Hybrid Deep Belief Networks for Semi-supervised Sentiment Classification

Shusen Zhou[⋆] **Qingcai Chen**[†] **Xiaolong Wang**[†] **Xiaoling Li**[⋆]
[⋆] School of Information and Electrical Engineering, Ludong University, Yantai 264025, China.
[†] Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China.
zhoushusen@gmail.com, qingcai.chen@hitsz.edu.cn
wangxl@insun.hit.edu.cn, appleling@live.cn

Abstract

In this paper, we develop a novel semi-supervised learning algorithm called hybrid deep belief networks (HDBN), to address the semi-supervised sentiment classification problem with deep learning. First, we construct the previous several hidden layers using restricted Boltzmann machines (RBM), which can reduce the dimension and abstract the information of the reviews quickly. Second, we construct the following hidden layers using convolutional restricted Boltzmann machines (CRBM), which can abstract the information of reviews effectively. Third, the constructed deep architecture is fine-tuned by gradient-descent based supervised learning with an exponential loss function. We did several experiments on five sentiment classification datasets, and show that HDBN is competitive with previous semi-supervised learning algorithm. Experiments are also conducted to verify the effectiveness of our proposed method with different number of unlabeled reviews.

1 Introduction

Recently, more and more people write reviews and share opinions on the World Wide Web, which present a wealth of information on products and services (Liu et al., 2010). These reviews will not only help other users make better judgements but they are also useful resources for manufacturers of products to keep track and manage customer opinions (Wei and Gulla, 2010). However, there are large amount of reviews for every topic, it is difficult for a user to manually learn the opinions of an interesting topic. Sentiment classification, which aims to classify a text according to the expressed sentimental polarities of opinions such as '*positive*' or '*negtive*', '*thumb up*' or '*thumb down*', '*favorable*' or '*unfavorable*' (Li et al., 2010), can facilitate the investigation of corresponding products or services.

In order to learn a good text classifier, a large number of labeled reviews are often needed for training (Zhen and Yeung, 2010). However, labeling reviews is often difficult, expensive or time consuming (Chapelle et al., 2006). On the other hand, it is much easier to obtain a large number of unlabeled reviews, such as the growing availability and popularity of online review sites and personal blogs (Pang and Lee, 2008). In recent years, a new approach called semi-supervised learning, which uses large amount of unlabeled data together with labeled data to build better learners (Zhu, 2007), has been developed in the machine learning community.

There are several works have been done in semi-supervised learning for sentiment classification, and get competitive performance (Li et al., 2010; Dasgupta and Ng, 2009; Zhou et al., 2010). However, most of the existing semi-supervised learning methods are still far from satisfactory. As shown by several researchers (Salakhutdinov and Hinton, 2007; Hinton et al., 2006), deep architecture, which composed of multiple levels of non-linear operations, is expected to perform well in semi-supervised learning because of its capability of modeling hard artificial intelligent tasks. Deep belief networks (DBN) is a representative deep learning algorithm achieving notable success for text classification, which is a directed belief nets with many hidden layers constructed by restricted Boltzmann machines (RBM), and refined by a gradient-descent based supervised learning (Hinton et al., 2006). Ranzato and Szummer (Ranzato and

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Page numbers and proceedings footer are added by the organisers. Licence details: http://creativecommons.org/licenses/by/4.0/

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1341–1349, Dublin, Ireland, August 23-29 2014.

Szummer, 2008) propose an algorithm to learn text document representations based on semi-supervised auto-encoders that are combined to form a deep network. Zhou et al. (Zhou et al., 2010) propose a novel semi-supervised learning algorithm to address the semi-supervised sentiment classification problem with active learning. The key issue of traditional DBN is the efficiency of RBM training. Convolutional neural networks (CNN), which are specifically designed to deal with the variability of two dimensional shapes, have had great success in machine learning tasks and represent one of the early successes of deep learning (LeCun et al., 1998). Desjardins and Bengio (Desjardins and Bengio, 2008) adapt RBM to operate in a convolutional manner, and show that the convolutional RBM (CRBM) are more efficient than standard RBM.

