Evolutionary DBN for the Customers' Sentiment Classification with Incremental Rules

Ping Yang, Dan Wang(✉), Xiao-Lin Du, and Meng Wang

Beijing University of Technology, Beijing 100022, People's Republic of China
yangping_sx@163.com, wangdan@bjut.edu.cn

Abstract. An increasing number of reviews from the customers have been available online. Thus, sentiment classification for such reviews has attracted more and more attention from the natural language processing (NLP) community. Related literature has shown that sentiment analysis can benefit from Deep Belief Networks (DBN). However, determining the structure of the deep network and improving its performance still remains an open question. In this paper, we propose a sophisticated algorithm based on fuzzy mathematics and genetic algorithm, called evolutionary fuzzy deep belief networks with incremental rules (EFDBNI). We evaluate our proposal using empirical data sets that are dedicated for sentiment classification. The results show that EFDBNI brings out significant improvement over existing methods.

Keywords: Semi-supervised · Deep learning · Fuzzy set
Sentiment classification · Genetic algorithm

1 Introduction

Various smart appliances have play more and more important roles in our daily life, such as Apple watch, smart car etc. In particular, customers can much easier express their opinions through their smart appliances in different ways. As a result, it is not surprising that nowadays there are tons of reviews available out there. Sentiment classification for such reviews has attracted more and more attention from the natural language processing (NLP) community.

Sentiment analysis refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Sentiment analysis aims to determine the attitude of a speaker with respect to some topic or the overall contextual polarity of a document. such as 'positive' or 'negative', 'thumbs up' or 'thumbs down' [1].

Methods for document sentiment classification are generally based on lexicon and corpus [7, 8]. Lexicon-based approaches can derive a sentiment measure for text based on sentiment lexicons. Corpus-based approaches involve a statistical classification method. The latter usually outperforms the former and has been used in unsupervised learning [9], supervised learning and semi-supervised learning.

P. Perner (Ed.): ICDM 2018, LNAI 10933, pp. 119–134, 2018.
https://doi.org/10.1007/978-3-319-95786-9_9

Early research within this field include Pang et al. [2] and Turney [3], who applied supervised learning and unsupervised learning for classifying sentiment of movie reviews, respectively. In particular, Pang et al. applied different methods based on n-gram grammar and POS (including Naïve Bayes, Maximum Entropy and SVM) to classify a review as either positive or negative. Unsupervised learning for sentiment classification works without any labeled reviews [3]. Although these methods turn out to have good performance, they all rely on labeled data which is normally difficult to obtain. Unsupervised learning of sentiment is difficult, because of the prevalence of sentimentally ambiguous reviews. SO-PMI [4] is a method for inferring the semantic orientation of a word from its statistical association with a set of positive and negative paradigm words. Further more, some scholars apply machine learning approaches to derive a classifier through supervised learning [5, 6].

Several semi-supervised learning approaches have been proposed, which use a large amount of unlabeled data together with labeled data for learning [10–12]. Sindhwani and Melville [10] propose a semi-supervised sentiment classification algorithm, which utilizes lexical prior knowledge in conjunction with unlabeled data. Recently, deep belief networks [11] is an effective model in semi-supervised learning for sentiment classification. DBN(deep belief networks) performs well in semi-supervised learning, and can be used for sentiment classification. To embedding prior knowledge in the network structure, Zhou et al. [13] propose a semi-supervised learning algorithm called fuzzy deep belief networks for sentiment classification, which is based on deep learning algorithm DBN and fuzzy sets [14].

However, there are several defects in existing semi-supervised learning methods. On one hand, they cannot deal with the data near the separating hyper-plane among classes reasonably. On the other hand, it is difficult to determine the correlation between the fuzzy sets and neurons. In this paper, we propose to enhance DBN with fuzzy mathematics and genetic algorithm in order to deal with the aforementioned challenges. In particular, our proposal maps input data to output using fuzzy rules according to their memberships of the fuzzy sets. Specifically, we design a new fuzzy set with special fuzzy rules for the data near the separating hyper-plane among the classes. And we introduced genetic algorithm to determine the correlation between the fuzzy sets and neurons. Our algorithm refers to evolutionary fuzzy deep belief networks with incremental rules (EFDBNI). We conclude that our proposal is able to tackle the aforementioned challenges and brings out better performance over existing approaches.

