Neurocomputing 120 (2013) 536–546

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Active deep learning method for semi-supervised sentiment classification

Shusen Zhou [a,*], Qingcai Chen [b], Xiaolong Wang [b]

[a] School of Information and Electrical Engineering, Ludong University, Yantai, PR China
[b] Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, PR China

ARTICLE INFO

Article history:
Received 9 May 2012
Received in revised form
9 April 2013
Accepted 15 April 2013
Communicated by M. Wang
Available online 23 May 2013

Keywords:
Neural networks
Deep learning
Active learning
Sentiment classification

ABSTRACT

In natural language processing community, sentiment classification based on insufficient labeled data is a well-known challenging problem. In this paper, a novel semi-supervised learning algorithm called active deep network (ADN) is proposed to address this problem. First, we propose the semi-supervised learning framework of ADN. ADN is constructed by restricted Boltzmann machines (RBM) with unsupervised learning based on labeled reviews and abundant of unlabeled reviews. Then the constructed structure is fine-tuned by gradient-descent based supervised learning with an exponential loss function. Second, in the semi-supervised learning framework, we apply active learning to identify reviews that should be labeled as training data, then using the selected labeled reviews and all unlabeled reviews to train ADN architecture. Moreover, we combine the information density with ADN, and propose information ADN (IADN) method, which can apply the information density of all unlabeled reviews in choosing the manual labeled reviews. Experiments on five sentiment classification datasets show that ADN and IADN outperform classical semi-supervised learning algorithms, and deep learning techniques applied for sentiment classification.

1. Introduction

In recent years, sentiment analysis has received considerable attention in natural language processing (NLP) community [1–3]. Sentiment classification is a special type of text categorization, where the criterion of classification is the attitude expressed in the text, such as '*positive*' or '*negative*', '*thumbs up*' or '*thumbs down*', '*favorable*' or '*unfavorable*', etc. [4], rather than the subject or topic. Labeling the reviews with their sentiment would provide succinct summaries to readers, which makes it possible to focus the text mining on areas in need of improvement or on areas of success [5] and is helpful in business intelligence applications, recommender systems, and message filtering tasks [1].

While topics are often identifiable by keywords alone, sentiment classification appears to be a more challenge task [1]. First, sentiment is often conveyed with subtle linguistic mechanisms such as the use of sarcasm and highly domain-specific contextual cues [6]. For example, although the sentence "the thief tries to protect his excellent reputation" contains the word "excellent", it tells us nothing about the author's opinion and in fact could be well embedded in a negative review. Second, sentiment classification systems are typically domain-specific, which makes the expensive process of annotating a large amount of data for each domain and is a bottleneck in building high-quality systems [3]. This motivates the task of learning robust sentiment models from minimal supervision [6].

Recently, semi-supervised learning, which uses a large amount of unlabeled data together with labeled data to build better learners [7,8], has attracted more and more attention in sentiment classification [3,6]. Wang et al. propose a novel semi-supervised learning algorithm named semi-supervised kernel density estimation, which is developed based on kernel density estimation approach, both labeled and unlabeled data are leveraged to estimate class conditional probability densities based on an extended form of kernel density estimation [9]. Wang et al. propose a method named optimized multigraph-based semi-supervised learning which aims to simultaneously tackle insufficiency of training data and the curse of dimensionality problems in a unified scheme [10]. Zha et al. propose a novel graph-based learning framework in the setting of semi-supervised learning with multiple labels [11]. As argued by several research works [12,13], deep architecture that is composed of multiple levels of non-linear operations [14], is expected to perform well in semi-supervised learning, because of its capability of modeling hard artificial intelligent tasks. Deep belief network (DBN) is a representative deep learning algorithm that has achieved notable success for semi-supervised learning in NLP community [14]. Ranzato and Szummer [15] introduce an algorithm to learn text

* Corresponding author. Tel.: +86 13583573988.
E-mail addresses: zhoushusen@hotmail.com (S. Zhou), qingcai.chen@hitsz.edu.cn (Q. Chen), wangxl@insun.hit.edu.cn (X. Wang).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.04.017

document representations, which is based on semi-supervised auto-encoders that are combined to form a deep network.

Active learning is another way that can minimize the number of required labeled data while getting a competitive result. Rather than choosing the training set randomly, active learning chooses the training data actively, which reduces the needs of labeled data [16]. It has been widely explored in multimedia research community for its capability of reducing human annotation effort [17]. Zha et al. propose a novel active learning approach based on the optimum experimental design criteria in statistics for interactive video indexing [18]. Active learning is well-suited to many NLP problems, where unlabeled data may be abundant but annotation is slow and expensive [19]. Druck et al. propose an active learning approach in which the machine solicits labels on features rather than instances [20]. Zhu et al. combine active and semi-supervised learning under a Gaussian random field model; the active learning scheme requires a much smaller number of queries to achieve high accuracy compared with random query selection [21]. Recently, active learning has been applied in sentiment classification [3].

Inspired by the study of semi-supervised learning, active learning and deep learning, this paper proposes a semi-supervised sentiment classification method called active deep network (ADN). It is based on a representative deep learning method DBN [14] and active learning method [16]. First, we introduce the semi-supervised learning procedure of ADN method, which constructs the deep architecture with all unlabeled and labeled reviews, and fine tunes the deep architecture with few labeled reviews. To maximize the separability of the classifier, an exponential loss function is suggested. Second, we introduce the active learning procedure of ADN method. It first identifies a small number of unlabeled reviews for manual labeling by an active learner, and then trains the deep architecture with the labeled reviews and all other unlabeled reviews. Moreover, we propose information ADN (IADN) method, to combine the information density with ADN, which puts the information density of all unlabeled reviews into consideration while choosing the unlabeled reviews for further labeling.

The main contributions of this paper include: First, this paper introduces a new deep architecture that integrates the abstraction ability of deep belief networks and the classification ability of backpropagation strategy. It improves the generalization capability by using the abundant number of unlabeled reviews, and directly optimizes the classification results in training dataset via the back propagation strategy, which makes it possible to achieve attractive classification performance with few labeled reviews. Second, this paper proposes two effective active learning methods that integrate the review selection ability of active learning and classification ability of deep architecture. Both the labeled review selector and classifier are based on the same architecture, which provides a unified framework for the semi-supervised classification task. Third, this paper applies semi-supervised learning and active learning to sentiment classification successfully and gets competitive performance. Our experimental results on five sentiment classification datasets show that both ADN and IADN outperform previous sentiment classification methods and deep learning methods.

This paper is an expanded version of Zhou et al. [22]. Many new contents are incorporated here: First, the related works of sentiment classification have been extended; more detail introduction about sentiment classification methods has been made. Second, an active learning method called IADN is proposed, which combines information density with ADN and achieves competitive performance for sentiment classification. Third, more experiments have been conducted to evaluate the performance of deep architecture, information density incorporation, and various loss functions. Moreover, we evaluate the proposed active learning methods with a different number of labeled and unlabeled reviews.

The rest of the paper is organized as follows. Section 2 gives an overview of sentiment classification. The proposed semi-supervised learning method ADN is described in Section 3. Section 4 combines ADN and information density into IADN method. Section 5 evaluates ADN and IADN by comparing their classification performance with existing sentiment classification methods and deep learning methods on sentiment datasets. The paper is closed with a conclusion.

2. Sentiment classification

Sentiment classification can be performed on words, sentences or documents, and is generally categorized into lexicon-based [23] and corpus-based classification methods [24]. The detail survey about techniques and approaches of sentiment classification can be seen in the book [25]. In this paper, we focus on corpus-based classification methods.