CRBM has been applied successfully to a wide range of visual and audio recognition tasks (Lee et al., 2009a; Lee et al., 2009b). Though the success of CRBM in addressing two dimensional issues, there is still no published research on the using of CRBM in textual information processing. In this paper, we propose a novel semi-supervised learning algorithm called hybrid deep belief networks (HDBN), to address the semi-supervised sentiment classification problem with deep learning. HDBN is a hybrid of RBM and CRBM deep architecture, the bottom layers are constructed by RBM, and the upper layers are constructed by CRBM, then the whole constructed deep architecture is fine tuned by a gradient-descent based supervised learning based on an exponential loss function.

The remainder of this paper is organized as follows. In Section 2, we introduce our semi-supervised learning method HDBN in details. Extensive empirical studies conducted on five real-world sentiment datasets are presented in Section 3. Section 4 concludes our paper.

2 Hybrid deep belief networks

2.1 Problem formulation

The sentiment classification dataset composed of many review documents, each review document composed of a bag of words. To classify these review documents using corpus-based approaches, we need to preprocess them in advance. The preprocess method for these reviews is similar with (Zhou et al., 2010). We tokenize and downcase each review and represent it as a vector of unigrams, using binary weight equal to 1 for terms present in a vector. Moreover, the punctuations, numbers, and words of length one are removed from the vector. Finally, we combine all the words in the dataset, sort the vocabulary by document frequency and remove the top 1.5%, because many of these high document frequency words are stopwords or domain specific general-purpose words.

After preprocess, each review can be represented as a vector of binary weight $\mathbf{x}^i$. If the j^{th} word of the vocabulary is in the i^{th} review, $\mathbf{x}^i_j = 1$; otherwise, $\mathbf{x}^i_j = 0$. Then the dataset can be represented as a matrix:

$$\mathbf{X} = \left[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{R+T}\right] = \left[\begin{array}{c} x_1^1,\, x_1^2,\, \ldots, x_1^{R+T} \\ x_2^1,\, x_2^2,\, \ldots,\, x_2^{R+T} \\ \vdots\,,\, \vdots\,,\, \ldots,\, \vdots \\ x_D^1, x_D^2, \ldots, x_D^{R+T} \end{array} \right] \tag{1}$$

where R is the number of training reviews, T is the number of test reviews, D is the number of feature words in the dataset. Every column of $\mathbf{X}$ corresponds to a sample $\mathbf{x}$, which is a representation of a review. A sample that has all features is viewed as a vector in $\mathbb{R}^D$, where the i^{th} coordinate corresponds to the i^{th} feature.

The L labeled reviews are chosen randomly from R training reviews, or chosen actively by active learning, which can be seen as:

$$\mathbf{X}^L = \mathbf{X}^R\left(\mathbf{S}\right),\ \mathbf{S} = [s_1,\ ...,\ s_L],\ 1 \leq s_i \leq R \tag{2}$$

where $\mathbf{S}$ is the index of selected training reviews to be labeled manually.

