OPEN ACCESS Freely available online

PLOS ONE

Active Semi-Supervised Learning Method with Hybrid Deep Belief Networks

Shusen Zhou[1]*, Qingcai Chen[2], Xiaolong Wang[2]

1 School of Information and Electrical Engineering, Ludong University, Yantai, Shandong, China, **2** Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, Guangdong, China

Abstract

In this paper, we develop a novel semi-supervised learning algorithm called active hybrid deep belief networks (AHD), 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. Finally, active learning method is combined based on the proposed deep architecture. We did several experiments on five sentiment classification datasets, and show that AHD is competitive with previous semi-supervised learning algorithm. Experiments are also conducted to verify the effectiveness of our proposed method with different number of labeled reviews and unlabeled reviews respectively.

Citation: Zhou S, Chen Q, Wang X (2014) Active Semi-Supervised Learning Method with Hybrid Deep Belief Networks. PLoS ONE 9(9): e107122. doi:10.1371/journal.pone.0107122

Editor: Catalin Buiu, Politehnica University of Bucharest, Romania

Received June 5, 2014; **Accepted** August 7, 2014; **Published** September 10, 2014

Data Availability: The authors confirm that all data underlying the findings are fully available without restriction. All relevant data are within the paper.

Funding: This work is supported in part by National Natural Science Foundation of China (No. 61300155), and Scientific Research Fund of Ludong University (LY2013004). Shusen Zhou received the funding from Scientific Research Fund of Ludong University. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* Email: zhoushusen@gmail.com

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 [1]. 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 [2]. However, there are large amounts 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 '*negative*', '*thumb up*' or '*thumb down*', '*favorable*' or '*unfavorable*' [3], 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 [4]. However, labeling reviews is often difficult, expensive or time consuming [5]. 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 [6]. 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 [7], has been developed in the machine learning community.

There are several works have been done in semi-supervised learning for sentiment classification, and have get competitive performance [3,8–10]. However, most of the existing semi-supervised learning methods are still far from satisfactory. As shown by several researchers [11,12], 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 [12]. Ranzato and Szummer [13] 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. [10] propose a novel semi-supervised learning algorithm to address the semi-supervised sentiment classification problem with active learning. Socher et al. [14] introduce a novel machine learning framework based on recursive autoencoders for sentence-level prediction of sentiment label distributions. Socher et al. [15] introduce the recursive neural tensor network for semantic compositionality over a sentiment treebank. 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 [16]. Desjardins and Bengio [17] 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 [18,19]. Though the success of CRBM in addressing two dimensional issues, there is still no published