The remainder of this paper is organized as follows. Section 2 introduces the related work of sentiment classification. Section 3 presents our semi-supervised learning method EFDBNI in details. Section 4 presents the experimental results. We conclude the paper in Sect. 5.

2 Related Work

Many works have been proposed for sentiment classification. According to the dependence on labeled data, methods of sentiment classification fall into three categories: supervised learning, unsupervised learning and semi-supervised learning.

The study supervised learning methods for sentiment classification has begun with the work in [2]. These methods are widely used in analyzing the sentiments of various topics, such as movie reviews [15], product reviews [16, 17], microblogs [18, 19] and so on. The idea is training a domain-specific sentiment classifier for each target domain using the labeled data in that domain. Although these methods turn out to have good performance, they all rely on labeled data as training set which is normally difficult to obtain even though several works use the domain adaptation approach [20–24] as the challenge of domain-specific and annotating a large scale corpus for each domain is very expensive.

Unsupervised learning for sentiment classification is to maximize likelihood of observed data without any labeled reviews [25]. In [3], the classification of a review is predicted by the average semantic orientation of the phrases in the review that contain adjectives or adverbs. In addition, a phrase has a positive semantic orientation when it has good associations (e.g., "subtle nuances") and a negative semantic orientation when it has bad associations (e.g., "very cavalier"). In [26], Read and Carroll investigate the effectiveness of word similarity techniques when performing weakly-supervised sentiment classification. Because labeled data are not used by unsupervised learning approaches, they are expected to be less accurate than those based on supervised learning [27].

Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers [28]. There are many semi-supervised learning methods of sentiment classification presented. To address the sentiment analysis task of rating inference, Goldberg and Zhu [29] present a graph-based semi-supervised learning algorithm which infers numerical ratings based on the perceived sentiment. In [30], a novel semi-supervised sentiment prediction algorithm utilizes lexical prior knowledge in conjunction with unlabeled examples. This method is based on joint sentiment analysis of documents and words based on a bipartite graph representation of the data. Recently, deep belief networks [11] is an effective model in semi-supervised learning for sentiment classification. DBN (deep belief networks) performs well in semi-supervised learning, and can be used for sentiment classification. To embedding prior knowledge in the network structure, Zhou et al. [13] propose a semi-supervised learning algorithm called fuzzy deep belief networks for sentiment classification. In [31], a novel sentiment analysis model is proposed based on recurrent neural network, which takes the partial document as input and then the next parts to predict the sentiment label distribution rather than the next word. In this paper, we focus on document level sentiment classification.

3 Method

We describe the procedure for training an Evolutionary Fuzzy Deep Belief Network with Incremental rules (EFDBNI). Suppose that we construct a EFDBNI with one input layer, one output layer and $N - 1$ hidden layers. Firstly, we preprocess the sentiment classification data set. Secondly, we train normal deep belief networks with one input layer, one output layer and $N - 1$ hidden layers. Thirdly, we define fuzzy sets associated with fuzzy rules. In addition, we construct special hidden layers corresponding to

these fuzzy sets based on the deep belief networks mentioned above. Then we will get a new fuzzy deep believe networks by applying membership functions to the deep structure. The whole procedure is shown in Fig. 1.

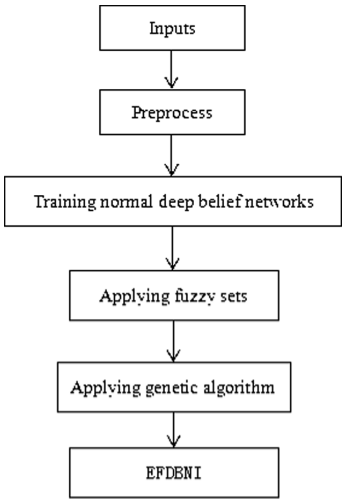

Fig. 1. The whole procedure for establishing an EFDBNI.

3.1 Preprocess

As the sentiment classification data set is normally composed of many review documents to a bag of words, we need to preprocess them in advance in the same way as that of [32].

3.2 Normal Deep Belief Networks

Preprocessed as mentioned above, each review is represented as a vector of binary weight x_i. If the *j*th word of the vocabulary is in the *i*th review, then we will set $x_j^i = 1$; otherwise, $x_j^i = 0$. Then the data set is denoted by

$$X = [x^1, x^2, \ldots, x^{R+T}] \tag{1}$$

where

$$x^i = [x_1^i, x_2^i, \ldots, x_D^i], i \in 1, 2, \ldots, R+T \tag{2}$$

where R is the amount of training reviews, T is the amount of test reviews, D is the amount of feature words in the data set.