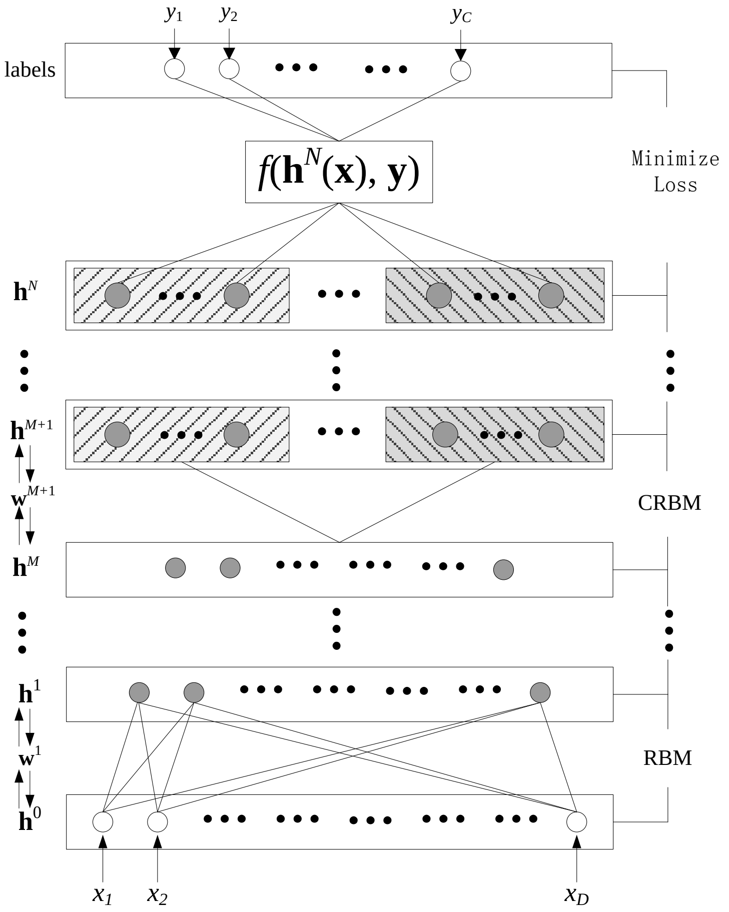

Figure 1: Architecture of HDBN.

The L labels correspond to L labeled training reviews is denoted as:

$$\mathbf{Y}^L = \left[\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^L\right] = \begin{bmatrix} y_1^1, \, y_1^2, \, \ldots, y_1^L \\ y_2^1, \, y_2^2, \, \ldots, \, y_2^L \\ \vdots \, , \, \vdots \, , \ldots, \, \vdots \\ y_C^1, y_C^2, \ldots, y_C^L \end{bmatrix} \tag{3}$$

where C is the number of classes. Every column of $\mathbf{Y}$ is a vector in $\mathbb{R}^C$, where the j^{th} coordinate corresponds to the j^{th} class.

$$y_j^i = \begin{cases} 1 & \text{if } \mathbf{x}^i \in j^{\text{th}} \text{ class} \\ -1 & \text{if } \mathbf{x}^i \notin j^{\text{th}} \text{ class} \end{cases} \tag{4}$$

For example, if a review $\mathbf{x}^i$ is positive, $\mathbf{y}^i = [1, -1]'$; otherwise, $\mathbf{y}^i = [-1, 1]'$.

We intend to seek the mapping function $\mathbf{X} \rightarrow \mathbf{Y}$ using the L labeled data and all unlabeled data. After training, we can determine **y** using the mapping function when a new sample **x** comes.

2.2 Architecture of HDBN

In this part, we propose a novel semi-supervised learning method HDBN to address the problem formulated in Section 2.1. The sentiment datasets have high dimension (about 10,000), and computation complexity of convolutional calculation is relatively high, so we use RBM to reduce the dimension of review with normal calculation firstly. Fig. 1 shows the deep architecture of HDBN, a fully interconnected directed belief nets with one input layer $\mathbf{h}^0$, N hidden layers $\mathbf{h}^1, \mathbf{h}^2, ..., \mathbf{h}^N$, and one label layer at the top. The input layer $\mathbf{h}^0$ has D units, equal to the number of features of sample review **x**. The hidden