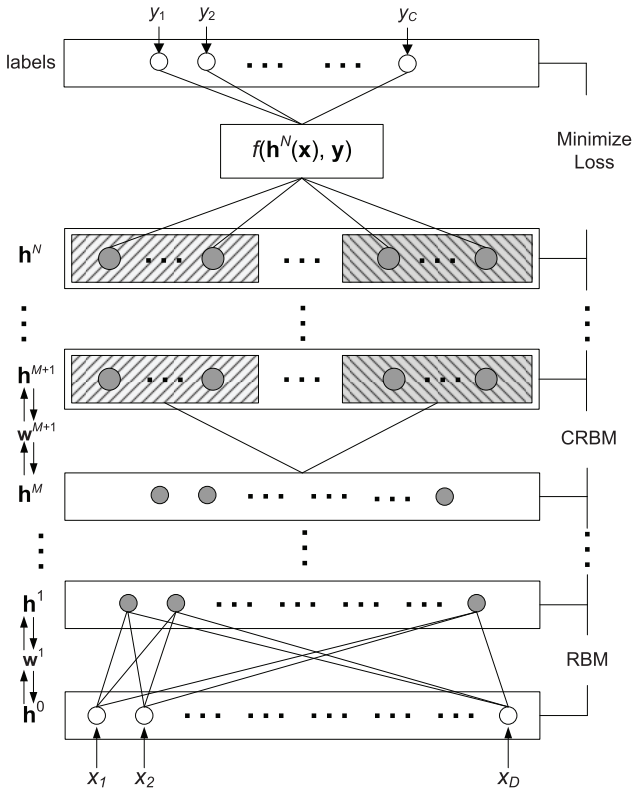

Figure 1. Architecture of HDBN.
doi:10.1371/journal.pone.0107122.g001

research on the using of CRBM in textual information processing. In this paper, we propose a novel semi-supervised learning algorithm called active hybrid deep belief networks (AHD), to address the semi-supervised sentiment classification problem with deep learning. AHD is an active learning method based on deep architecture, which 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.

Hybrid Deep Belief Networks Method

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 [9,10]. 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:

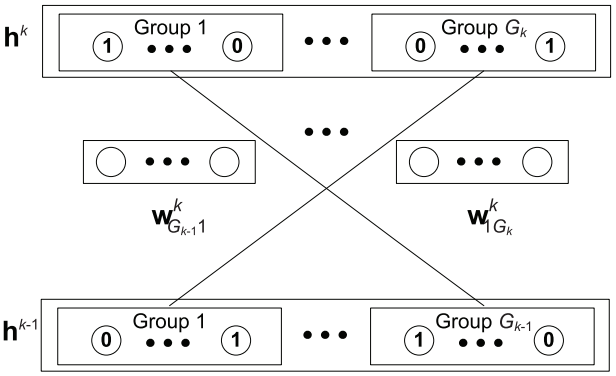

Figure 2. Architecture of CRBM.
doi:10.1371/journal.pone.0107122.g002

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

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:

$$\mathbf{X}^L = \mathbf{X}^R(\mathbf{S}), \mathbf{S} = [s_1, \ldots, s_L], 1 \le s_i \le R \tag{2}$$

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

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

Architecture of HDBN

In this part, we propose a novel semi-supervised learning method HDBN to address the sentiment classification problem. 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 $\mathbf{x}$. The hidden 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 $\mathbf{y}$. 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 $\mathbf{W}$ with N layers.

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

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 [12]. 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 [18], and the calculation is operated in a convolutional manner [17].

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

$$E(\mathbf{h}^{k-1},\mathbf{h}^k;\theta)= -\sum_{s=1}^{G_{k-1}}\sum_{t=1}^{G_k}(\tilde{w}_{st}^k * h_s^{k-1})\bullet h_t^k-\sum_{s=1}^{G_{k-1}} b_s^{k-1}\sum_{u=1}^{D_{k-1}} h_s^{k-1}-\sum_{t=1}^{G_k} c_t^k\sum_{v=1}^{D_k} h_t^k \quad (5)$$

where $\theta=(\mathbf{w},\mathbf{b},\mathbf{c})$ are the model parameters: w_{st}^k 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_{st}^k is equal to $D_{k-1}-D_k+1$. b_s^{k-1} is the s^{th} bias of layer $\mathbf{h}^{k-1}$ and c_t^k 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., $A\bullet B=trA^TB$ [18].

Gibbs sampler can be performed based on the following conditional distribution.

Table 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 ϑ and learning rate η;

Output:

deep architecture with parameter space $\mathbf{W}$

1. Greedy layer-wise unsupervised learning

for $k=1;\ k<N-1;\ k++$ **do**

  for $q=1;\ q\le Q;\ q++$ **do**

    for $r=1;\ r\le R+T;\ r++$ **do**

      Calculate the non-linear positive and negative phase:

      if $k\le M$ **then**

        Normal calculation.

      else

        Convolutional calculation according to Eq. 6 and Eq. 7.

      end if

      Update the weights and biases:

