in equation (9) as follows.

$$\min L(\theta) = \frac{1}{N^s + N^t}(L_{senti} + \beta L_{diff} + \gamma L_{recon} - \eta L_{domain}) . \tag{9}$$

Where β, γ and η are hyperparameters. The update of $L(\theta)$ is formulated (10) as follows where μ is learning rate.

$$
\begin{aligned}
&\theta_{s-enc} \leftarrow \theta_{s-enc} - \mu(\frac{\partial L_{senti}}{\partial \theta_{s-enc}} + \beta \frac{\partial L_{diff}}{\partial \theta_{s-enc}} - \eta \frac{\partial L_{domain}}{\partial \theta_{s-enc}}) \\
&\theta_{p-enc} \leftarrow \theta_{p-enc} - \mu(\frac{\partial L_{diff}}{\partial \theta_{p-enc}}) \\
&\theta_{dec} \leftarrow \theta_{dec} - \mu(\frac{\partial L_{rcon}}{\partial \theta_{dec}}) \\
&\theta_{s} \leftarrow \theta_{s} - \mu(\frac{\partial L_{sent}}{\partial \theta_{s}}) \\
&\theta_{d} \leftarrow \theta_{d} - \mu(\frac{\partial L_{domain}}{\partial \theta_{d}})
\end{aligned}
\qquad \textbf{(10)}
$$

3 Experimental Result

3.1 Datasets

This paper uses the Amazon product review dataset provided by Glorot et al. [4] to evaluate the proposed model. This data set contains comments on specific products in four different areas, including books, DVD disk, electronics, and kitchen appliances. Each of these areas contains 2,000 labeled reviews (1000 positive reviews and 1000 negative reviews) and a number of unlabeled reviews. In this paper, each dataset of four different domain is used as the source domain dataset for one time, and the other three datasets are used as the target domain dataset at this time. The target domain data is divided into training set and test set (1600 is used for training, 400 used for testing). The statistics of the four data sets are show in Table 1.

Table 1. Dataset statistics

	Book	DVD	electronics	Kitchen
Pos	1000	1000	1000	1000
Neg	1000	1000	1000	1000
Unlabel	6000	34741	13153	16785
Vocabulary	171760	739346	237346	278187

In the text representation, all the data of the source domain and the target domain are processed with uni-grams and bi-grams statistics, and we removed the stop words with the english sentiment analysis stop words table provided by Baidu, and then the tf-idf of the first 5,000 words is used as the representation of the text data. The length of the dictionary for the final statistics is shown in vocabulary in Table 1.

3.2 Parameter Settings

W-DSN uses a three-layer fully-connected network to construct an encoder and a decoder. The number of neurons in the three-layer fully-connected network in the encoder is set to 1000, 500, and 200, respectively, and that in decoder is set to 200, 500, and 1000, respectively. The fully connected network in both the decoder and the encoder use relu as the activation function. This paper uses the RMSprop [14] to optimize network weights, and its learning rate is set to 0.001. The three hyper parameters β , γ and λ involved in the objective function are set to 1. All the experiments are finished on Tesla P100-PCIE GPU woks station. The operate system is Linux, the development environment are Python 2.7, Tensorflow 1.3.0, and the development tool is PyCharm.

3.3 Compared Methods

In order to prove the effectiveness of the proposed method for the prediction of sentiment polarity in the target domain, the method proposed is compared with the best method proposed in the existing research. At the same time, in order to understand the characteristics of the proposed model more intuitively, the paper reduce features into two dimensions after the data is extracted by different algorithms, and then visualizes the data.

We compare the method proposed in this paper with the following methods:

(1) LR: A logistic regression classifier is constructed using a three-layer fully connected neural network. And then trained in the source domain and tested on the target domain.

(2) SDA: Glorot et al. proposed Stack Denosing Autoencoder (SDAE) in [4]. The method uses a multi-layer MLP to construct a SDAE for main feature extraction, and then trains the classifier.

(3) mSDA: Chen et al. [15] proposed an improved model based on SDA. This method uses SDAE for main feature extraction and was time saving.