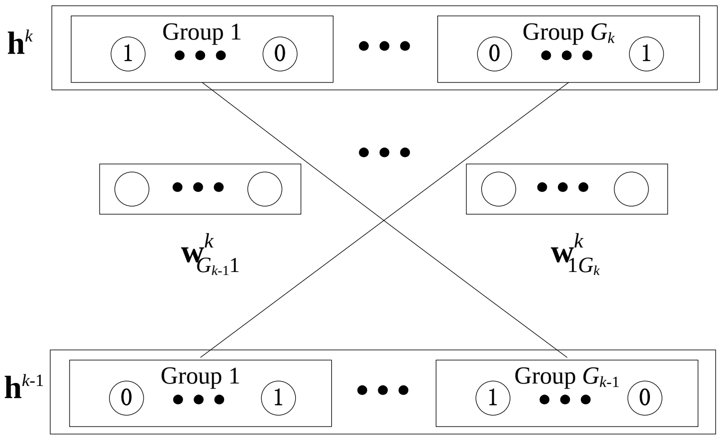

Figure 2: Architecture of CRBM.

layer has M layers constructed by RBM and $N - M$ layers constructed by CRBM. The label layer has C units, equal to the number of classes of label vector **y**. The numbers of hidden layers and the number of units for hidden layers, currently, are pre-defined according to the experience or intuition. The seeking of the mapping function $\mathbf{X} \rightarrow \mathbf{Y}$, here, is transformed to the problem of finding the parameter space $\mathbf{W} = \{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^N\}$ for the deep architecture.

The training of the HDBN can be divided into two stages:

1. HDBN is constructed by greedy layer-wise unsupervised learning using RBMs and CRBMs as building blocks. L labeled data and all unlabeled data are utilized to find the parameter space **W** with N layers.

2. HDBN is trained according to the exponential loss function using gradient descent based supervised learning. The parameter space **W** is refined using L labeled data.

2.3 Unsupervised learning

As show in Fig. 1, we construct HDBN layer by layer using RBMs and CRBMs, the details of RBM can be seen in (Hinton et al., 2006), and CRBM is introduced below.

The architecture of CRBM can be seen in Fig. 2, which is similar to RBM, a two-layer recurrent neural network in which stochastic binary input groups are connected to stochastic binary output groups using symmetrically weighted connections. The top layer represents a vector of stochastic binary hidden feature $\mathbf{h}^k$ and the bottom layer represents a vector of binary visible data $\mathbf{h}^{k-1}$, $k = M+1, ..., N$. The k^{th} layer consists of G_k groups, where each group consists of D_k units, resulting in $G_k \times D_k$ hidden units. The layer $\mathbf{h}^M$ is consist of 1 group and D_M units. $\mathbf{w}^k$ is the symmetric interaction term connecting corresponding groups between data $\mathbf{h}^{k-1}$ and feature $\mathbf{h}^k$. However, comparing with RBM, the weights of CRBM between the hidden and visible groups are shared among all locations (Lee et al., 2009a), and the calculation is operated in a convolutional manner (Desjardins and Bengio, 2008).

We define the energy of the state $(\mathbf{h}^{k-1}, \mathbf{h}^k)$ as:

$$E\left(\mathbf{h}^{k-1}, \mathbf{h}^k; \theta\right) = -\sum_{s=1}^{G_{k-1}} \sum_{t=1}^{G_k} (\tilde{w}^k_{st} * h^{k-1}_s) \bullet h^k_t - \sum_{s=1}^{G_{k-1}} b^{k-1}_s \sum_{u=1}^{D_{k-1}} h^{k-1}_s - \sum_{t=1}^{G_k} c^k_t \sum_{v=1}^{D_k} h^k_t \qquad (5)$$

where $\theta = (\mathbf{w}, \mathbf{b}, \mathbf{c})$ are the model parameters: w^k_{st} is a filter between unit s in the layer $\mathbf{h}^{k-1}$ and unit t in the layer $\mathbf{h}^k$, $k = M+1, ..., N$. The dimension of the filter w^k_{st} is equal to $D_{k-1} - D_k + 1$. b^{k-1}_s is the s^{th} bias of layer $\mathbf{h}^{k-1}$ and c^k_t is the t^{th} bias of layer $\mathbf{h}^k$. A tilde above an array ($\tilde{w}$) denote flipping the array, $*$ denote valid convolution, and $\bullet$ denote element-wise product followed by summation, i.e., $A \bullet B = trA^TB$ (Lee et al., 2009a).