Corpus-based methods use a labeled corpus to train a sentiment classifier [24]. Pang et al. [1] are the first who apply machine learning approach to corpus-based sentiment classification. They found that standard machine learning techniques outperform human-produced baselines. They also carried out important experiments on selecting the best features and concluded that unigrams performed better than bigrams or unigrams and bigrams together. Dave et al. [26] draw attention on information retrieval techniques for feature extraction and scoring in the sentiment classification task. Pang and Lee [27] apply text-categorization techniques to the subjective portions of the sentiment documents. These portions are extracted by efficient techniques for finding minimum cuts in graphs. Gamon [5] demonstrates that high accuracy can be achieved by using large feature vectors in combination with feature reduction in the very noisy domain of customer feedback data. Mullen and Collier [28] use support vector machines to bring together diverse sources of potentially pertinent information for sentiment classification, including several favorability measures for phrases and adjectives and, where available, knowledge of the topic of the text. Ng et al. [29] demonstrate that sentiment classification can be performed with high accuracy using only unigrams as features. Mcdonald et al. [30] investigate a structured model for jointly classifying the sentiment of text at various levels of granularity, which is based on standard sequence classification techniques using constrained Viterbi to ensure consistent solutions. Xia et al. [31] introduce the sentiment vector space model to represent song lyric document, assign the sentiment labels such as light-hearted and heavy-hearted. Li et al. [32] propose a machine learning approach to incorporate polarity shifting information into a document-level sentiment classification system. Liu et al. [33] present an adaptive sentiment analysis model that aims to capture the hidden sentiment factors in reviews through the capability of being incrementally updated as more data becoming available. Wei et al. [34] propose a novel approach to label attributes of a product and their associated sentiments in product reviews by a hierarchical learning process with a defined sentiment ontology tree.

Supervised sentiment classification systems are domain-specific and annotating a large-scale corpus for each domain is very expensive [3]. There exists several solutions for this issue.

The first solution is cross-domain sentiment classification. Aue and Gamon [35] survey four different approaches to customize a sentiment classification system for a new target domain in the absence of large amounts of labeled data. Blitzer et al. [2] investigate domain adaptation for sentiment classifiers, which reducing the relative error due to adaptation between domains by an average of 46% over a supervised baseline, and identify a measure of domain similarity that correlates well with the potential for adaptation of a classifier from one domain to another. Tan et al. [36] combine old-domain labeled examples with new-domain unlabeled ones, and retrain the base classifier over all

these examples. Li and Zong [37] study multi-domain sentiment classification which aims to improve performance through fusing training data from multiple domains. Pan et al. [38] propose a cross-domain sentiment classification method that aligns domain-specific words extracted from different domains into unified clusters, with the help of domain-independent words as a bridge. Bollegala et al. [39] automatically create a sentiment sensitive thesaurus using both labeled and unlabeled data from multiple source domains to find the association between words that express similar sentiments in different domains. He et al. [40] modify the joint sentiment-topic model by incorporating word polarity priors through modifying the topic-word Dirichlet priors, study the polarity-bearing topics extracted by joint sentiment-topic model and show that by augmenting the original feature space with polarity-bearing topics, achieve the state-of-the-art performance of 95% on the movie review data and an average of 90% on the multi-domain sentiment dataset.

The second solution is semi-supervised sentiment classification. Goldberg and Zhu [41] present a graph-based semi-supervised learning algorithm to address the sentiment analysis task of rating inference, inferring numerical ratings based on the perceived sentiment. Sindhwani and Melville [42] propose a semi-supervised sentiment classification algorithm that utilizes lexical prior knowledge in conjunction with unlabeled data. Dasgupta and Ng [3] first mine the unambiguous reviews using spectral techniques, and then exploit them to classify the ambiguous reviews via a novel combination of active learning, transductive learning, and ensemble learning. Li et al. [4] adopt two views, personal and impersonal views, and employ them in both supervised and semi-supervised sentiment classification.

The third solution is unsupervised sentiment classification. Zagibalov and Carroll [43] describe an automatic seed word selection method for unsupervised sentiment classification of product reviews in Chinese.

There are also several other methods to solve this issue. Read [44] demonstrates that training data automatically labeled with encountered there emoticons has the potential of being independent of domain, topic and time. Wan [24] studies on cross-lingual sentiment classification, which leverages an available English corpus for Chinese sentiment classification by using the English corpus as training data. Machine translation services are used for eliminating the language gap between the training set and test set, English features and Chinese features are considered as two independent views of the classification problem. Lu et al. [45] present a novel approach for joint bilingual sentiment classification at the sentence level that augments available labeled data in each language with unlabeled parallel data.

However, unsupervised learning of sentiment is difficult, partially because of the prevalence of sentimentally ambiguous reviews [3]. Using multi-domain sentiment corpus to sentiment classification is also hard to apply. It is because that each domain has a very limited amount of training data, due to annotating a large corpus is difficult and time-consuming [37]. Cross-domain, semi-supervised learning and unsupervised learning methods are used based on the background that training data is not enough. When there are not enough training data for each domain, we can use cross-domain methods. When there are not enough labeled data, we can use semi-supervised learning methods. When there is no labeled data, we can use unsupervised learning methods. In this paper, we just focus on semi-supervised sentiment classification methods.

3. Active deep networks

In this part, we propose a semi-supervised learning algorithm, active deep network (ADN), to address the sentiment classification problem with active learning. Section 3.1 formulates the ADN problem. Section 3.2 proposes the semi-supervised learning method of ADN. Section 3.3 proposes active learning method of ADN. Section 3.4 gives the ADN procedure.

3.1. Problem formulation

The dataset is compound by a substantial amount of product reviews. We preprocess the reviews to be classified, the experimental setting is same with [3]. Each review is represented as a vector of unigrams, using binary weight equal to 1 for terms present in a vector. Moreover, the punctuation, numbers, and words of length one are removed from the vector. Finally, we sort the vocabulary by document frequency and remove the top 1.5%. It is because that many of these high document frequency words are stopwords or domain specific general-purpose words (e.g., "book" in the book domain), these noise words would not be helpful for sentiment classification. These words typically comprise 1–2% of a vocabulary, the decision of exactly how many terms to remove is subjective: a large corpus typically requires more removals than a small corpus. To be consistent, we simply remove the top 1.5% high frequency words.

After preprocessing, each review is represented by a vector. Then the dataset is represented as a matrix

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

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. Each column of $\mathbf{X}$ corresponds to a sample $\mathbf{x}$ of review. A sample that has all features is viewed as a vector in $\mathbb{R}^D$, where the jth coordinate corresponds to the jth feature.

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

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

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

Let $\mathbf{Y}$ be a set of labels correspond to L labeled training reviews and is denoted as

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

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

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

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 $\mathbf{X}^L\to\mathbf{Y}^L$ using the L labeled reviews and all unlabeled reviews in order to determine $\mathbf{y}$ when a new review $\mathbf{x}$ comes.

3.2. Semi-supervised learning

To address the problem formulated in Section 3.1, we propose a deep architecture for ADN method, as shown in Fig. 1. The deep architecture is a fully interconnected directed belief nets with one input layer $\mathbf{h}^0$, N hidden layers $\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^N$, and one labeled layer

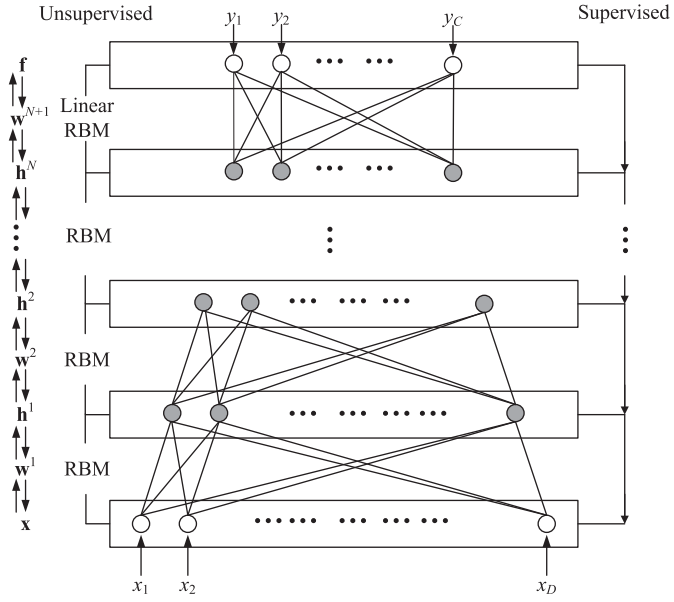

Fig. 1. Architecture of active deep networks.

at the top. The input layer $\mathbf{h}^0$ has D units, equal to the number of features of sample review $\mathbf{x}$. The label layer has C units, equal to the number of classes of label vector $\mathbf{y}$. The numbers of units for hidden layers, currently, are pre-defined according to the experience or intuition. The seeking of the mapping function, 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 semi-supervised learning method based on ADN architecture can be divided into two stages: First, ADN architecture is constructed by greedy layer-wise unsupervised learning using RBMs as building blocks. All the unlabeled reviews together with L labeled reviews are utilized to find the parameter space $\mathbf{W}$ with N layers. Second, ADN architecture is trained according to the exponential loss function using gradient descent method. The parameter space $\mathbf{W}$ is retrained by an exponential loss function using L labeled data. As it is difficult to optimize a deep architecture using supervised learning directly, the unsupervised learning stage can abstract the reviews effectively, and prevent overfitting of the supervised training.