      $w_{st}^k=\vartheta w_{st}^k+\eta\left(\langle h_{s,r}^{k-1}h_{t,r}^k\rangle_{P_0}-\langle h_{s,r}^{k-1}h_{t,r}^k\rangle_{P_1}\right)$

    end for

  end for

end for

2. Supervised learning based on gradient descent

$$\arg\min_W \sum_{i=1}^{L}\sum_{j=1}^{C}\exp(-h^N(x_j^i)y_j^i)$$

doi:10.1371/journal.pone.0107122.t001

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

$$p\left(h_{t,v}^k=1|\mathbf{h}^{k-1}\right)=\mathrm{sigm}\left(c_t^k+\left(\sum_s \tilde{w}_{st}^k * h_s^{k-1}\right)_v\right) \quad (6)$$

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

$$p\left(h_{s,u}^{k-1}=1|\mathbf{h}^k\right)=\mathrm{sigm}\left(b_s^{k-1}+\left(\sum_t w_{st}^k \star h_t^k\right)_u\right) \quad (7)$$

where the logistic function is:

$$\mathrm{sigm}(\eta)=1/(1+e^{-\eta}) \quad (8)$$

A star $\star$ denotes full convolution.

Table 2. Algorithm of AHD.

Input:
data $\mathbf{X}$, $(\mathbf{X}^L,\mathbf{Y}^L)$ (one positive and one negative)
number of training data R
number of iterations I
number of active choosing data for every iteration U
parameter space $\mathbf{W}=\{\mathbf{w}^1,\ldots,\mathbf{w}^N\}$
Output:
deep architecture with parameter space $\mathbf{W}$
for $i=1; i\leq I; i++$ **do**
Train HDBN with labeled dataset $\mathbf{X}^L$ and all unlabeled data in $\mathbf{X}$.
Choose U reviews which near the separating line from train dataset $\mathbf{X}^R$ through Eq. 17.
Add U reviews into the labeled data set $\mathbf{X}^L$.
end for
Train HDBN with labeled dataset $\mathbf{X}^L$ and all unlabeled data in $\mathbf{X}$.

doi:10.1371/journal.pone.0107122.t002

The convolution computation can extract the information of text effectively based on deep architecture, although it needs more computation time.

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.

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 $\mathbf{h}^0$:

$$h_t^k(\mathbf{x})=\mathrm{sigm}\left(c_t^k+\sum_{s=1}^{D_{k-1}} w_{st}^k h_s^{k-1}(\mathbf{x})\right),t=1,\ldots,D_k \qquad (9)$$

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

$$h_t^k(\mathbf{x})=\mathrm{sigm}\left(c_t^k+\sum_{s=1}^{G_{k-1}} \tilde{w}_{st}^k * h_s^{k-1}(\mathbf{x})\right),t=1,\ldots,G_k \qquad (10)$$

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

$$h_t^N(\mathbf{x})=c_t^N+\sum_{s=1}^{G_{N-1}\times D_{N-1}} w_{st}^N h_s^{N-1}(\mathbf{x}),\ t=1,\ldots,D_N \qquad (11)$$

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:

$$\arg\min_{\mathbf{W}} \quad \mathrm{f}\left(h^N(\mathbf{X}^L),\mathbf{Y}^L\right) \qquad (12)$$

where

$$\mathrm{f}\left(h^N(\mathbf{X}^L),\mathbf{Y}^L\right)=\sum_{i=1}^{L}\sum_{j=1}^{C}\mathrm{T}\left(h_j^N(\mathbf{x}^i)y_j^i\right) \qquad (13)$$

and the loss function is defined as

$$T(r)=\exp(-r) \qquad (14)$$

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.

Classification using HDBN

The training procedure of HDBN is given in Table 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:

$$\arg\max_j h^N(\mathbf{x}) \qquad (15)$$

Active Hybrid Deep Belief Networks Method

AHD description

Given an unlabeled pool $\mathbf{X}^R$ and an initial labeled data set $\mathbf{X}^L$ (one positive, one negative), the AHD architecture $\mathbf{h}^N(\mathbf{x})$ will decide which instance in $\mathbf{X}^R$ to query next. Then the parameters of $\mathbf{h}^N(\mathbf{x})$ are adjusted after new reviews are labeled and inserted into the labeled data set $\mathbf{X}^L$. We choose the reviews that are near the separating hyperplane as the labeled training data.