Similar to RBM, Gibbs sampler can be performed based on the following conditional distribution.

The probability of turning on unit v in group t is a logistic function of the states of $\mathbf{h}^{k-1}$ and w^k_{st}:

$$p\left(h^k_{t,v}=1|\mathbf{h}^{k-1}\right)=\mathrm{sigm}\left(c^k_t+(\sum_s \tilde{w}^k_{st}*h^{k-1}_s)_v\right) \tag{6}$$

The probability of turning on unit u in group s is a logistic function of the states of $\mathbf{h}^k$ and w^k_{st}:

$$p\left(h^{k-1}_{s,u}=1|\mathbf{h}^{k}\right)=\mathrm{sigm}\left(b^{k-1}_s+(\sum_t w^k_{st}\star h^{k}_t)_u\right) \tag{7}$$

A star $\star$ denote full convolution.

2.4 Supervised learning

In HDBN, we construct the deep architecture using all labeled reviews with unlabeled reviews by inputting them one by one from layer $\mathbf{h}^0$. The deep architecture is constructed layer by layer from bottom to top, and each time, the parameter space $\mathbf{w}^k$ is trained by the calculated data in the $k-1^{th}$ layer.

Algorithm 1: Algorithm of HDBN

```
Input: data $\mathbf{X}$, $\mathbf{Y}^L$
number of training data $R$; number of test data $T$;
number of layers $N$; number of epochs $Q$;
number of units in every hidden layer $D_1 ... D_N$;
number of groups in every convolutional hidden layer $G_M ... G_N$;
hidden layer $\mathbf{h}^1, \ldots, \mathbf{h}^M$;
convolutional hidden layer $\mathbf{h}^{M+1}, \ldots, \mathbf{h}^{N-1}$;
parameter space $\mathbf{W} = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$;
biases $\mathbf{b}$, $\mathbf{c}$; momentum $\vartheta$ and learning rate $\eta$;
Output: deep architecture with parameter space $\mathbf{W}$
1. Greedy layer-wise unsupervised learning
for $k = 1; k \leq N-1$ do
    for $q = 1; q \leq Q$ do
        for $r = 1; r \leq R+T$ do
            Calculate the non-linear positive and negative phase:
            if $k \leq M$ then
                Normal calculation.
            else
                Convolutional calculation according to Eq. 6 and Eq. 7.
            end
            Update the weights and biases:
            $w^k_{st} = \vartheta w^k_{st} + \eta \left( \langle h^{k-1}_{s,r} h^k_{t,r} \rangle_{P_0} - \langle h^{k-1}_{s,r} h^k_{t,r} \rangle_{P_1} \right)$
        end
    end
end
2. Supervised learning based on gradient descent
$\arg\min_{W} \sum_{i=1}^{L} \sum_{j=1}^{C} \exp(-h^N(x^i_j) y^i_j)$
```

According to the $\mathbf{w}^k$ calculated by RBM and CRBM, the layer $\mathbf{h}^k, k=1,\ldots,M$ can be computed as following when a sample $\mathbf{x}$ inputs from layer $\mathbf{h}^0$:

$$h^k_t(\mathbf{x})=\mathrm{sigm}\left(c^k_t+\sum_{s=1}^{D_{k-1}} w^k_{st}h^{k-1}_s(\mathbf{x})\right), t=1,\ldots,D_k \tag{8}$$

When $k = M + 1, \ldots, N - 1$, the layer $\mathbf{h}^k$ can be represented as:

$$h_t^k(\mathbf{x}) = \text{sigm}\left(c_t^k + \sum_{s=1}^{G_{k-1}} \tilde{w}_{st}^k * h_s^{k-1}(\mathbf{x})\right), t = 1, \ldots, G_k \tag{9}$$

The parameter space $\mathbf{w}^N$ is initialized randomly, just as backpropagation algorithm.