For unsupervised learning, we define the energy of the joint configuration $(\mathbf{h}^{k-1},\mathbf{h}^k)$ as [14]

$$E(\mathbf{h}^{k-1},\mathbf{h}^k;\theta)=-\sum_{s=1}^{D_{k-1}}\sum_{t=1}^{D_k}w_{st}^k h_s^{k-1}h_t^k-\sum_{s=1}^{D_{k-1}}b_s^{k-1}h_s^{k-1}-\sum_{t=1}^{D_k}c_t^k h_t^k \tag{5}$$

where $\theta=(\mathbf{w},\mathbf{b},\mathbf{c})$ are the model parameters: w_{st}^k is the symmetric interaction term between unit s in the layer $\mathbf{h}^{k-1}$ and unit t in the layer $\mathbf{h}^k$, $k=1,\ldots,N-1$. b_s^{k-1} is the sth bias of layer $\mathbf{h}^{k-1}$ and c_t^k is the tth bias of layer $\mathbf{h}^k$. D_k is the number of units in the kth layer. The network assigns a probability to every possible data via this energy function. The probability of a training data can be raised by adjusting the weights and biases to lower the energy of that data and to raise the energy of similar, confabulated data that $\mathbf{h}^k$ would prefer to the real data. When we input the value of $\mathbf{h}^k$, the network can learn the content of $\mathbf{h}^{k-1}$ by minimizing this energy function.

The probability that the model assigns to a $\mathbf{h}^{k-1}$ is

$$P(\mathbf{h}^{k-1};\theta)=\frac{1}{Z(\theta)}\sum_{\mathbf{h}^k}\exp(-E(\mathbf{h}^{k-1},\ \mathbf{h}^k;\theta)) \tag{6}$$

$$Z(\theta)=\sum_{\mathbf{h}^{k-1}}\sum_{\mathbf{h}^k}\exp(-E(\mathbf{h}^{k-1},\ \mathbf{h}^k;\ \theta)) \tag{7}$$

where $Z(\theta)$ denotes the normalizing constant.

The conditional distributions over $\mathbf{h}^k$ and $\mathbf{h}^{k-1}$ are given as

$$p(\mathbf{h}^k|\mathbf{h}^{k-1})=\prod_t p(h_t^k|\mathbf{h}^{k-1}) \tag{8}$$

$$p(\mathbf{h}^{k-1}|\mathbf{h}^k)=\prod_s p(h_s^{k-1}|\mathbf{h}^k) \tag{9}$$

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

$$p(h_t^k=1|\mathbf{h}^{k-1})=\text{sigm}\left(c_t^k+\sum_s w_{st}^k h_s^{k-1}\right) \tag{10}$$

the probability of turning unit s is a logistic function of the states of $\mathbf{h}^k$ and w_{st}^k

$$p(h_s^{k-1}=1|\mathbf{h}^k)=\text{sigm}\left(b_s^{k-1}+\sum_t w_{st}^k h_t^k\right) \tag{11}$$

where the logistic function is

$$\text{sigm}\,(\eta)=1/(1+e^{-\eta}) \tag{12}$$

The derivative of the log-likelihood with respect to the model parameter $\mathbf{w}^k$ can be obtained from Eq. (6)

$$\frac{\partial\log p(\mathbf{h}^{k-1})}{\partial\, w_{st}^k}=\langle h_s^{k-1}h_t^k\rangle_{P_0}-\langle h_s^{k-1}h_t^k\rangle_{P_{Model}} \tag{13}$$

where $\langle\cdot\rangle_{P_0}$ denotes an expectation with respect to the data distribution and $\langle\cdot\rangle_{P_{Model}}$ denotes an expectation with respect to the distribution defined by the model [46].

The expectation $\langle\cdot\rangle_{P_{Model}}$ cannot be computed analytically. In practice, $\langle\cdot\rangle_{P_{Model}}$ is replaced by $\langle\cdot\rangle_{P_1}$, which denotes a distribution of samples when the feature detectors are being driven by reconstructed $\mathbf{h}^{k-1}$. This is an approximation to the gradient of a different objective function, called the contrastive divergence (CD) [47]

$$\Delta w_{st}^k=\eta(\langle h_s^{k-1}h_t^k\rangle_{P_0}-\langle h_s^{k-1}h_t^k\rangle_{P_1}) \tag{14}$$

where η is the learning rate.

Then the parameter $\mathbf{w}^k$ can be adjusted through

$$w_{st}^k=\vartheta w_{st}^k+\Delta w_{st}^k \tag{15}$$

where ϑ is the momentum.

The above discussion is based on the training of the parameters between two hidden layers with one sample review x. For unsupervised learning, we construct the deep architecture using all labeled reviews with unlabeled reviews by inputting them one by one from layer $\mathbf{h}^0$, train the parameters between $\mathbf{h}^0$ and $\mathbf{h}^1$. Then $\mathbf{h}^1$ is constructed, the value of $\mathbf{h}^1$ is calculated by $\mathbf{h}^0$ and the trained parameters between $\mathbf{h}^0$ and $\mathbf{h}^1$. We can use it to construct the next layer $\mathbf{h}^2$. The deep architecture is constructed layer by layer from bottom to top. In 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 above, the layer $\mathbf{h}^k$ is obtained as below for a sample $\mathbf{x}$ fed from layer $\mathbf{h}^0$

$$h_t^k(\mathbf{x})=\text{sigm}\left(c_t^k+\sum_{s=1}^{D_{k-1}}w_{st}^k h_s^{k-1}(\mathbf{x})\right)\quad t=1,\ldots,D_k;\ k=1,\ldots,N-1 \tag{16}$$

The parameter space $\mathbf{w}^N$ is initialized randomly, just as back-propagation algorithm. Then ADN architecture is constructed. The top hidden layer is formulated as

$$h_t^N(\mathbf{x})=c_t^N+\sum_{s=1}^{D_{N-1}}w_{st}^N h_s^{N-1}(\mathbf{x})\quad t=1,\ldots,D_N \tag{17}$$

For supervised learning, the ADN architecture is trained by L labeled data. The optimization problem is formulized as

$$\underset{h^N}{\text{argmin}}\quad f(h^N(\mathbf{X}^L),\mathbf{Y}^L) \tag{18}$$

where

$$f(h^N(\mathbf{X}^L), \mathbf{Y}^L) = \sum_{i=1}^{L} \sum_{j=1}^{C} T(h_j^N(\mathbf{x}^i) y_j^i) \tag{19}$$

and the loss function is defined as

$$T(r) = \exp(-r) \tag{20}$$

In the supervised learning stage, the stochastic activities are replaced by deterministic, real valued probabilities. The greedy layer-wise unsupervised learning is just used to initialize the parameter of deep architecture, the parameters of the deep architecture are updated based on Eq. (15). After initialization, real values are used in all the nodes of the deep architecture. We use gradient-descent through the whole deep architecture to retrain the weights for optimal classification.

3.3. Active learning

Semi-supervised learning allows us to classify reviews with few labeled data. However, annotating the reviews manually is expensive, so we expect to get higher performance with fewer labeled data. Active learning can help to choose those reviews that should be labeled manually in order to achieving higher classification performance with the same number of labeled data. For such purpose, we incorporate pool-based active learning with the ADN method, which accesses to a pool of unlabeled instances and requests the labels for some number of them [16].

Given an unlabeled pool $\mathbf{X}^R$ and an initial labeled dataset $\mathbf{X}^L$ (one positive, one negative), the ADN architecture $\mathbf{h}^N$ will decide which instance in $\mathbf{X}^R$ to query next. Then the parameters of $\mathbf{h}^N$ are adjusted after new reviews are labeled and inserted into the labeled dataset. The main issue for an active learner is the choosing of next unlabeled instance to query. In this paper, we choose the reviews of which the labels are most uncertain for the classifier. Drawing on previous work on active learning [3,16], we define the uncertainty of a review as the reciprocal of its distance from the separating hyperplane. In other words, reviews that are near the separating hyperplane are more uncertain than the reviews that are farther away.