The L training reviews to be labeled manually is denoted by X^L. These reviews are chosen randomly. The labels corresponding to L labeled training reviews are aggregated into a set of labels Y. And it is denoted as

where

$$Y = [y^1, y^2, \ldots, y^L] \tag{3}$$

$$y^i = [y_1^i, y_2^i, \ldots, y_c^i]' \tag{4}$$

$$y_j^i = \begin{cases} 1, x \in jth\ class \\ 0, x \notin jth\ class \end{cases} \tag{5}$$

c is the number of classes. In sentiment classification, if a review x_i is positive, $y_i = [1, 0]'$; otherwise $y_i = [0, 1]'$.

We construct DBN with one input layer, one output layer and N − 1 hidden layers, while the desired EFDBNI also has one input layer, one output layer and N − 1 hidden layers. In both of the DBN as semi-manufacture and the EFDBNI as the made-up article, the input layer h^0 has D units and the output layer has C units. The output layer has a linear activation function. And every hidden layer uses a sigmoid function as its activation function.

We train the DBN using all reviews as inputs. Firstly, we build the DBN layer by layer using RBM through the traditional algorithm [33]. Each RBM is consisted of a binary input layer and a binary output layer [34]. Secondly, we refine the parameter space W using L labeled reviews by back-propagation. In this task, we define the optimization problem as

$$\arg\min_w f(h^N(X^L, Y^L)) \tag{6}$$

$$f(h^N(X^L), Y^L) = \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{C}(h_j^N(x^i) - y_j^i)^2 \tag{7}$$

where C is the number of classes and it is equal to 2 in the case of sentiment classification.

The layer h^N is obtained as follows:

$$h_t^N(x) = c_t^N + \sum_{s=1}^{D_{N-1}} w_{st}^N h_s^{N-1}(x), t = 1, 2, \ldots, D_N \tag{8}$$

$$h_t^k(x) = sigm(c_t^k + \sum_{s=1}^{D_{k-1}} w_{st}^k h_s^{k-1}(x)), t = 1, 2, \ldots, D; k = 1, 2, \ldots, N-1 \tag{9}$$

where w_{st}^k is the symmetric interaction term between unit s in the layer h^{k-1} and unit t in the layer h^k, k = 1, 2, … N − 1, while c_t^k is the tth bias of layer h^k. And w_{st}^N is the symmetric interaction term between units in the layer h^{N-1} and unit t in the layer h^N. c is the tth bias of layer h^N.

3.3 Membership Function of Fuzzy Rule

Training reviews are divided into three fuzzy sets A, B and C, each of which is inferred using the fuzzy rules.

Definition 1. *For a data set X, positive fuzzy set A in X is characterized by a membership function $\mu_A(x)$ which associates each review x with a real number in the interval [0, 1], representing the grade of x as positive review in A:*

$$A = \{(x, \mu_A(x)); x \in X\} \tag{10}$$

where

$$\mu_A(x) : X \to [0, 1] \tag{11}$$

Definition 2. *For a data set X, negative fuzzy set B in X is characterized by a membership function μ_B (x) which associates each review x with a real number in the interval [0, 1], representing the grade of x as negative review in B:*

$$B = \{(x, \mu_B(x)); x \in X\} \tag{12}$$

where

$$\mu_B(x) : X \to [0, 1] \tag{13}$$

The membership functions are based on the value of $h^N(x)$ from the deep belief networks trained in Sect. 2.1. In the case of sentimental classification, the dimension of $h^N(x)$ is 2(corresponding to positive or negative class). Thus, the class separation line is

$$h_1^N = h_2^N \tag{14}$$

The distance between a point $h^N(x_i)$ and a separation line is

$$d(x^i) = (h_1^N(x^i) - h_2^N(x^i))/\sqrt{2} \tag{15}$$

If $d(x^i) > 0$, x^i is positive; otherwise, x^i is negative.