$$h_t^N(\mathbf{x}) = c_t^N + \sum_{s=1}^{G_{N-1} \times D_{N-1}} w_{st}^N h_s^{N-1}(\mathbf{x}),\ t = 1, \ldots, D_N \tag{10}$$

After greedy layer-wise unsupervised learning, $\mathbf{h}^N(\mathbf{x})$ is the representation of $\mathbf{x}$. Then we use L labeled reviews to refine the parameter space $\mathbf{W}$ for better discriminative ability. This task can be formulated as an optimization problem:

$$\arg\min_{\mathbf{W}} \quad \text{f}\left(h^N\left(\mathbf{X}^L\right), \mathbf{Y}^L\right) \tag{11}$$

where

$$\text{f}\left(h^N\left(\mathbf{X}^L\right), \mathbf{Y}^L\right) = \sum_{i=1}^{L}\sum_{j=1}^{C} \text{T}\left(h_j^N\left(\mathbf{x}^i\right) y_j^i\right) \tag{12}$$

and the loss function is defined as

$$T(r) = \exp(-r) \tag{13}$$

We use gradient-descent through the whole HDBN to refine the weight space. In the supervised learning stage, the stochastic activities are replaced by deterministic, real valued probabilities.

2.5 Classification using HDBN

The training procedure of HDBN is given in Algorithm 1. For the training of HDBN architecture, the parameters are random initialized with normal distribution. All the reviews in the dataset are used to train the HDBN with unsupervised learning. After training, we can determine the label of the new data through:

$$\arg\max_j h^N(\mathbf{x}) \tag{14}$$

3 Experiments

3.1 Experimental setup

We evaluate the performance of the proposed HDBN method using five sentiment classification datasets. The first dataset is MOV (Pang et al., 2002), which is a classical movie review dataset. The other four datasets contain products reviews come from the multi-domain sentiment classification corpus, including books (BOO), DVDs (DVD), electronics (ELE), and kitchen appliances (KIT) (Blitzer et al., 2007). Each dataset contains 1,000 positive and 1,000 negative reviews.

The experimental setup is same as (Zhou et al., 2010). We divide the 2,000 reviews into ten equal-sized folds randomly, maintaining balanced class distributions in each fold. Half of the reviews in each fold are random selected as training data and the remaining reviews are used for test. Only the reviews in the training data set are used for the selection of labeled reviews by active learning. All the algorithms are tested with cross-validation.

We compare the classification performance of HDBN with four representative semi-supervised learning methods, i.e., semi-supervised spectral learning (Spectral) (Kamvar et al., 2003), transductive SVM (TSVM) (Collobert et al., 2006), deep belief networks (DBN) (Hinton et al., 2006), and personal/impersonal views (PIV) (Li et al., 2010). Spectral learning, TSVM methods are two baseline methods for sentiment classification. DBN (Hinton et al., 2006) is the classical deep learning method proposed recently. PIV (Li et al., 2010) is a new sentiment classification method proposed recently.

Table 1: HDBN structure used in experiment.

Dataset	Structure
MOV	100-100-4-2
KIT	50-50-3-2
ELE	50-50-3-2
BOO	50-50-5-2
DVD	50-50-5-2

Table 2: Test accuracy with 100 labeled reviews for semi-supervised learning.

Type	MOV	KIT	ELE	BOO	DVD
Spectral	67.3	63.7	57.7	55.8	56.2
TSVM	68.7	65.5	62.9	58.7	57.3
DBN	71.3	72.6	73.6	64.3	66.7
PIV	-	**78.6**	70.0	60.1	49.5
HDBN	**72.2**	74.8	**73.8**	**66.0**	**70.3**