The deep architecture of ADN is trained by all unlabeled data and initial labeled training set with DBN based semi-supervised learning first, which has been introduced in Section 3.2. After semi-supervised learning, the parameters of ADN are adjusted. Given an unlabeled pool $\mathbf{X}^R$, the next unlabeled instance to be queried is chosen according to the location of $\mathbf{h}^N(\mathbf{X}^R)$. The distance between a point $\mathbf{h}^N(\mathbf{x}^i)$ and the classes separation line $h_1^N = h_2^N$ is

$$\mathbf{d}^i = |h_1^N(\mathbf{x}^i) - h_2^N(\mathbf{x}^i)| / \sqrt{2} \tag{21}$$

The selected training review to be labeled manually is given by

$$s = \{j : \mathbf{d}^j = \min(\mathbf{d})\} \tag{22}$$

We can select a group of the most uncertain reviews to label at each time.

The experimental setting is similar with Dasgupta and Ng [3]. We perform active learning for five iterations and select 20 of the most uncertainty reviews to be queried each time. Then the ADN is retrained on all labeled and unlabeled reviews so far with semi-supervised learning. At last, the label of a review $\mathbf{x}$ is determined according to the output $\mathbf{h}^N(\mathbf{x})$ of the ADN architecture as below

$$y_j = \begin{cases} 1 & \text{if } h_j^N(\mathbf{x}) = \max(\mathbf{h}^N(\mathbf{x})) \\ -1 & \text{if } h_j^N(\mathbf{x}) \neq \max(\mathbf{h}^N(\mathbf{x})) \end{cases} \tag{23}$$

As shown by Tong and Koller [16], the balance random method, which randomly sample an equal number of positive and negative instances from the pool, has much better performance than the regular random method. So we incorporate this “Balance” idea with ADN method. However, it is not possible to choose an equal number of positive and negative instances without labeling the entire pool of instances in advance. So we present a simple way to approximate the balance of positive and negative reviews. For each iteration, we count, first, the number of positive and negative labeled reviews respectively. Second, we classify the unlabeled reviews in the pool with the deep architecture trained by the previous iteration. Third, we choose the appropriate number of positive and negative reviews labeled in the second step and add them into the labeled dataset, to let the number of labeled and unlabeled review equally. Fourth, we relabel all these new added reviews manually to ensure the correctness of all the review's label in the labeled dataset.

3.4. ADN procedure

The procedure of ADN is shown in Algorithm 1. For the training of ADN architecture, the parameters are random initialized with normal distribution. All the training data and test data are used to train the ADN with unsupervised learning, which can be seen as transductive learning [48]. The training set $\mathbf{X}^R$ can be seen as an unlabeled pool. We randomly select one positive and one negative review in the pool to input as the initial labeled training set that are used for supervised learning. The number of units in hidden layer $D_1, \ldots, D_N$ and the number of epochs Q are set manually based on the dimension of the input data and the size of training dataset. The iteration times I and the number of active choosing reviews for each iteration G can be set based on the number of labeled reviews in the experiment.

For each iteration, the ADN architecture is re-trained by all the unlabeled and labeled reviews with unsupervised learning and supervised learning first, the parameters of deep architecture are initialized with the training results of previous iteration. Then we choose G reviews from the unlabeled pool based on the distance of these data from the separating line, label these reviews manually, and add them into the labeled dataset. For the next iteration, the unsupervised learning is initialized with the parameters trained in the supervised stage of previous iteration, then the supervised learning is applied based on the new labeled dataset again. The unsupervised and supervised learning stages in turn can adjust the parameters with each other, and improve the abstraction and classification ability of the deep architecture. At last, ADN architecture is retrained by all the unlabeled reviews and existing labeled reviews. After training, the ADN architecture is tested based on Eq. (23).

Since the proposed ADN method can active choose the labeled dataset and classify the reviews with the same architecture, which avoids the barrier between choosing and training procedures with different architectures. More importantly, the parameters of ADN are trained iteratively on the labeled data selection process, which further improves the performance of ADN.

Algorithm 1. Active deep networks procedure.

Input: data $\mathbf{X}$, $(\mathbf{X}^L, \mathbf{Y}^L)$ (one positive and one negative reviews);
number of layers N; number of epochs Q;
number of training data R; number of test data T;
normal distribution based random initialize parameter space $\mathbf{W}$;
number of iterations I; number of active choosing reviews for each iteration G;
Output: deep architecture with parameter space $\mathbf{W}$
for $i = 1; i \leq I$ **do**
  Step 1. Greedy layer-wise unsupervised learning
  for $n = 1; n \leq N-1$ **do**

for $q=1; q \le Q$ **do**
  for $k=1; k \le R+T$ **do**
    Calculate the non-linear positive and negative:
    $p(h_t^k=1|\mathbf{h}^{k-1}) = \text{sigm}\,(c_t^k + \sum_s w_{st}^k h_s^{k-1})$
    $p(h_s^{k-1}=1|\mathbf{h}^k) = \text{sigm}\,(b_s^{k-1} + \sum_t w_{st}^k h_t^k)$
    Update the weights and biases:
    $\Delta w_{st}^k = \eta(\langle h_s^{k-1} h_t^k \rangle_{P_0} - \langle h_s^{k-1} h_t^k \rangle_{P_1})$
  end
 end
end
Step 2. Supervised learning with gradient descent
Minimize $f(h^N(\mathbf{X}), \mathbf{Y})$ on labeled dataset $\mathbf{X}^L$, update the parameter
space $\mathbf{W}$ according to: $\text{argmin}_{h^N} \quad f(h^N(\mathbf{X}^L), \mathbf{Y}^L)$
Step 3. Choose instances for labeled dataset
Choose G instances which near the separating line by:
$s = \{j : \mathbf{d}^j = \min(\mathbf{d})\}$
Add G instances into the labeled dataset $\mathbf{X}^L$
end
Train ADN with Steps 1 and 2.

4. Information ADN

In this part, we combine information density idea [19] with ADN, propose a novel information ADN (IADN) method for semi-supervised sentiment classification.

The proposed ADN method can actively choose the reviews that are near the separating hyperplane as the training data to be labeled manually. However, ADN does not consider the information density of these review candidates. For example, in Fig. 2, the samples $\vec{A}$ and $\vec{B}$ are two labeled examples, the other circles are unlabeled data. Since $\vec{C}$ is the nearest sample to decision boundary, it should be chosen by ADN method. However, $\vec{C}$ is far from the center of two classes, i.e., it is not a representative sample in the distribution. In this case, querying $\vec{D}$ is likely to contain more information about the dataset. The IADN method is proposed to put this observation into consideration.

When the deep architecture is trained by L labeled data and all unlabeled data, the parameters are adapted, $\mathbf{h}^N(\mathbf{x})$ is used to represent the sample $\mathbf{x}$. Given an unlabeled pool $\mathbf{X}^R$, the next unlabeled instance to be queried is chosen according to the location of $\mathbf{h}^N(\mathbf{X}^R)$. The informativeness of $\mathbf{h}^N(\mathbf{x})$ is weighted by its average similarity to other samples which in the same side of the separation line with $\mathbf{h}^N(\mathbf{x})$. It is formalized as

$$\mathbf{ID}^i = \mathbf{d}^i \times \left(\frac{1}{U-1} \sum_{j=1, j \ne i}^{U} dis(\mathbf{h}^N(\mathbf{x}^i), \mathbf{h}^N(\mathbf{x}^j)) \right)^{\beta} \tag{24}$$

where

$$\mathbf{X}^U = \{j : \mathbf{x}^j \in \mathbf{X}^R \cap (h_1^N(\mathbf{x}) - h_2^N(\mathbf{x})) \times (h_1^N(\mathbf{x}^j) - h_2^N(\mathbf{x}^j)) > 0\} \tag{25}$$

indicates the unlabeled instances that belong to the same class of $\mathbf{x}$ based on the classification result of current trained classifier

$$dis(\mathbf{h}^N(\mathbf{x}^i), \mathbf{h}^N(\mathbf{x}^j)) = |h_1^N(\mathbf{x}^i) - h_1^N(\mathbf{x}^j)| + |h_2^N(\mathbf{x}^i) - h_2^N(\mathbf{x}^j)| \tag{26}$$

denotes the distance of $\mathbf{h}^N(\mathbf{x}^i)$ and $\mathbf{h}^N(\mathbf{x}^j)$. $\mathbf{d}^i$ denotes the distance between a point $\mathbf{h}^N(\mathbf{x}^i)$ and the separation line, and is defined by Eq. (21). β controls the relative importance of the density term.