The two membership functions $\mu_A(x)$ and $\mu_B(x)$ are expressed as

$$\mu_A(x; \beta, \gamma) = \begin{cases} S(d(x); \gamma - \beta, \gamma - \beta/2, \gamma), & d(x) \le \gamma \\ 1, & d(x) > \gamma \end{cases} \tag{16}$$

$$\mu_B(x; \beta, -\gamma) = \begin{cases} 1, & d(x) < -\gamma \\ 1 - S(d(x); -\gamma, -\gamma + \beta/2, -\gamma + \beta), & d(x) \ge -\gamma \end{cases} \tag{17}$$

$$S(d; \alpha, \beta, \gamma) = \begin{cases} 0, & d \le \alpha \\ 2(\frac{d-\alpha}{\gamma-\alpha})^2, & \alpha \le d \le \beta \\ 1 - 2(\frac{d-\gamma}{\gamma-\alpha})^2, & \beta \le d < \gamma \\ 1, & d \ge \gamma \end{cases} \tag{18}$$

If $\mu_A(x) > \mu_B(x)$ as $d(x) > 0$, the grade of membership in A is bigger than in B. If $\mu_A(x) < \mu_B(x)$ as $d(x) < 0$, the grade of membership in A is smaller than in B.

To estimate two parameters β and γ, we have

$$\gamma = \max(d(x^i)) \tag{19}$$

$$\beta = \xi \times \gamma, \xi \geq 2 \tag{20}$$

where ξ is a constant which indicates the degree of separation for the two classes.

However, it is not proper to partition the data set in this way. These fuzzy sets only model the two sentimental polarities and their fuzzy rules. Then, the input data is mapped to the output using the fuzzy rules according to their memberships of the fuzzy sets. The higher degree the input data belongs to a fuzzy set, the more proper to use its corresponding rules. In contrast, although the fuzzy sets theory allows an input data locate at the overlapping among several fuzzy sets, there are no reasonable rules can be applied to the input if it belongs to every set in a low degree. In another word, it leads to inexact results to use the existing fuzzy rules for inference with the data near the separating hyper-plane. Hence, we design a new fuzzy set with special fuzzy rules for the data near the separating hyper-plane in this paper. This new rule could compensate for the previous rules.

Definition 3. *For a data set X, neutral fuzzy set C in X is characterized by a membership function $\mu_C(x)$ which associates each review x with a real number in the interval [0, 1], representing the grade of x as neutral review in C:*

$$C = \{(x, \mu_C(x)); x \in X\} \tag{21}$$

According to set theory, we reformulate the definition of fuzzy set C as follows:

$$C = A^c \cap B^c \tag{22}$$

where X^c represents the complementary set of a set X.

On the base of fuzzy mathematics, we have

$$\mu_C(x) = \min(1 - \mu_A(x), 1 - \mu_B(x)) \tag{23}$$

To be specific, the membership function $\mu_C(x)$ is formalized as

$$\mu_C(x) = \begin{cases} 1, d \leq \gamma \\ 1 - 2[\frac{d-(\gamma-\beta)}{\beta}]^2, \gamma < d \leq \gamma - \frac{\beta}{2} \\ 2(\frac{d-\gamma}{\beta})^2, \gamma - \frac{\beta}{2} < d \leq -\gamma + \frac{\beta}{2} \\ 1 - 2[\frac{d+(\gamma-\beta)}{\beta}]^2, -\gamma + \frac{\beta}{2} < d \leq \gamma \\ 0, \gamma < d \end{cases} \tag{24}$$

The parameters β and γ in (24) are as the same as the two parameters in (16) and (17). Thus, the parameters β and γ in (24) can also be estimated by (19) and (20).

3.4 Evolutionary Fuzzy Deep Belief Networks Algorithm with Incremental Rules

After extracting fuzzy parameters, deep architecture has been constructed. The architecture is shown in Fig. 2. The top hidden layer h^{N-1} is divided into three parts corresponding to each of the fuzzy rules respectively.

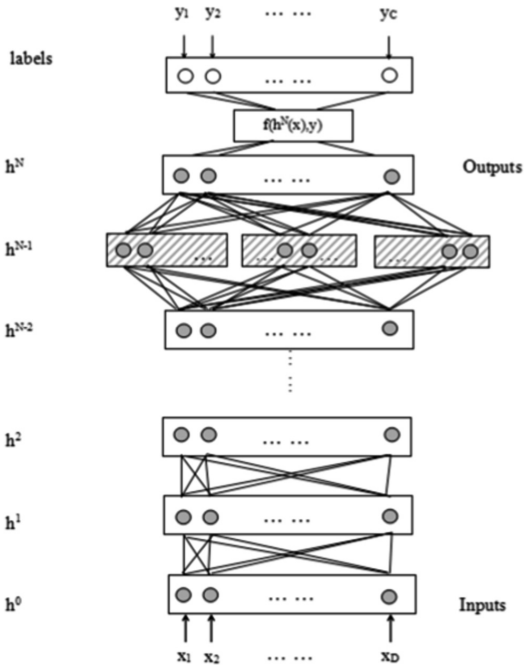

Fig. 2. Architecture of fuzzy deep belief networks.