3.2 Performance of HDBN

The HDBN architecture used in all our experiments have 2 normal hidden layer and 1 convolutional hidden layer, every hidden layer has different number of units for different sentiment datasets. The deep structure used in our experiments for different datasets can be seen in Table 1. For example, the HDBN structure used in MOV dataset experiment is 100-100-4-2, which represents the number of units in 2 normal hidden layers are 100, 100 respectively, and in output layer is 2, the number of groups in 1 convolutional hidden layer is 4. The number of unit in input layer is the same as the dimensions of each datasets. For greedy layer-wise unsupervised learning, we train the weights of each layer independently with the fixed number of epochs equal to 30 and the learning rate is set to 0.1. The initial momentum is 0.5 and after 5 epochs, the momentum is set to 0.9. For supervised learning, we run 30 epochs, three times of linear searches are performed in each epoch.

The test accuracies in cross validation for five datasets and five methods with semi-supervised learning are shown in Table 2. The results of previous two methods are reported by (Dasgupta and Ng, 2009). The results of DBN method are reported by (Zhou et al., 2010). Li et al. (Li et al., 2010) reported the results of PIV method. The result of PIV on MOV dataset is empty, because (Li et al., 2010) did not report it. HDBN is the proposed method.

Through Table 2, we can see that HDBN gets most of the best results except on KIT dataset, which is just slight worse than PIV method. However, the preprocess of PIV method is much more complicated than HDBN, and the PIV results on other datasets are much worse than HDBN method. HDBN method is adjusted by DBN, all the experiment results on five datasets for HDBN are better than DBN. This could be contributed by the convolutional computation in HDBN structure, and proves the effectiveness of our proposed method.

3.3 Performance with variance of unlabeled data

To verify the contribution of unlabeled reviews for our proposed method, we did several experiments with fewer unlabeled reviews and 100 labeled reviews.

The test accuracies of HDBN with different number of unlabeled reviews and 100 labeled reviews on five datasets are shown in Fig. 3. The architectures for HDBN used in this experiment are same as Section 3.2 too, which can be seen in Table 1. We can see that the performance of HDBN is much worse when just using 400 unlabeled reviews. However, when using more than 1200 unlabeled reviews, the performance of HDBN is improved obviously. For most of review datasets, the accuracy of HDBN with 1200 unlabeled reviews is close to the accuracy with 1600 and 2000 unlabeled reviews. This proves that HDBN can get competitive performance with just few labeled reviews and appropriate number of

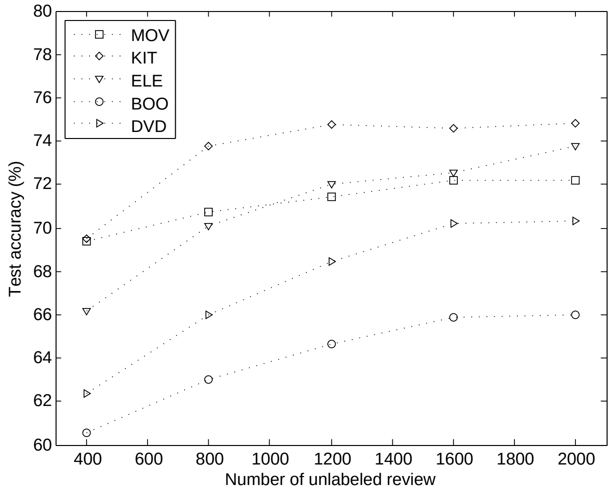

Figure 3: Test accuracy of HDBN with different number of unlabeled reviews on five datasets.

unlabeled reviews. Considering the much time needed for training with more unlabeled reviews and less accuracy improved for HDBN method, we suggest using appropriate number of unlabeled reviews in real application.

4 Conclusions

In this paper, we propose a novel semi-supervised learning method, HDBN, to address the sentiment classification problem with a small number of labeled reviews. HDBN seamlessly incorporate convolutional computation into the DBN architecture, and use CRBM to abstract the review information effectively. To the best of our knowledge, HDBN is the first work that uses convolutional neural network to improve sentiment classification performance. One promising property of HDBN is that it can effectively use the distribution of large amount of unlabeled data, together with few label information in a unified framework. In particular, HDBN can greatly reduce the dimension of reviews through RBM and abstract the information of reviews through the cooperate of RBM and CRBM. Experiments conducted on five sentiment datasets demonstrate that HDBN outperforms state-of-the-art semi-supervised learning algorithms, such as SVM and DBN based methods, using just few labeled reviews, which demonstrate the effective of deep architecture for sentiment classification.