Table 3. 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

doi:10.1371/journal.pone.0107122.t003

When HDBN is trained by L labeled data and all unlabeled data, the parameters of deep architecture are adjusted, $\mathbf{h}^N(\mathbf{x})$ is the representation of $\mathbf{x}$. Given an unlabeled pool $\mathbf{X}^R$, the next unlabeled instance to be queried are chosen according to the location of $\mathbf{h}^N(\mathbf{X}^R)$. For review document, there are only 2 classes (*positive* or *negative*), so the dimension of $\mathbf{h}^N(\mathbf{x})$ is 2, the classes separation line is $h_1^N = h_2^N$. The distance between a point $\mathbf{h}^N(\mathbf{x}^i)$ and separation line is:

$$d(\mathbf{x}^i) = |h_1^N(\mathbf{x}^i) - h_2^N(\mathbf{x}^i)| \Big/ \sqrt{2} \tag{16}$$

The selected training reviews to be labeled manually are given by:

$$s = \{j : d(\mathbf{x}^j) = \min(d(\mathbf{X}^R))\} \tag{17}$$

Classification using AHD

The training procedure of AHD is given in Table 2. The training set $\mathbf{X}^R$ can be seen as an unlabeled pool. We randomly select one positive and one negative reviews in the pool to input as the initial labeled dataset $\mathbf{X}^L$ that are used for supervised learning. The iteration times I and the number of active choosing data U for each iteration can be set manually based on the number of labeled reviews in the experiment.

For each iteration, the HDBN architecture is trained by all the unlabeled reviews and labeled reviews in existence with unsupervised learning and supervised learning firstly. Then U reviews are chosen from the unlabeled pool based on the distance of these review mapping results from the separating line. At last, these U reviews are labeled manually and added to the labeled dataset $\mathbf{X}^L$. For the next iteration, the HDBN architecture can be re-trained by all reviews with unsupervised learning and all labeled reviews with the new increased labeled dataset $\mathbf{X}^L$. At last, HDBN architecture is retrained by all the reviews with unsupervised learning and existing labeled reviews with supervised learning.

After active training, we can use the Eq. 15 to determine the label of the new data. The purpose of active learning is choose more useful label data to train the deep architecture, which can use fewer label data to train better classifier.

Experiments

Experimental setup

We evaluate the performance of the proposed HDBN and AHD method using five sentiment classification datasets. The first dataset is MOV [20], 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) [21]. Each dataset contains 1,000 positive and 1,000 negative reviews.

The experimental setup is same as [9] and [10]. 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) [22], transductive SVM (TSVM) [23], deep belief networks (DBN) [12], and personal/impersonal views (PIV) [3]. Spectral learning, TSVM methods are two baseline methods for sentiment classification. DBN [12] is the classical deep learning method proposed recently. PIV [3] is a new sentiment classification method proposed recently.

We also compare the classification performance of AHD with three representative active semi-supervised learning methods, i.e., active learning (Active) [24], mine the easy classify the hard

Table 4. 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**

doi:10.1371/journal.pone.0107122.t004

Table 5. Test accuracy with 100 labeled reviews for active semi-supervised learning.

Type	MOV	KIT	ELE	BOO	DVD
Active	68.9	68.1	63.3	58.6	58.0
MECH	76.2	74.1	70.6	62.1	62.7
ADN	**76.3**	**77.5**	**76.8**	69.0	71.6
AFD	75	77	**76.8**	**70.1**	**73.7**

doi:10.1371/journal.pone.0107122.t005

(MECH) [9], and active deep networks (ADN) [10]. Active learning [24] is a baseline active learning method for sentiment classification. MECH [9] and ADN [10] are two new active learning method for sentiment classification proposed recently.