The label of new data is denoted by $\hat{j}$. It is determined by

$$\hat{j} = \arg\max_{j} h^N(x) \tag{25}$$

The procedure of getting final outputs of the networks using $\mu_A(x)$, $\mu_B(x)$, $\mu_C(x)$ and outputs from h^{N-1} is formulated as

$$h_t^N(x) = c_t^N + \mu_A(x)P + \mu_B(x)Q + \mu_C(x)R, t = 1, 2, \ldots, D_N \tag{26}$$

where

$$P = \sum_{s=1}^{D_{N-1}/3} w_{st}^N h_s^{N-1}(x) \tag{27}$$

$$Q = \sum_{s=D_{N-1}/3+1}^{2D_{N-1}/3} w_{st}^{N} h_{s}^{N-1}(x) \tag{28}$$

$$R = \sum_{s=2D_{N-1}/3+1}^{D_{N-1}} w_{st}^{N} h_{s}^{N-1}(x) \tag{29}$$

where the description of h^{N-1} can be seen in (9). However, we use genetic algorithm to determine which of units in layer h^{N-1} are used to represent each rule.

That is we determine the indexes of the hidden units in layer involve in the (27), (28) and (29). In each of the equations, the index is denoted by s. We divide the units in layer into three groups. Each of the groups is associated with one of the fuzzy rules. According to the conclusion in [35], it is helpful to determine the units associated with each fuzzy rule. As such, we need some principles to determine the features represented by the units in layer h^{N-1} associated with each fuzzy rule. Therefore, we implement this process by applying genetic algorithm (GA) for grouping problem. The fuzzy rules are groups. The features are objects to be grouped.

In our work, we only need to consider the equal group-size problem. Different from previous genetic algorithm for maximally diverse grouping problems [36, 37], our purpose is to assign the right fuzzy rules to the features represented by the units in layer h^{N-1}. Thus, we design a novel genetic algorithm which tackles this particular problem in our paper. The details are described as follows:

We divide the units in layer h^{N-1} into three manually disjoint groups. These groups are numbered with one, two and three.

As the previous genetic algorithm for grouping problems, we encode the solution as chromosome. In our work, it is suitable to use any encoding scheme which can represent the space of solutions. Thus, we adopt the most straightforward encoding scheme, namely one gene per object. For example, the chromosome 132312 would encode the solution where the first object is in group 1, the second in group 3, third in 2, fourth in 3, fifth in 1, sixth in 2.

Although there are various compositions of the groups, we can simplify all kinds of diversities among the compositions into two types. We denote three objects in three groups respectively by A, B and C. Group 1, Group 2 and Group 3 include object A, object B and object C respectively. These relationships are described in Table 1.

Table 1. Units for magnetic properties.

Group number	Index of object
1	A
2	B
3	C

Another solution to the grouping problem is to assign these three objects to different groups. We list all the cases in Table 2.

If we ignore the differences caused by capital letter (for example, the pair of ABC and BAC is equivalent to the pair of CBA and BCA), we can further simplify the cases into three. This is described in Table 3.

The differences between these solutions are induced to the assignment (grouping) of each object. So the differences can be classified into two types described in Table 3. For example, a solution is represented by chromosome 123123, while another solution is 213312. On the one hand, the diversity between the first three genes 123 of the first chromosome and the first three genes 213 of the second chromosome is the case described in the second row (the row includes the phase 'group number' is the first row) and third row in Table 3. On the other hand, the diversity between the last three genes 123 of the first chromosome and the last three genes 312 of the second chromosome is the case described in the second row and fourth row in Table 3. So we can break up a pair of solution according to their consistent part and inconsistent part. Based on this fact, we design specific GA operators for our problem. The details are described as follows.