The training reviews that should be labeled manually are given by

$$s = \{\mathbf{x}^i : \mathbf{ID}^i = \min(\mathbf{ID})\} \tag{27}$$

The balance selection procedure in ADN is not consider the cases when there are not enough positive (or negative) reviews for section, it will select the reviews randomly in this case. For IADN method, we select all remaining positive (or negative) reviews, and fill the gap with remaining negative (or positive) reviews in this case. The density calculation relies on the prediction of the labels by the classifier is necessary, although it might be mislead in. Because we do not know the label of the reviews in the pool. The classifier can recognize most of the reviews rightly. Even though some reviews which near the separation line are recognized wrongly, there are little effect for the density calculation. Because these wrong recognized reviews are close to these two classes of reviews at the same time.

The ADN and IADN architecture have different number of hidden units for each hidden layer. The width of deep architecture used in different datasets is setting based on the scale of the dataset and the dimension of the input data. If the scale of the dataset is increasing, and the dimension of the input data is increasing, the number of hidden units for each hidden layer is increasing too. Because more parameters in the large deep architecture need more high dimension training data to train the architecture effectively. Moreover, the number of units in the last hidden layer is more than other hidden layers, because the last hidden layer is linear, and the other hidden layers are non-linear, more units are needed to be used in linear layer to represent a model.

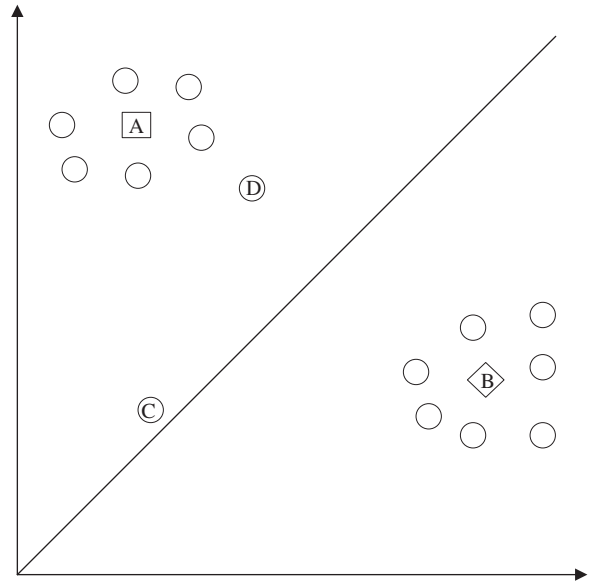

Fig. 2. Illustration of information density idea.

5. Experiments

We conduct several experiments to compare the performance of ADN and IADN with which of existing methods. We have the following questions in mind while designing and conducting the experiments:

1. How do ADN and IADN perform when compared with other state-of-the-art semi-supervised learning methods for sentiment classification?
2. How do ADN and IADN perform when compared with semi-supervised learning method based on our proposed deep architecture?
3. How does information density performs when there are few labeled data?
4. How does deep architecture performs for different loss functions?
5. How does varying the number of labeled reviews affect the performance of ADN and IADN?

6. How does varying the number of unlabeled reviews affect the performance of ADN and IADN?

These questions are answered in the following subsections: question 1 in Section 5.2, question 2 in Section 5.3, question 3 in Section 5.4, question 4 in Section 5.5, question 5 in Section 5.6, and question 6 in Section 5.7.

5.1. Experimental setup

The performance of proposed ADN and IADN methods is evaluated by five sentiment classification datasets. The first dataset is MOV [1], which is a widely-used movie review dataset. The other four datasets contain reviews of four types of products, which include books (BOO), DVDs (DVD), electronics (ELE), and kitchen appliances (KIT) respectively [2,3]. Each dataset contains 1000 positive and 1000 negative reviews.

For MOV dataset, the ADN and IADN structures used in this experiment are 100-100-200-2, which represents the number of units in output layer is 2, and in 3 hidden layers are 100, 100, and 200 respectively. For the other four datasets, the ADN and IADN structures used in this experiment are 50-50-200-2. The number of units in input layer is the same as the dimensions of each dataset. The number of units in the third hidden layer is more than previous two hidden layers, because the unit of third hidden layer is linear, more units can improve the representation ability of third hidden layer. As the size of vocabulary in MOV dataset is larger than in other four datasets, the number of units in previous two hidden layers for MOV dataset is more than for other four datasets. The architecture of ADN and IADN is similar with DBN, but with a different loss function introduced for supervised learning stage. The parameters of the deep architecture are fixed as the default parameter settings of Hinton DBN package [14]. For greedy layer-wise unsupervised learning, we train the weights of each layer independently with the 30 epochs and the learning rate is set to 0.1. The initial momentum is 0.5 and after five epochs, the momentum is set to 0.9. For supervised learning, we run 10 epochs, three times of linear searches are performed in each epoch.

We compare the classification performance of ADN and IADN with six representative classifiers, i.e., semi-supervised spectral learning (Spectral) [49], transductive SVM (TSVM), active learning (Active) [16], mine the easy classify the hard (MECH) [3], deep belief networks (DBN) [14], and recursive autoencoders (RAE) [23]. Spectral learning, TSVM, and active learning method are three baseline methods for sentiment classification. Spectral learning incorporates labeled data into the clustering framework in the form of must-link and cannot-link constraints. TSVM is the semi-supervised learning version of SVM. Active learning is implemented based on SVM, which training an inductive SVM on one labeled review from each class, iteratively labeling the most uncertain unlabeled reviews and re-training the SVM until 100 reviews are labeled. MECH is a new semi-supervised method for sentiment classification [3], which first mines the unambiguous reviews using spectral techniques, and then exploits them to classify the ambiguous reviews via a novel combination of active learning, transductive learning, and ensemble learning. The implementation details of Spectral, TSVM, Active, and MECH methods are introduced by Dasgupta and Ng [3]. DBN is a classical deep learning method proposed recently [14]. The parameters of the DBN in our experiment are fixed as the default parameter settings of Hinton DBN package [14], the minimum mean squared error loss function is used in DBN architecture. RAE learns vector space representations for multi-word phrases based on recursive autoencoders. The implementation details of RAE are introduced by Socher et al. [23].

5.2. ADN performance

To compare the performance of ADN and IADN with previous works, similar to Dasgupta and Ng [3], we randomly divide the 2000 reviews into 10 folds and test all the algorithms using cross-validation. All reviews are used as unlabeled data, where 1000 used for training and the other 1000 for test. In each fold, 100 reviews are random selected as training data and the remaining 100 reviews are used for test. For the randomness involved in the choice of labeled data, all the results of Spectral, TSVM, and DBN methods are acquired by repeating 10 times for each fold and then taking average over results. For Active, MECH, ADN, and IADN methods, one positive and one negative reviews are selected for the initialization of active leaning, 100 labeled reviews are chosen from the training dataset by active learning and are used for training the classifier. For Active, MECH, ADN, and IADN methods, the active learning is performed for five iterations. In each iteration, 20 of the most uncertain points are selected and labeled, and then the classifier is retrained on all of the unlabeled reviews and labeled reviews annotated so far. After five iterations, 100 labeled reviews are used for training. For these active learning methods, the initial two labeled reviews are selected randomly, so we repeat 30 times for each method, and the results are averaged. For Spectral, TSVM, DBN, and RAE methods, 100 labeled reviews are selected randomly. For Active, MECH, ADN, and IADN methods, 100 labeled reviews are selected by active learning, just the first two labeled reviews are selected randomly.

The classification accuracies on test data in cross validation for five datasets and eight methods are shown in Table 1. The results of previous four methods are reported by Dasgupta and Ng [3]. The structure and parameter used for DBN are the same as ADN and IADN in this experiment. The experiment of RAE is done based on the default parameters of the source code proposed by Socher et al. [23]. Through Table 1, we can see that the performance of DBN is competitive with MECH. Since MECH is the combination of spectral clustering, TSVM and active learning, DBN is just a classification method based on deep neural network, this result shows the good learning ability of deep architecture. ADN is a combination of semi-supervised learning and active learning based on deep architecture, the performance of ADN is better than previous six methods on all the five datasets. This could be contributed by: First, ADN uses a deep architecture to guide the output vector of samples belonged to different regions of new Euclidean space, which can abstract the useful information that is not accessible to other learners; Second, ADN uses an exponential loss function to maximize the separability of labeled reviews in global refinement for better discriminability; Third, ADN fully exploits the embedding information from the large amount of unlabeled reviews to improve the robustness of the classifier; Fourth, ADN chooses the useful training reviews actively, which also improves the classification performance. The performance of IADN is better than previous seven methods. It is because that the semi-supervised learning method used in IADN is same as in ADN,

Table 1
Test accuracy with 100 labeled reviews for five datasets and eight methods.