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 3. 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 4. The results of previous two methods are reported by [9]. The results of DBN method are reported by [10]. Li et al. [3] reported the results of PIV method. The result of PIV on MOV dataset is empty, because [3] did not report it. HDBN is the proposed method.

Through Table 4, 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.

Performance of AHD

To evaluate the performance of AHD, we compare its results with several previous active learning methods for sentiment classification. The architectures used in this experiments can be seen in Table 3. We perform active learning for 5 iterations. In each iteration, we select and label 20 of the most uncertain reviews, and then retrain the deep architecture on all of the unlabeled reviews and labeled reviews annotated so far. After 5 iterations, 100 labeled reviews are used for training.

The test accuracies in cross validation for five datasets and four methods with active semi-supervised learning are shown in Table 5. The results of previous two methods are reported by

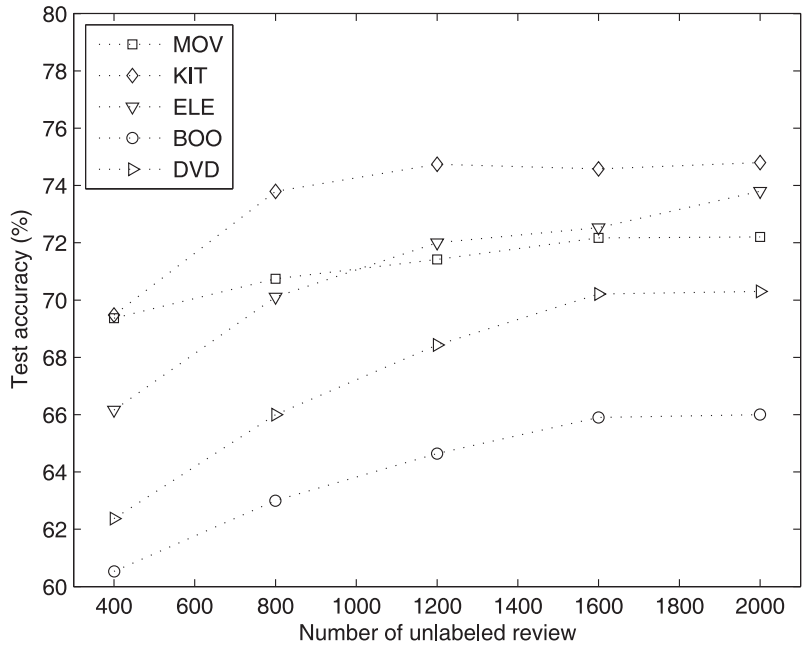

Figure 3. Test accuracy of HDBN with different number of unlabeled reviews on five datasets.
doi:10.1371/journal.pone.0107122.g003