Acknowledgements

This work is supported in part by National Natural Science Foundation of China (No. 61300155, No. 61100115 and No. 61173075), Natural Science Foundation of Shandong Province (No. ZR2012FM008), Science and Technology Development Plan of Shandong Province (No. 2013GNC11012), Science and Technology Research and Development Funds of Shenzhen City (No. JC201005260118A and No. JC201005260175A), and Scientific Research Fund of Ludong University (LY2013004).

References

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2006. *Semi-supervised learning*. MIT Press, USA.

Ronan Collobert, Fabian Sinz, Jason Weston, and Leon Bottou. 2006. Large scale transductive svms. *Journal of Machine Learning Research*, 7:1687–1712.

Sajib Dasgupta and Vincent Ng. 2009. Mine the easy, classify the hard: A semi-supervised approach to automatic sentiment classfication. In *Joint Conference of the 47th Annual Meeting of the Association for Computational*

Linguistics and 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, pages 701–709, Stroudsburg, PA, USA. Association for Computational Linguistics.

Guillaume Desjardins and Yoshua Bengio. 2008. Empirical evaluation of convolutional rbms for vision. Technical report.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554.

Sepandar Kamvar, Dan Klein, and Christopher Manning. 2003. Spectral learning. In *International Joint Conferences on Artificial Intelligence*, pages 561–566, Catalonia, Spain. AAAI Press.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. 2009a. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning*, pages 609–616, Montreal, Canada. ACM.

Honglak Lee, Yan Largman, Peter Pham, and Andrew Y. Ng. 2009b. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems*, pages 1096–1103, Vancouver, B.C., Canada. NIPS Foundation.

Shoushan Li, Chu-Ren Huang, Guodong Zhou, and Sophia Yat Mei Lee. 2010. Employing personal/impersonal views in supervised and semi-supervised sentiment classification. In *Annual Meeting of the Association for Computational Linguistics*, pages 414–423, Uppsala, Sweden. Association for Computational Linguistics.

Yang Liu, Xiaohui Yu, Xiangji Huang, and Aijun An. 2010. S-plasa+: Adaptive sentiment analysis with application to sales performance prediction. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 873–874, New York, NY, USA. ACM.

Bo Pang and Lillian Lee. 2008. *Opinion Mining and Sentiment Analysis*, volume 2 of *Foundations and Trends in Information Retrieval*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Conference on Empirical Methods in Natural Language Processing*, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marc'Aurelio Ranzato and Martin Szummer. 2008. Semi-supervised learning of compact document representations with deep networks. In *International Conference on Machine Learning*, pages 792–799, Helsinki, Finland. ACM.

Ruslan Salakhutdinov and Geoffrey E. Hinton. 2007. Learning a nonlinear embedding by preserving class neighbourhood structure. *Journal of Machine Learning Research*, 2:412–419.

Wei Wei and Jon Atle Gulla. 2010. Sentiment learning on product reviews via sentiment ontology tree. In *Annual Meeting of the Association for Computational Linguistics*, pages 404–413, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yi Zhen and Dit-Yan Yeung. 2010. Sed: Supervised experimental design and its application to text classification. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 299–306, Geneva, Switzerland. ACM.

Shusen Zhou, Qingcai Chen, and Xiaolong Wang. 2010. Active deep networks for semi-supervised sentiment classification. In *International Conference on Computational Linguistics*, pages 1515–1523, Beijing, China. Coling 2010 Organizing Committee.

Xiaojin Zhu. 2007. *Semi-supervised learning literature survey*. Ph.D. thesis.