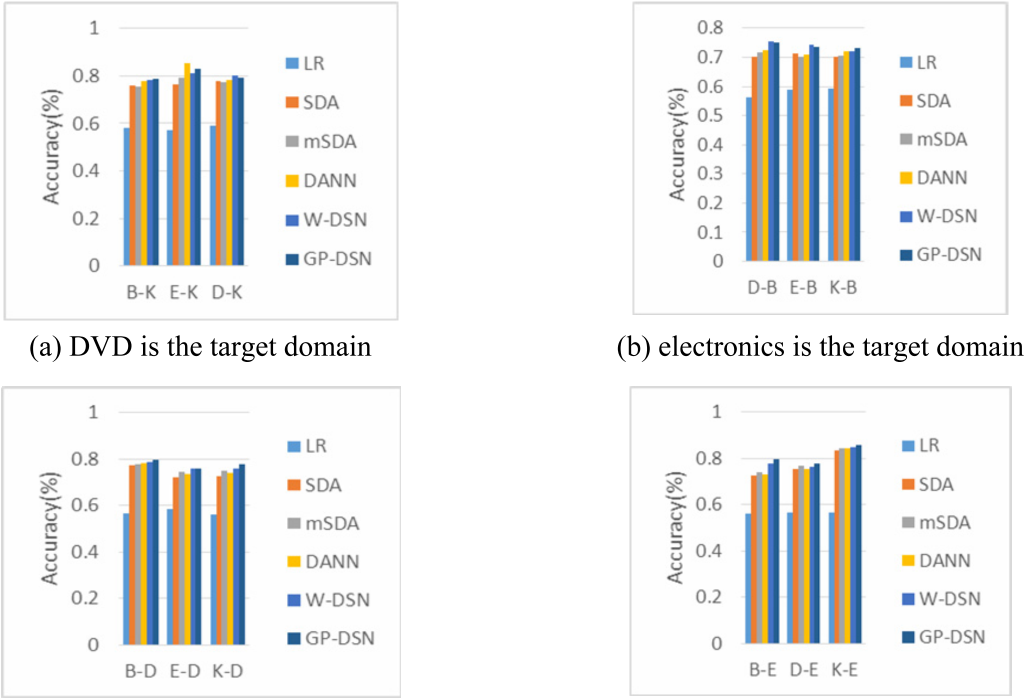
(4) DANN: Ajakan et al. [6] proposed domain adversarial net which combines mSDA to extract features firstly and then making the domain deep features over the same distribution via domain adversarial. Training the sentiment classifier with the deep features of the source domain and do sentiment classification task on the target domain.

(5) **W-DSN**: W-DSN is our proposed model. It first extracts the deep features of different domains with DAE, and then achieves domain adaptation with adversarial and orthogonal in the adversarial net. Finally training the sentiment classifier with the deep features of the source domain and do sentiment classification task on the target domain as DANN did.

(6) GP-DSN: Gradient Regulation Domain Separation (GP-DSN) is an optimized version of W-DSN. It regularized the gradient before the network weights are updated, and thus accelerate the convergence of the loss function.

3.4 Experimental Results and Analysis

Fig. 2 shows the performance of W-DSN, GP-DSN and the compared methods in different domains. As shown in Fig. 2, our proposed method is superior to existing methods on accuracy of sentiment classification.