Step 1. *We denote crossover rate as φ. The parameter ranges from 0.5 to 1. And it is determined manually.*
Step 2. *We classify the diversities between the parents into two types in Table 3. And these diversities are integrated into a set. We assign a random number each terms of the set. The number ranges from 0 to 1. If a term's number is bigger than φ, it is added to the list for crossover.*
Step 3. *We make duplication of parents.*
Step 4. *We take the top term out of the list. And we exchange the objects in the duplication of the first parent. We only need to regroup the objects involved in the diversity term. Then the assignment of the objects are as the same as that in the second parent. At the same time, we make the assignment of the three objects in the duplication of the second parent to be as the same as the first parent. Then we repeat the process described above until the list is empty.*

It should be note that the crossover rate φ is larger than 0.5. Because if it is too small, the process may create springs which are the same as parents. Since small crossover rate may lead to all of the diversities are added into the list. Then the duplication of a parent change into the chromosome which is the same as the other parent.

We redefine the mutation operator as follows:

Step 1. *We denote mutation rate as ϕ. The parameter ranges from 0 to 0.5. And it is determined manually.*
Step 2. *Generate a random number with uniform distribution. If the number is bigger than ϕ, we will go to the next step; otherwise, quite the process.*
Step 3. *Pick up two groups for mutation randomly. Here, the three groups have the same probabilities to be chosen. Then mutation occurs in each of the group at a random position. The probability of a mutation of a bit in a group is equal to* $3/D^{N-1}$, *where* D^{N-1} *is the number of units in layer* h^{N-1}. *After we determine the positions, we exchange the group number of the two objects.*

It should be note that in step 3 we keep the size of each group constantly. And we ensure that an object is only included in one group.

To evaluate the solution, we need a fitness function. Different from previous genetic algorithm for maximally diverse grouping problems, our purpose is to assign the right fuzzy rules to the features represented by the units in layer h^{N-1}. So we design a new fitness function for our problem. Further, we use the error rate of the networks that have the structure described by a solution as the criterion for fitness. To obtain the initial population, we select a number of solutions from whole possible solutions. To obtain the initial population, we select a number of solutions from whole possible solutions. It should be noted that according to the work in [35] the characteristics of a fuzzy model depend heavily on the structures rather than on the parameters of the membership functions. Further, they confirm that it is practical to select the structures before the identification of the parameters of the membership functions. In our proposed GA, we only need to determine the connections between layer h^{N-1} and h^N. In the neural networks, the structure from layer h^0 to h^{N-1} represents the membership functions. And the parameters of this structure have been well adjusted for the learning in Sect. 2.1. Thus, for one generation, we just use gradient-descent to adjust the parameters for symmetric interaction term between layer h^{N-1} and layer h^N as well as the bias of layer h^N, i.e., w^N and b^N. The process of GA is shown in Algorithm 1. The number of a generation is variable gen. The index of population is i.

Table 2. The composition of groups.

Group number	Index of object	Index of object	Index of object	Index of object	Index of object	Index of object
1	A	B	A	C	C	B
2	B	A	C	B	A	C
3	C	C	B	A	B	A

Table 3. The composition of groups.

Group number	Index of object	Index of object	Index of object
1	A	B	C
2	B	A	A
3	C	C	B

Algorithm 1. Genetic Algorithm.

```
Initialization:initial population of τ individuals,set max generation as maxGen (an
integer),selection rate ω;
for gen in 1 to maxGen
    for individual i in population
        Train the neural networks, whose structure is determined by the solution
          represented by individual i;
        Calculate the fitness value of individual i by error rate;
    end
    Resort the individuals in descending order by fitness value;
    Select top ω×p individuals as parents into a set p.
    j=1;
    while j<0.5×ω×p
        Select a pair of parent individuals randomly from set p;
        Implement crossover and mutation to the parents and create two
          off-springs;
        Replace two worst individuals in the current population by the off-springs.
        j++;
    end
end
```

After we determine the structure of the whole networks using GA, we will identify all parameters of the whole fuzzy deep belief networks. This is supervised learning. The EFDBNI algorithm also uses gradient-descent to retrain the parameters through the whole deep architecture. The optimization problem is the same as the one described in (6) and (7). Hence, the whole fuzzy deep belief networks algorithm is shown in Algorithm 2.

Algorithm 2. Algorithm of EFDBNI.

