Constructive Semi-Supervised Classification Algorithm and Its Implement in Data Mining

Arvind Singh Chandel[1], Aruna Tiwari[1], and Narendra S. Chaudhari[2]

[1] Department of Computer Engg.
Shri GS Inst of Tech.& Sci.
SGSITS, 23, Park Road,
Indore (M.P.) India 452001
atiwari@sgsits.ac.in, asc_chandel@rediffmail.com
[2] Department of Computer Science and Engineering (CSE)
IIT, Indore
M-Block, IET-DAVV Campus,
Khandwa Road, Indore-452017(M.P.)
NSchaudhari@iitbombay.org, NSC183@gmail.com

Abstract. In this paper, we propose a novel fast training algorithm called Constructive Semi-Supervised Classification Algorithm (CS-SCA) for neural network construction based on the concept of geometrical expansion. Parameters are updated according to the geometrical location of the training samples in the input space, and each sample in the training set is learned only once. It's a semi-supervised based approach, the training samples are semi-labeled i.e. for some samples, labels are known and for some samples, data labels are not known. The method starts with clustering, which is done by using the concept of geometrical expansion. In clustering process various clusters are formed. The clusters are visualizes in terms of hyperspheres. Once clustering process over labeling of hyperspheres is done, in which class is assigned to each hypersphere for classifying the multi-dimensional data. This constructive learning avoids blind selection of neural network structure. The method proposes here is exhaustively tested with different benchmark datasets and it is found that, on increasing value of training parameters number of hidden neurons and training time both are getting decrease. Through our experimental work we conclude that CS-SCA result in simple neural network structure by less training time.

Keywords: Semisupervised classification, Geometrical Expansion, Binary Neural Network, Hyperspheres.

1 Introduction

Constructive learning begins with a minimal or empty structure, and dramatically increases the network by adding hidden neurons until a satisfactory solution is found. Numbers of Constructive learning algorithms are available to overcome the problem of traditional algorithms for classification. It includes Fast Covering Learning Algorithm (FCLA) for Supervised Learning [2], Constructive Unsupervised Learning

S. Chaudhury et al. (Eds.): PReMI 2009, LNCS 5909, pp. 62–67, 2009.

Algorithm (CULA) [1], Constructive set Covering learning algorithm (CSCLA) by Ma and Yang [4], Boolean-like training algorithm (BLTA) by Gray and Michel [5], expand and truncate learning algorithm (ETL) by Kim and Park [3]. BLTA is a dynamic technique and derives its original principle from Boolean algebra with extension. ETL finds a set of required separating hyperplanes and automatically determines a required number of neurons in the hidden layer based on geometrical analysis of the training set. CSCLA was proposed based on the idea of weighted Hamming distance hypersphere. In general, ETL, IETL, CSCLA, have no generalization capability. BLTA has generalization capability, but needs more hidden neurons.

Moreover to it, FCLA is supervised learning algorithm, thus labeled samples are used for learning but labeled data sample are expensive to obtain as they require the effort of experienced human annotators. On the other hand unlabeled data samples are easy to obtain but there are very few way to process them. Thus approach of semi-supervised was used that uses large amount of unlabeled data sample with small amount of labeled data sample to build the classifier.

In this paper, we propose a novel fast training algorithm called Constructive semi-supervised classification Approach (CS-SCA) for neural network construction. The proposed method is implemented using two processes, first is clustering and second is labeling. We illustrate the advantages of CS-SCA by using it in classification problems. There are various features of CS-SCA like it is a semi-supervised constructive approach. Sample reordering is allowed in proposed classifier and because of reordering, learning is fast in this approach. As we know that CS-SCA is a semi-supervised approach that's why it requires less human effort. This CS-SCA approach is tested with number of benchmark datasets and compared with SVM [6] based classifier.

The paper is organized as follows. Section 2 gives an overview of CS-SCA. Section 3 explains the method for CS-SCA in detail give algorithmic formulation of our methodology. In section 4, we give experimental results to demonstrate the usefulness of our approach; it also contains detail of data preparation. These experimental results include two well-known datasets [7], namely, Riply dataset and Wisconsin Breast cancer dataset. Finally, in section 5, we give concluding remarks.