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
Active	68.9	68.1	63.3	58.6	58.0
MECH	76.2	74.1	70.6	62.1	62.7
DBN	71.3	72.6	73.6	64.3	66.7
RAE	66.3	69.4	68.2	61.3	63.1
ADN	76.3	77.5	76.8	69.0	71.6
IADN	**76.4**	**78.2**	**77.9**	**69.7**	**72.2**

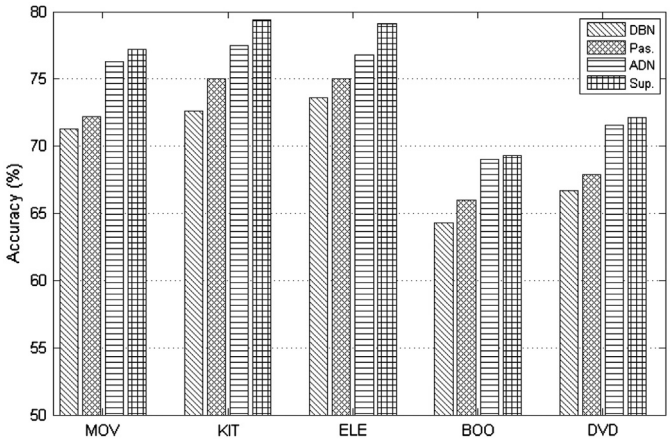

Fig. 3. Test accuracy of DBN and ADN with different experiment setting on five datasets.

and the active learning method used in IADN improves the classification performance.

5.3. Effect of active learning

To evaluate the contribution of the active learning in the proposed methods, we conduct following experiments. The architectures used in this section are the same as Section 5.2.

Passive learning: We are randomly select 100 reviews from the training fold and use them as labeled data. Then the semi-supervised learning method used in ADN is applied, to train and test the performance, which is called ADN with passive learning method (or Passive learning for short) and is denoted as "Pas." in Fig. 3. The experiment is run 10 times for each fold, and the average is taken over all results. The test accuracies of DBN, ADN with passive learning, and ADN on five datasets are shown in Fig. 3. Compared with DBN, the mean accuracy on five datasets for ADN with passive learning has been improved from 69.7% to 71.2%, which is contributed by the exponential loss function used in ADN architecture. Compared with ADN method, the mean accuracy on five datasets for ADN with passive learning is reduced from 74.2% to 71.2%, which proves the effectiveness of the active learning.

Fully supervised learning: We train a fully supervised classifier using all 1000 training reviews based on the ADN architecture, which is called ADN with supervised learning and is denoted by "Sup." in Fig. 3. The test accuracies of ADN, and ADN with supervised learning on five datasets are also shown in Fig. 3. Compared with the ADN method, we can see that employing only 100 active learning points enables us to reach nearly the same performance as the fully-supervised method on three datasets. Compared with ADN method, the mean accuracy on five datasets for ADN with supervised learning is just improved from 74.2% to 75.4%. However, the number of required labeled data has been increased from 100 to 1000.

Performance curve: We use KIT dataset to test the performance curve of ADN and IADN with iterations of active learning, the results are shown in Fig. 4. Through the figure, we can see that the performance of IADN is much better than ADN, especially in previous iterations. This proves the effect of information density method. Moreover, with iterations of active leaning, ADN and IADN methods curve quickly.

5.4. Effect of information density

To evaluate the contribution of the information density idea in the proposed IADN methods, we conduct following experiments.

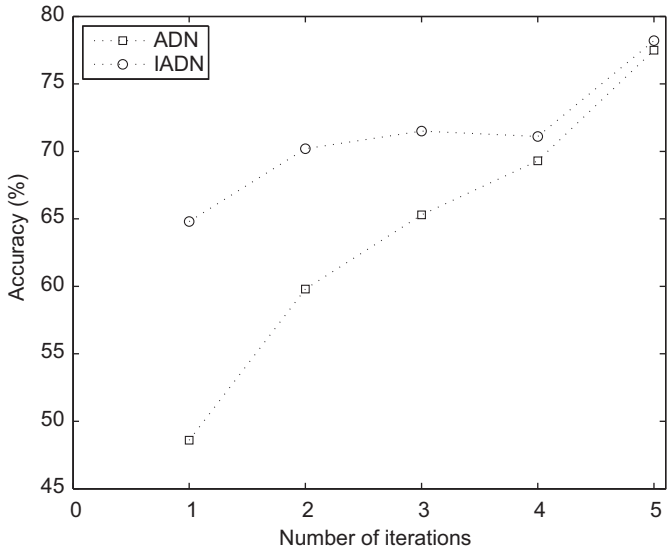

Fig. 4. Performance curve of ADN and IADN with iterations of active learning.

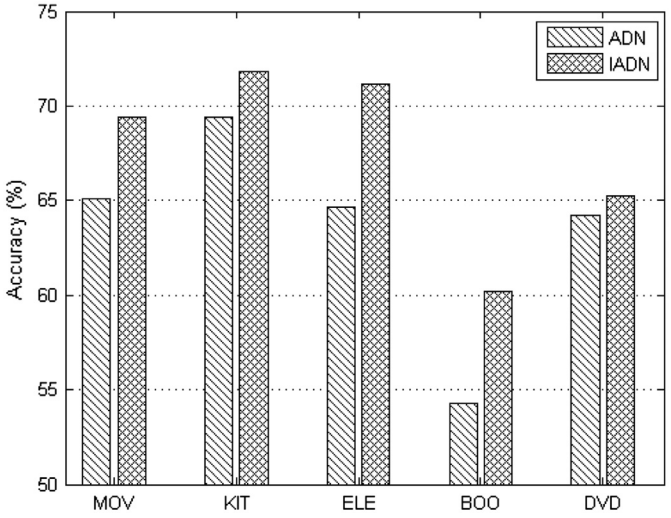

Fig. 5. Test accuracy of ADN and IADN with 10 labeled reviews on five datasets.

The architectures used in this section are the same as Section 5.2. Different from previous experiments, for active learning of ADN and IADN, two of the most uncertain points are selected and labeled in each iteration, after five iterations, 10 labeled reviews are used for training.

The test accuracies of ADN and IADN with 10 labeled reviews on five datasets are shown in Fig. 5. We can see that the performance of IADN is better than ADN in all five datasets. Because for every iteration, just two most uncertain points are selected and labeled. The wrong select of any points can let the performance of the classifier worse. So this experimental setting can emphasize the effect of information density idea.

5.5. Effect of loss function

In ADN and IADN architectures, we use exponential loss function to replace the squared error loss function of classical DBN architecture. Another type of popular loss function is hinge loss function used in SVM. The detail analysis about these loss functions can be seen in [50]. In this part, we just experimentally evaluate the performance of these loss functions for sentiment classification.

The test accuracies of deep architectures with different loss functions on five datasets are shown in Fig. 6. The results show that exponential loss function reaches the best performance on all sentiment datasets, and the differences against the second best methods are statistically significant ($p < 0.05$) with the paired t-test for all five datasets. The performance of squared error loss function is competitive with hinge loss function. This proves the effectiveness of exponential loss function used in ADN and IADN architecture.

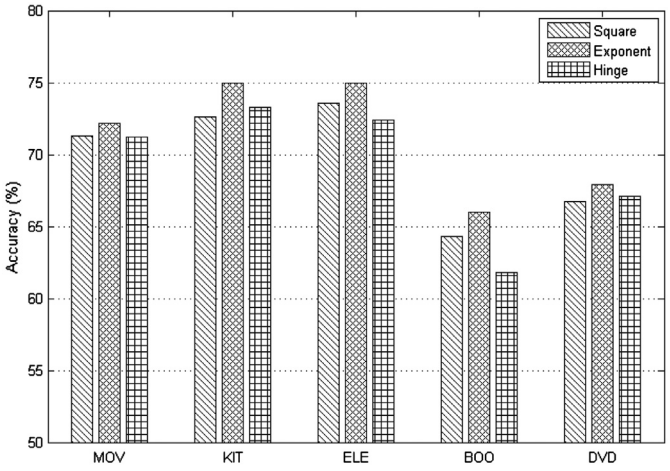

Fig. 6. Test accuracy of deep architecture with different loss function on five datasets.