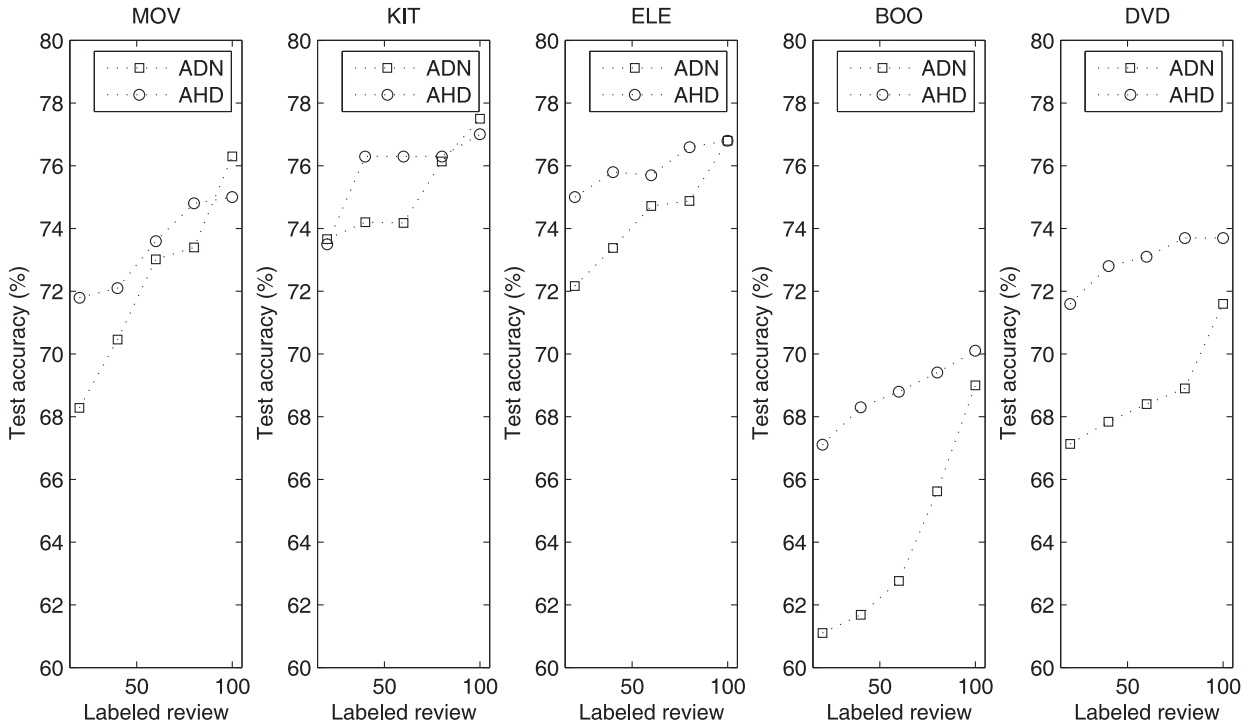

Figure 4. Test accuracy of ADN and AHD with different number of labeled reviews on five datasets.
doi:10.1371/journal.pone.0107122.g004

[9]. The results of ADN method are reported by [10]. AHD is the proposed active learning method in this paper. Through Table 5, we can see that the results of AHD is better than Active and MECH methods, and competitive with ADN method. Because ADN and AHD methods are both deep learning method, these results prove that deep architecture is good for sentiment classification.

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. We use HDBN method in this part, considering AHD method choose the reviews need to label from an unlabeled pool, it is unfair to compare the performance of AHD when the size of unlabeled pool is different.

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 can be seen in Table 3. 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 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.

Performance with variance of labeled data

To verify the contribution of labeled reviews for our proposed method, we did several experiments with different number of labeled reviews on five datasets. To compare the active learning performance with ADN [10], we use AHD method in this experiment, all the experimental setting are same as ADN. The architectures for AHD used in this experiment can be seen in Table 3.

The test accuracies of ADN and AHD with different number of labeled reviews on five datasets are shown in Fig. 4. We can see that the performance of AHD is better than ADN for most of the experimental setting, although they are both based on the DBN method. This proves that the convolutional computation has better performance than the normal computation in the deep architecture for sentiment classification. We can also see that both ADN and AHD can get high accuracy even with just 20 labeled reviews for training. This proves the effect of deep learning method for semi-supervised learning with very few labeled reviews.

Conclusions

In this paper, we propose a novel semi-supervised learning method, AHD, to address the sentiment classification problem with a small number of labeled reviews. AHD seamlessly incorporate convolutional computation into the DBN architecture, and use CRBM to abstract the review information effectively. One promising property of AHD 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, AHD can greatly reduce the dimension of reviews through RBM and abstract the information of reviews through the cooperate of RBM and CRBM. Then an exponential loss function is used to refine the constructed deep architecture with few label information. Moreover, it can choose the review to be labeled actively, improve the performance of deep architecture effectively.

Experiments conducted on five sentiment datasets demonstrate that AHD outperforms most of previous methods and is

competitive with DBN based method, which demonstrates the performance of deep architecture for sentiment classification. Experiments are also conducted to verify the effectiveness of AHD method with different number of labeled reviews, the results show that AHD can reach very competitive performance with few labeled reviews and large amount of unlabeled reviews. It provides soundness support for the effectiveness of AHD for real applications, where collecting enough unlabeled data is a relatively easy task while it is hard to get enough labeled data.