2 Overview of CS-SCA

2.1 Basic Concept

Boolean functions have the geometrical property which makes it possible to transform non-linear representation to linear representation for each hidden neuron. We consider a Boolean function with n input and one output,

$$y = f(x_1, x_2, ..., x_n),$$

$$\text{Where } y \in (0,1) \text{ and } x_i \in (0,1), i = (1.....n).$$

These 2^n binary patterns $(0,1)^n$ can be considered as a $n-$dimensional unit hypercube.This ex-hypersphere is defined as the reference hypersphere (RHS) [5] as follows:

$$(x_1 - 1/2)^2 + (x_2 - 1/2)^2 + ... + (x_n - 1/2)^2 = n/4. \quad (1)$$

2.2 Network Construction

CS-SCA constructs a three-layered feed forward neural network with an input layer, a hidden layer and an output layer, as shown in Fig-1. We illustrate the advantages of CS-SCA by its implement in classification problems.

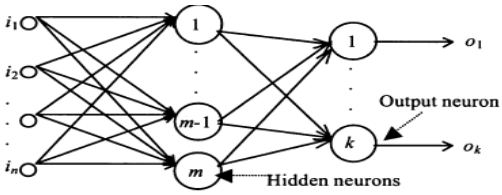

Fig. 1. Neural Network Structure by CS-SCA

3 Proposed Method: CS-SCA

CS-SCA begins with an empty hidden layer. To learn a sample, CS-SCA either adds a new hidden neuron to represent it or updates the parameters of some hidden neuron by expanding its corresponding hypersphere. This is done by clustering process and once clustering gets over by using the concept of majority voting labeling of hypersphere is done.

3.1 Clustering Process

CS-SCA constructs a three-layered feed forward neural network, of which first layer represent to the input data sample that will be in the binary coded format. Then input data samples will be grouped into various clusters. The middle layer of network architecture represents the hyperspheres(hidden neuron).A hidden neuron in Fig. 1 represents a corresponding hyper sphere with center c and radius r_1 . While constructing a hidden neuron, suppose that $\{x^1, x^2, ..., x^v\}$ are v (true) sample included in one hyper sphere (hidden neuron). In terms of these samples, the center is defined as the gravity center $c = (c_1, c_2, ..., c_n)$;

$$c_i = \sum_{k=1}^{v} \frac{x_i^k}{v} \tag{2}$$

The radius r_1 is defined as the minimal Euclidean distance such that all the v vertices are exactly in or on the surface of the corresponding hyper sphere.

$$r_1 = \min_{j=1}^{v} \| x^j - c \| = \min_{j=1}^{v} (\sum_{i=1}^{n} (x_i^k - c_i)^2)^{1/2}$$

Where n is the dimension of the input and ||*|| is the euclidean distance.

Given c and r_1 we can separate these v true sample from the remaining samples. In another words, this corresponding hypersphere represents these v true samples.

Two secondary central radii r_2 and r_3 are introduced to find compact cluster. Samples should be a compact cluster where $r_1 < r_2 < r_3$.

CS-SCA begins with an empty hidden layer. To construct the neural network, we examine whether a coming "true" sample can be covered by one of the existing hidden neurons. When the first sample x^1 comes, the hidden layer is empty and no hidden neuron covers this sample. A new hidden neuron, the first hidden neuron, is created to represent it. This new created hidden neuron represents a hyper sphere centered at x^1. Samples, which have been represented, are removed after parameter updating. The training process goes on. A coming sample x^k causes one of the following actions.

1. Update the parameters of some hidden neuron, and remove c;
2. Create a new hidden neuron to represent it, and remove x^k;
3. Back up x^k to be learned in the next training circle.