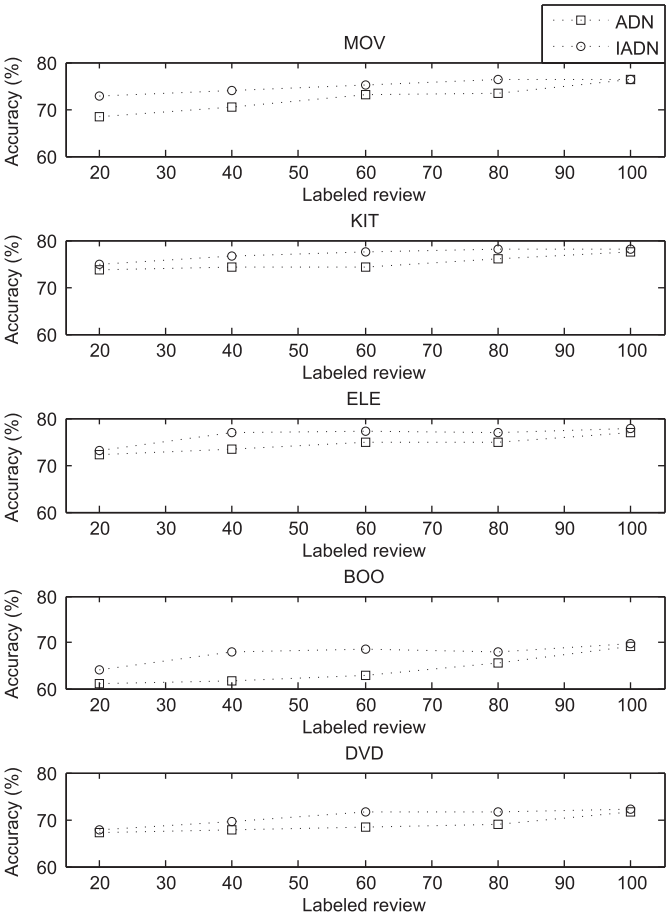

Fig. 7. Test accuracy of ADN and IADN with different number of labeled reviews on five datasets.

5.6. Semi-supervised learning with variance of labeled data

To verify the performance of ADN and IADN with different number of labeled data, we conduct another series of experiments on five datasets and show the results in Fig. 7. The architectures for ADN and IADN used in this experiment are the same as Section 5.2. For both ADN and IADN methods, we repeat 30 times for each experimental setting, and the results are averaged.

Fig. 7 shows that ADN and IADN can reach a relative high accuracy by using just 20 labeled reviews for training. For most of the five sentiment datasets, the test accuracies are increasing slowly while the number of labeled review is growing. It also shows that the performance of ADN is competitive with IADN. In MOV and ELE datasets, the performance of ADN is even better than IADN. Because there are few abnormal reviews in these two datasets, the information density restriction does not take effect in these two experiments. In the other three datasets, the performance of IADN is better than ADN.

5.7. Semi-supervised learning with variance of unlabeled data

To verify the contribution of unlabeled reviews for ADN and IADN methods, we conduct several experiments with different number of unlabeled reviews and 100 labeled reviews. In these experiments, 1000 reviews are used as training data, ADN and IADN can select 100 reviews actively and labeled these reviews for supervised learning. Comparing with the experiments in Section 5.2, we just reduce the number of unlabeled data in unsupervised learning stage. The architectures for ADN and IADN used here are also the same as Section 5.2. For both ADN and IADN methods, we repeat 30 times for each experimental setting, and the results are averaged.

The test accuracies of ADN and IADN with different number of unlabeled reviews and 100 labeled reviews on five datasets are shown in Fig. 8. We can see that ADN and IADN perform well when

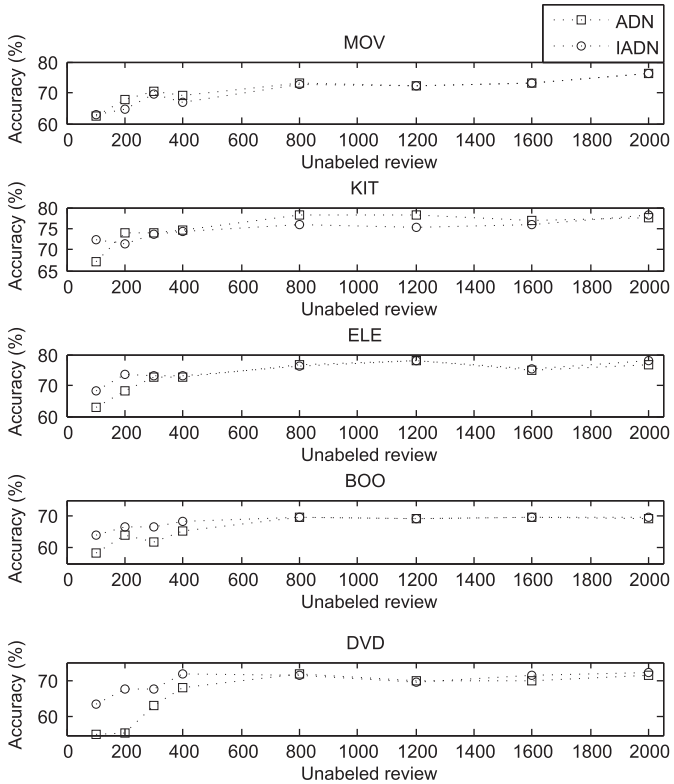

Fig. 8. Test accuracy of ADN and IADN with different number of unlabeled reviews on five datasets.

just using 800 unlabeled reviews. When the number of unlabeled reviews is reduced from 2000 to 800, the performance of ADN and IADN is not worse. For DVD dataset, the performance of ADN and IADN which use 800 unlabeled reviews is better than them using 2000 unlabeled reviews. When the number of unlabeled reviews is reduced from 800 to 100, the performance of ADN and IADN get worse quickly. This proves that ADN and IADN can get competitive performance with just few labeled reviews and appropriate number of unlabeled reviews. Inclusion of unlabeled data does always improve the performance, however, if there are enough unlabeled data, add more unlabeled data just add much time needed for training, the performance will not improve significantly. Considering the much time needed for training with more unlabeled reviews and less accuracy improved for ADN and IADN method, we suggest using appropriate number of unlabeled reviews in real application. In this experiment, the performance of ADN is competitive with IADN too.

6. Conclusions

This paper proposes a novel semi-supervised learning algorithm ADN to address the sentiment classification problem with a small number of labeled reviews. ADN can choose the proper training reviews to be labeled manually, and fully exploit the embedding information from the large amount of unlabeled reviews to improve the robustness of the classifier. We propose a new architecture to guide the output vector of samples locating into different regions of new Euclidean space, and use an exponential loss function to maximize the separability of labeled reviews in global refinement for better discriminability. Moreover, we also propose IADN method, which puts the information density of different reviews into consideration when choosing reviews to be labeled manually.

The performance of ADN and IADN is compared with existing semi-supervised learning methods and deep learning technique. Experiment results show that both ADN and IADN reach better performance than compared methods. We also conduct experiments to verify the effectiveness of ADN and IADN with different number of labeled reviews and unlabeled reviews separately, and demonstrate that ADN and IADN can get competitive classification performance just by using few labeled reviews and appropriate number of unlabeled reviews.

Acknowledgments

This work is supported in part by the Scientific Research Fund of Ludong University (LY2013004) and National Natural Science Foundation of China (Nos. 61173075 and 60973076).