(a) DVD is the target domain

(b) electronics is the target domain

(c) kitchen is the target domain

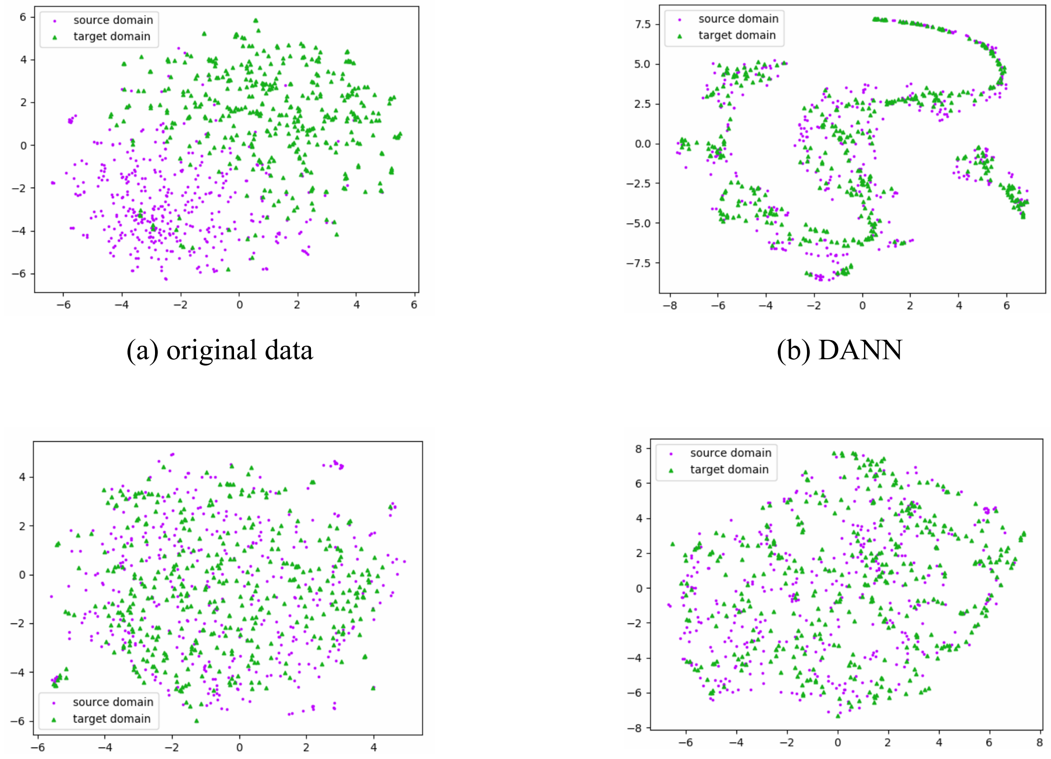
(d) book is the target domain

Fig. 2. The accuracy of cross-domain sentiment classification in four different domains

As shown in Fig. 2(a), LR only uses the source domain data to do cross-domain text sentiment analysis, but without using any cross domain related methods, so it doesn't achieve higher accuracy. This proves that training sentiment classifier on the source domain cannot adopt the features of the target domain. SDA and mSDA have better performance than LR, which proves that DAE can extract the shared feature between the source domain and the target domain and so that the sentiment classifier trained by these features achieves better performance. Further more, the performance of DANN is better than SDA and mSDA. It indicates that the domain adversarial has better generalization ability than SDA and mSDA. And domain adversarial has a positive role in cross-domain analysis. W-DSN not only use the structure of DAE, but also use processing on domain adversarial. The accuracy of W-DSN is 4.0% higher than DANN. It indicates that combining Wasserstein distance based domain adversarial with orthogonal constraint is helpful to cross-domain text sentiment analysis. Fig. 2 shows that GP-DSN has the best performance, which indicates the positive effect of gradient regularization in cross-domain modelling.

3.5 Data Distribution

In this section, the effectiveness of models will be illustrated by visualizing the distribution of data features. The adaptation of data will also be discussed. Taking book as the source domain data, dvd as the target domain data, and then we use T-SNE [16] to reduce the dimension of data to 2D and visualize the distribution of data features as shown in Fig.3.



(a) original data

(b) DANN

(c) W-DSN

(d) GP-DSN

Fig. 3. Feature distribution

Fig. 3(a) shows the feature distribution of the original data, and it can be seen that the feature distribution of the source domain data and the target domain data is inconsistent. Therefore, it is intractable to use the classifier trained to the source domain on the classification task on the target. Fig. 3(b) is the data feature distribution obtained from DANN. Compared Fig. 3(a) with Fig. 3(b), it can be found that the feature distribution of the source domain data and the target domain data is confused after domain adversarial training. However, the data feature distribution obtained from DANN has a narrow coverage. Fig. 3(c) and Fig. 3(d) respectively show the feature distribution of W-DSN and GP-DSN. It can be found that W-DSN and GP-DSN perform better in the confused of feature distribution, and the coverage is more complete, which leads to more for features are reserved for sentiment classification, thereby, W-DSN and GP-DSN gets more robust performance.