Input: data X, Y^L;
number of units in every hidden layer $D_1,D_2,\ldots D_N$;
number of layers N; number of epochs Q;
number of training data R; number of test data T;
hidden layer $h^1,h^2,\ldots h^{N-1}$;
parameter space $W=\{w^1,w^2,\ldots,w^N\}$;biases b, c;
Output: deep architecture with parameter space W
1.Estimated parameters of $\mu_A(x),\mu_B(x)$ and $\mu_C(x)$
(1) Greedy layer-wise unsupervised learning using Eq.(8) and Eq.(9).
(2) Supervised learning with DBN architecture using Eq. (6) and (7).
(3) Estimate parameters based on Eqs. (19) and(20).
2. Refine the EFDBNI using X^L, Y^L,$\mu_A(x),\mu_B(x)$ and $\mu_C(x)$
(1) Use GA to determine the architecture of the EFDBNI.
(2) Supervised learning with EFDBNI architecture using Eq. (6) and (7).
(3) Classify the reviews based on the trained FDBN architecture using Eq. (25).

4 Results and Discussion

4.1 Experimental Setup

We use three sentiment classification data sets. The data sets includes electronics (ELE), restaurants (RES) and movies (MOV). Each of them contains 1000 positive and 1000 negative reviews.

We divide the 2000 reviews into two parts. Half of the reviews are randomly selected as training data and the remaining reviews are used for testing. Only 10% of the training reviews are labeled.

We set all of the neural networks consist of one input layer, one output layer and three hidden layers. And there are 100, 100, and 200 hidden units in the three hidden layers respectively. However, the structures of the neural networks are different among the three data sets. Because the number of units in the input layer is the same as the dimensions of each data set. The max number of iterations of unsupervised-learning is set to 1000, and the supervised-learning is repeat for 10 times for each labeled data.

The parameter ξ is set by experience for different data sets. When $\xi = 3$, EFDBNI can get relatively better results on all data sets. Thus, we set ξ as 3. Here, we set the max generation of GA as 100.

4.2 Performance Comparison

We compare the classification performance of EFDBNI with two representative semi-supervised learning classifiers, i.e., transductive SVM (TSVM) [38] and Fuzzy deep belief networks (FDBN) [13].

The test accuracy on three data sets with three rules can be seen in Fig. 3, we can see that the performance of the proposed method is better than the others on all three data sets.

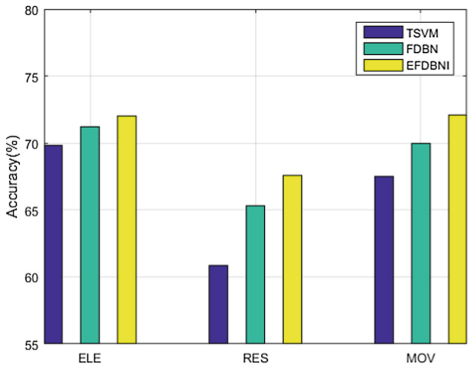

Fig. 3. Test accuracy with 100 labeled reviews on three data sets for TSVM, FDBN and EFDBNI.

5 Conclusion

This paper proposes a novel semi-supervised learning algorithm EFDBNI to address the sentiment classification problem with a small number of labeled reviews. Our proposal inherits the advantages of previous works about deep learning for sentimental classification, and has significantly improved the performance of existing deep learning architecture.

Acknowledgement. This work was supported by Beijing Natural Science Foundation P.R. China (4173072).

References

1. Li, S., Lee, S.Y.M., Chen, Y., Huang, C., and Zhou, G.: Sentiment classification and polarity shifting. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 635–643 (2010)
2. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of EMNLP-2002, pp. 79–86 (2002)
3. Turney, P.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Annual Meeting of the Association of Computational Linguistics, pp. 417–424 (2002)
4. Turney, P., Littman, M.: Measuring praise and criticism: inference of semantic orientation from association. ACM Trans. Inf. Syst. **21**, 315–346 (2003)
5. Mcdonald, R., Hannan, K., Neylon, T., Wells, M., Reynar, J.: Structured models for fine-to-coarse sentiment analysis. In: Annual Meeting of the Association of Computational Linguistics, pp. 432–439 (2007)
6. Xia, Y., Wang, L., Wong, K.-F., Xu, M.: Lyric-based song sentiment classification with sentiment vector space model. In: Annual Meeting of the Association of Computational Linguistics, pp. 133–136 (2008)
7. Wan, X.: Co-training for cross-lingual sentiment classification. In: IEEE 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, pp. 235–243 (2009)
8. Pang, B., Lee, L.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Annual Meeting of the Association of Computational Linguistics, pp. 271–278. Association for Computational Linguistics, Barcelona (2004)
9. Zagibalov, T., Carroll, J.: Automatic seed word selection for unsupervised sentiment classification of Chinese text. In: International Conference on Computational Linguistics, pp. 1073–1080 (2008)
10. Sindhwani, V., Melville, P.: Document-word co-regularization for semi-supervised sentiment analysis. In: IEEE International Conference on Data Mining, pp. 1025–1030 (2008)
11. Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)
12. Zhou, S., Chen, Q., Wang, X.: Active deep networks for semi-supervised sentiment classification. In: International Conference on Computational Linguistics, Poster, pp. 1515–1523 (2010)
13. Zhou, S., Chen, Q., Wang, X.: Fuzzy deep belief networks for semi-supervised sentiment classification. Neurocomputing **131**, 312–322 (2014)
14. Zadeh, A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
15. Zhuang, L., Jing, F., Zhu, X.-Y.: Movie review mining and summarization. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 43–50. ACM (2006)
16. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177 (2004)

17. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th International Conference on World Wide Web, pp. 519–528. ACM (2003)
18. Go, A., Bhayani, R., Huang, L.: Twitter Sentiment Classification Using Distant Supervision. CS224N Project Report, pp. 1–12. Stanford (2009)
19. Wu, F., Song, Y., Huang, Y.: Microblog sentiment classification with contextual knowledge regularization. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2332–2338 (2015)
20. Aue, A., Gamon, M.: Customizing sentiment classifiers to new domains: a case study. In: International Conference on Recent Advances in Natural Language Processing (2005)
21. Tan, S., Wu, G., Tang, H., Cheng, X.: A novel scheme for domain-transfer problem in the context of sentiment analysis, pp. 979–982 (2007)
22. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In: Annual Meeting of the Association of Computational Linguistics, pp. 440–447 (2007)
23. Li, S., Zong, C.: Multi-domain sentiment classification. In: Annual Meeting of the Association of Computational Linguistics, pp. 257–260. Association for Computational Linguistics (2008)
24. Pan, S.J., Ni, X., Sun, J., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: International World Wide Web Conference, pp. 751–760. ACM (2010)
25. Li, S., Huang, C., Zhou, G., Lee, S.Y.M.: Employing personal/impersonal views in supervised and semi-supervised sentiment classification. In: Annual Meeting of the Association for Computational Linguistics, pp. 414–423. Association for Computational Linguistics, Uppsala (2010)
26. Read, J., Carroll, J.: Weakly supervised techniques for domain-independent sentiment classification. In: Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion, TSA 2009, pp. 45–52. ACM, New York (2009)
27. Silva, D.N., Coletta, L., Hruschka, E., Hruschka, E.J.: Using unsupervised information to improve semi-supervised tweet sentiment classification. Inf. Sci. **355–356**, 348–365 (2016)
28. Zhu, X.: Semi-supervised learning literature survey. Ph.D. thesis (2007)
29. Goldberg, A.B., Zhu, X.: 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, pp. 45–52. Association for Computational Linguistics (2006)
30. Sindhwani, V., Melville, P.: Document-word co-regularization for semi-supervised sentiment analysis. In: International Conference on Data Mining, pp. 1025–1030. IEEE, Pisa (2008)
31. Rong, W., Peng, B., Ouyang, Y., Li, C., Xiong, Z.: Structural information aware deep semi-supervised recurrent neural network for sentiment analysis. Front. Comput. Sci. **9**(2), 171–184 (2015)
32. Dasgupta, S., Ng, V.: 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, pp. 701–709 (2009)
33. Bengio, Y.: Learning deep architecture for AI. Found. Trends Mach. Learn. **2**, 1–127 (2009)
34. Smolensky, P.: Information processing in dynamical systems: foundations of harmony theory. In: Parallel Distributed Processing: Explorations in the Micro structure of Cognition, vol. 1, pp. 194–281 (1986)

35. Lin, C.T., Lee, C.S.G.: Neural-network-based fuzzy logic control and decision system. IEEE Trans. Comput. **40**, 1320–1336 (1991)
36. Falkenauer, E.: A genetic algorithm for grouping. In: Proceedings of the Fifth International Symposium on Applied Stochastic Models and Data Analysis, pp. 198–206 (1991)
37. Smith, D.: Bin packing with adaptive search. In: Proceedings of the First International Conference on Genetic Algorithms and Their Applications, pp. 202–207 (1985)
38. Kamvar, S., Klein, D., Manning, C.: Spectral learning. In: International Joint Conferences on Artificial Intelligence, pp. 561–566. AAAI Press, Catalonia (2003)