References

[1] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, in: Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, pp. 79–86.
[2] J. Blitzer, M. Dredze, F. Pereira, Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification, in: Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 440–447.
[3] S. Dasgupta, V. Ng, Mine the easy, classify the hard: a semi-supervised approach to automatic sentiment classification, 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, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 701–709.
[4] S. Li, C.-R. Huang, G. Zhou, S.Y.M. Lee, Employing personal/impersonal views in supervised and semi-supervised sentiment classification, in: Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 414–423.
[5] M. Gamon, Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis, in: International Conference on Computational Linguistics, Association for Computational Linguistics, Switzerland, 2004, pp. 841–847.
[6] T. Li, Y. Zhang, V. Sindhwani, A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge, 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, Association for Computational Linguistics, Singapore, 2009, pp. 244–252.
[7] R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, Self-taught learning: transfer learning from unlabeled data, in: International Conference on Machine learning, ACM, Corvallis, Oregon, USA, 2007, pp. 759–766.
[8] X. Zhu, Semi-supervised Learning Literature Survey, Technical Report, University of Wisconsin Madison, Madison, WI, USA, 2007.
[9] M. Wang, X.-S. Hua, T. Mei, R. Hong, G. Qi, Y. Song, L.-R. Dai, Semi-supervised kernel density estimation for video annotation, Comput. Vis. Image Understanding 113 (2009) 384–396.
[10] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, Y. Song, Unified video annotation via multigraph learning, IEEE Trans. Circuits Syst. Video Technol. 19 (2009) 733–746.
[11] Z. Zha, T. Mei, J. Wang, Z. Wang, X. Hua, Graph-based semi-supervised learning with multiple labels, J. Visual Commun. Image Representation 20 (2009) 97–103.
[12] Y. Bengio, Learning Deep Architectures for AI, Technical Report, IRO, Universite de Montreal, 2007.
[13] R. Salakhutdinov, G.E. Hinton, Learning a nonlinear embedding by preserving class neighbourhood structure, J. Mach. Learn. Res. 2 (2007) 412–419.
[14] G.E. Hinton, R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.
[15] M. Ranzato, M. Szummer, Semi-supervised learning of compact document representations with deep networks, in: International Conference on Machine Learning, ACM, Helsinki, Finland, 2008, pp. 792–799.
[16] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, J. Mach. Learn. Res. 2 (2002) 45–66.
[17] M. Wang, X.-S. Hua, Active learning in multimedia annotation and retrieval: a survey, ACM Trans. Intell. Syst. Technol. 2 (2011) 1–21.
[18] Z. Zheng-Jun, W. Meng, Z. Yan-Tao, Y. Yi, H. Richang, C. Tat-Seng, Interactive video indexing with statistical active learning, IEEE Trans. Multimedia 14 (2012) 17–27.
[19] B. Settles, M. Craven, An analysis of active learning strategies for sequence labeling tasks, in: Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, 2008, pp. 1070–1079.
[20] G. Druck, B. Settles, A. McCallum, Active learning by labeling features, in: Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2009, pp. 81–90.
[21] X. Zhu, J. Lafferty, Z. Ghahramani, Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions, in: ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, AAAI, Washington DC, USA, 2003, pp. 58–65.
[22] S. Zhou, Q. Chen, X. Wang, Active deep networks for semi-supervised sentiment classification, in: International Conference on Computational Linguistics, Coling 2010 Organizing Committee, Beijing, China, 2010, pp. 1515–1523.
[23] R. Socher, J. Pennington, E.H. Huang, A.Y. Ng, C.D. Manning, Semi-supervised recursive autoencoders for predicting sentiment distributions, in: Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, UK, 2011, pp. 151–161.
[24] X. Wan, Co-training for cross-lingual sentiment classification, 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, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 235–243.
[25] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inf. Retr. 2, 2008.
[26] K. Dave, S. Lawrence, D.M. Pennock, Mining the peanut gallery: opinion extraction and semantic classification of product reviews, in: International Conference on World Wide Web, ACM, New York, NY, USA, 2003, pp. 519–528.
[27] B. Pang, L. Lee, A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts, in: 42th Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 271–278.
[28] T. Mullen, N. Collier, Sentiment analysis using support vector machines with diverse information sources, in: Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 412–418.
[29] V. Ng, S. Dasgupta, S.M.N. Arifin, Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews, in: 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Sydney, Australia, 2006, pp. 611–618.
[30] R. Mcdonald, K. Hannan, T. Neylon, M. Wells, J. Reynar, Structured models for fine-to-coarse sentiment analysis, in: Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 432–439.

[31] Y. Xia, L. Wang, K.-F. Wong, M. Xu, Lyric-based song sentiment classification with sentiment vector space model, in: Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics, Columbus, Ohio, 2008, pp. 133–136.
[32] S. Li, S.Y.M. Lee, Y. Chen, C.-R. Huang, G. Zhou, Sentiment classification and polarity shifting, in: International Conference on Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 635–643.
[33] Y. Liu, X. Yu, X. Huang, A. An, S-PLASA+: adaptive sentiment analysis with application to sales performance prediction, in: International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2010, pp. 873–874.
[34] W. Wei, J.A. Gulla, Sentiment learning on product reviews via sentiment ontology tree, in: Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 404–413.
[35] A. Aue, M. Gamon, Customizing sentiment classifiers to new domains: a case study, in: International Conference on Recent Advances in Natural Language Processing, RANLP 2005 Organising Committee, Borovets, Bulgaria, 2005.
[36] S. Tan, G. Wu, H. Tang, X. Cheng, A novel scheme for domain-transfer problem in the context of sentiment analysis, in: Conference on Information and Knowledge Management, ACM, New York, NY, USA, 2007, pp. 979–982.
[37] S. Li, C. Zong, Multi-domain sentiment classification, in: 46th Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics, Columbus, Ohio, 2008, pp. 257–260.
[38] S.J. Pan, X. Ni, J.-T. Sun, Q. Yang, Z. Chen, Cross-domain sentiment classification via spectral feature alignment, in: International World Wide Web Conference, ACM, New York, NY, USA, 2010, pp. 751–760.
[39] D. Bollegala, D. Weir, J. Carroll, Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification, in: Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 132–141.
[40] Y. He, C. Lin, H. Alani, Automatically extracting polarity-bearing topics for cross-domain sentiment classification, in: Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 123–131.
[41] A.B. Goldberg, X. Zhu, Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization, in: Proceedings of TextGraphs: The First Workshop on Graph Based Methods for Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, 2006, pp. 45–52.
[42] V. Sindhwani, P. Melville, Document-word co-regularization for semi-supervised sentiment analysis, in: International Conference on Data Mining, IEEE, Pisa, Italy, 2008, pp. 1025–1030.
[43] T. Zagibalov, J. Carroll, Automatic seed word selection for unsupervised sentiment classification of Chinese text, in: International Conference on Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, 2008, pp. 1073–1080.
[44] J. Read, Using emoticons to reduce dependency in machine learning techniques for sentiment classification, in: the Association of Computational Linguistics Student Research Workshop, Association for Computational Linguistics, Ann Arbor, Michigan, 2005, pp. 43–48.
[45] B. Lu, C. Tan, C. Cardie, B.K. Tsou, Joint bilingual sentiment classification with unlabeled parallel corpora, in: Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 320–330.
[46] R. Salakhutdinov, I. Murray, On the quantitative analysis of deep belief networks, in: International Conference on Machine learning, ACM, Helsinki, Finland, 2008, pp. 872–879.
[47] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (2002) 1771–1800.
[48] T. Joachims, Transductive inference for text classification using support vector machines, in: International Conference on Machine Learning, Morgan Kaufmann Publishers, Bled, Slovenia, 1999, pp. 200–209.
[49] S. Kamvar, D. Klein, C. Manning, Spectral learning, in: International Joint Conferences on Artificial Intelligence, AAAI Press, Catalonia, Spain, 2003, pp. 561–566.
[50] Y. Liu, S. Zhou, Q. Chen, Discriminative deep belief networks for visual data classification, Pattern Recognition 44 (2011) 2287–2296.

Shusen Zhou received the Ph.D. degree in computer application technology from the Harbin Institute of Technology in 2012. He is currently an assistant professor in Ludong University. His main research interests include machine learning, artificial intelligence, multimedia content analysis and computational linguistics.

Qingcai Chen received the Ph.D. degree in computer science from the Computer Science and Engineering Department, Harbin Institute of Technology. From September 2003 to August 2004, he worked for Intel (China) Ltd. as a senior software engineer. Since September 2004, he has been with the Computer Science and Technology Department of Harbin Institute of Technology Shenzhen Graduate School as an associate professor. His research interests include machine learning, pattern recognition, speech signal processing, and natural language processing.

Xiaolong Wang received the B.E. degree in computer science from the Harbin Institute of Electrical Technology, Harbin, China, in 1982, the M.E. degree in computer architecture from Tianjin University, Tianjin, China, in 1984, and the Ph.D. degree in computer science and engineering from the Harbin Institute of Technology in 1989. He was an Assistant Lecturer in 1984 and an Associate Professor in 1990 with the Harbin Institute of Technology. From 1998 to 2000, he was a Senior Research Fellow with the Department of Computing, Hong Kong Polytechnic University, Kowloon. He is currently a Professor of computer science with the Harbin Institute of Technology Shenzhen Graduate School. His research interest includes artificial intelligence, machine learning, computational linguistics, and Chinese information processing.