4 Conclusion

In recent years, cross-domain sentiment analysis has become an increasingly important research hotspot. This paper proposes two cross-domain sentiment analysis models based on Wasserstein distance with domain adversarial model, which better solves the problem of domain adaptation and improves the accuracy of sentiment classification. That is, the proposed method uses DAE to extract the private features and domain shared features, and uses adversarial and orthogonal constraint to achieve domain adaptation, thus that the features distribution of the source domain and the target domain becomes more similar in the feature space, and more sentiment features are reserved. The sentiment classifier is trained with the features of the source domain and applied to sentiment classification on the target domain. The effectiveness of the proposed methods have been evaluated on four benchmark datasets, and the experiment results show that the proposed method is superior to the existing methods, indicating that the proposed method can better solve cross-domain sentiment classification and domain adaptation problem. Although the methods have achieved good results, it does not consider the context of the sentence. The semantic loss of the sentence is still very serious. Therefore, our future plan is how to integrate the semantic information into the feature extraction process for better performance on cross-domain sentiment classification.

Acknowledgments

This work is supported by the Chinese National Science Foundation (#61763007), The Science and Technology Major Project of Guangxi Province (AA18118039-3) and the Guangxi Natural Science Foundation (#2017JJD160017).

References

[1] S. Tan, X. Cheng, Y. Wang, H. Xu, Adapting naive Bayes to domain adaptation for sentiment analysis, in: Proc. 2009 European Conference on Information Retrieval, 2009.

[2] S.J. Pan, X. Ni, J.-T. Sun, Q. Yang, Z. Chen, Cross-domain sentiment classification via spectral feature alignment, in: Proc. 2010 Proceedings of the 19th international conference on World Wide Web, 2010.

[3] W. Pan, E. Zhong, Q. Yang, Transfer learning for text mining, in: Proc. 2012 Mining Text Data, 2012.

[4] X. Glorot, A. Bordes, Y. Bengio, Domain adaptation for large-scale sentiment classification: a deep learning approach, in: Proc. 2011 Proceedings of the 28th international conference on machine learning (ICML-11), 2011.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proc. 2014 Advances in Neural Information Processing Systems, 2014.

[6] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, Domain-adversarial neural networks. <https://arxiv.org/abs/1505.07818>, 2015.

[7] M. Arjovsky, L Bottou, Towards principled methods for training generative adversarial networks. <https://arxiv.org/abs/1701.04862>, 2017.

[8] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: Proc. 2017 International Conference on Machine Learning, 2017.

[9] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, D. Erhan, Domain separation networks, in: Proc. 2016 Advances in Neural Information Processing Systems, 2016.

[10] M. Salzmann, C. Henrik Ek, R. Urtasun, T. Darrell, Factorized orthogonal latent spaces, in: Proc. the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010.

[11] J. Heinonen, Lectures on Lipschitz analysis, No. 100. University of Jyväskylä, 2005.

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of wasserstein gans, in: Proc. 2017 Advances in Neural Information Processing Systems, 2017.

[13] Y. Ganin, Vi. Lempitsky, Unsupervised domain adaptation by backpropagation. <https://arxiv.org/abs/1409.7495>, 2014.

[14] S. Ruder, An overview of gradient descent optimization algorithms. <https://arxiv.org/abs/1609.04747>, 2016.

[15] M. Chen, Z. Xu, K. Weinberger, F. Sha, Marginalized denoising autoencoders for domain adaptation. <https://arxiv.org/abs/1206.4683>, 2012.

[16] L. van der Maaten, G. Hinton. Visualizing data using t-SNE, Journal of Machine Learning Research 9(2008) 2579-2605.