Author Contributions

Conceived and designed the experiments: SZ QC XW. Performed the experiments: SZ. Analyzed the data: SZ. Contributed reagents/materials/analysis tools: SZ QC. Contributed to the writing of the manuscript: SZ.

References

1. Liu Y, Yu X, Huang X, An A (2010) S-plasa+: Adaptive sentiment analysis with application to sales performance prediction. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: ACM, pp. 873–874.
2. Wei W, Gulla JA (2010) Sentiment learning on product reviews via sentiment ontology tree. In: Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics. pp. 404–413.
3. Li S, Huang CR, Zhou G, Lee SYM (2010) Employing personal/impersonal views in supervised and semi-supervised sentiment classification. In: Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden: Association for Computational Linguistics, pp. 414–423.
4. Zhen Y, Yeung DY (2010) Sed: Supervised experimental design and its application to text classification. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. Geneva, Switzerland: ACM, pp. 299–306.
5. Chapelle O, Scholkopf B, Zien A (2006) Semi-Supervised Learning. Cambridge, MA, USA: MIT Press.
6. Pang B, Lee L (2008) Opinion mining and sentiment analysis, volume 2 of *Foundations and Trends in Information Retrieval*.
7. Zhu X (2007) Semi-supervised learning literature survey. Technical report, University of Wisconsin Madison, Madison, WI, USA.
8. Sindhwani V, Melville P (2008) Document-word co-regularization for semi-supervised sentiment analysis. In: International Conference on Data Mining. Pisa, Italy: IEEE, pp. 1025–1030.
9. Dasgupta S, Ng V (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. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 701–709.
10. Zhou S, Chen Q, Wang X (2010) Active deep networks for semi-supervised sentiment classification. In: International Conference on Computational Linguistics. pp. 1515–1523.
11. Salakhutdinov R, Hinton GE (2007) Learning a nonlinear embedding by preserving class neighbourhood structure. Journal of Machine Learning Research 2: 412–419.
12. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Computation 18: 1527–1554.
13. Ranzato M, Szummer M (2008) Semi-supervised learning of compact document representations with deep networks. In: International Conference on Machine Learning. Helsinki, Finland: ACM, pp. 792–799.
14. Socher R, Pennington J, Huang EH, Ng AY, Manning CD (2011) Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Edinburgh, Scotland, UK.: Association for Computational Linguistics, pp. 151–161.
15. Socher R, Perelygin A, Wu J, Chuang J, Manning CD, et al. (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1631–1642.
16. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE 86: 2278–2324.
17. Desjardins G, Bengio Y (2008) Empirical evaluation of convolutional rbms for vision. Technical report.
18. Lee H, Grosse R, Ranganath R, Ng A (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: International Conference on Machine Learning. Montreal, Canada: ACM, pp. 609–616.
19. Lee H, Largman Y, Pham P, Ng A (2009) Unsupervised feature learning for audio classification using convolutional deep belief networks. In: Advances in Neural Information Processing Systems. Vancouver, B.C., Canada: NIPS Foundation, pp. 1096–1103.
20. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? sentiment classification using machine learning techniques. In: Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 79–86.
21. Blitzer J, Dredze M, Pereira F (2007) Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Annual Meeting of the Association of Computational Linguistics. Prague, Czech Republic: Association for Computational Linguistics, pp. 440–447.
22. Kamvar S, Klein D, Manning C (2003) Spectral learning. In: International Joint Conferences on Artificial Intelligence. Catalonia, Spain: AAAI Press, pp. 561–566.
23. Collobert R, Sinz F, Weston J, Bottou L (2006) Large scale transductive svms. Journal of Machine Learning Research 7: 1687–1712.
24. Tong S, Koller D (2002) Support vector machine active learning with applications to text classification. Journal of Machine Learning Research 2: 45–66.