Given a hidden neuron j with the center c^j and three radii r_1^j, r_2^j and r_3^j, we firstly compute the function for the hidden neuron j defined as:

$$f(w^j, x^k) = \sum_{i=1}^{n} w_i^j x_i^k \tag{3}$$

Where w^j is ($w_1^j, w_2^j, ..., w_n^j$), the weight vector and x^k is the kth vertex. The training process is continued as follows:

I. If $f(w^j, x^k) \geq t_1^j$, already covered, so nothing needs to be done.

II. If $t_1^j > f(w^j, x^k) \geq t_2^j$ the sample x is within the "claim region"; so to include it an immediate expansion is needed.

III. If $f(w^j, x^k) \geq t_3^j$ the sample is confusing sample so back up x^k to be dealt with in the next training circle.

IV. If for all j's, $f(w^j, x^k) < t_3^j$ then create a new hidden neuron and remove the sample x^k from the training set.

Thus the number of neurons generated is equal to the number of clusters. After this, the labeled samples are useful for labeling the clusters. The details are given next.

3.2 Labeling Process

In labeling process labels are assigned to hyperspheres formed after the clustering process by using the mechanism of Majority voting concept. Thus these labeled hypersphere can be represented as output neuron in the output layer of network architecture. After clustering when hyperspheres are identified, we assign labels to hyperspheres.

1. Repeat the step 2 and step 3 for each of the hyper sphere.
2. Perform majority voting by count number of samples belongs to one particular class.

3. Majority of samples of particular class in an individual hypersphere would decide the class of that hypersphere.
4. If a particular hypersphere is not covering any labeled data in that case merge this hypersphere with other which is closure to it.

4 Experimental Work

We used a Personal Computer (PC) with Pentium processor with 2.99 GHz speed and 1GB of RAM having windows XP operating system for testing. We used Matlab 7.0.1 for implementation.

Table 1.

Dataset	Dimensions, Number of classes	Training	Testing
Fisher's Iris	4, 3	123	27
Breast Cancer	9,2	683	68
Balance Scale	4,3	492	76
Riply	2,2	720	80

Each training samples $x^k = (x_1^k, x_2^k, ..., x_n^k)$ are normalized as follows:

$$x_i^k = x_i^k - \min(x_i) / \max(x_i) - \min(x_i) \tag{4}$$

After this transformation, each data sample is transformed in the range 0 and 1. CS-SCA requires binary form of input data therefore after normalization re-quantizes the data into eight levels as follows.

1. Apply each sample as an input in quantized function given in step 3.
2. Quantized value can be obtained by: $y = uencode\ (u, n, v)$
3. Repeat step 3, till the whole sample binary coded

After data preparation, for experimentation 80% of the original data taken as training data and rest 20% considered as testing samples. The datasets used for experimentation are given in table 1. Results are evaluated in terms of classification accuracy, training time, confusing samples and number of hyperspheres required. For different value of training parameter results for each dataset are getting change. After calculating the performance of CS-SCA, same datasets are applied in SVM based classifier [6], to compare the performance of both the classifiers, in terms of Classification accuracy, Training time. In SVM based classifier, training parameter α used in clustering process. Number of support vector in SVM based classifier depends on the value of training parameter α . Comparison results of both the classifier are displayed in Table3.

Table 2. For 10-fold cross validation results

Dataset	Average Accuracy
Wisconsin Beast Cancer	85.1 %
Riply	80.1 %

Table 3. Comparison with SVM

Dataset	Accuracy by CS-SCA	Accuracy in SVM	Training Time in CS-SCA	Training Time in SVM
Fisher's Iris	92.59 %	77 %	0.96 sec.	5.89 sec.
Balance Scale	80.26 %	77 %	2.18 sec.	65.9 sec.
Wisconsin breast cancer	85 %	70 %	4.29 sec.	1086 sec.
Riply	80.1 %	75 %	3.98 sec.	36.12 sec.

We give results for 10-fold cross validation on Wisconsin Breast Cancer and Riply dataset in table2 shown above. For 10-fold cross validation 90% of the data taken as training and rest 10% taken as testing data. From the results shown in above table3, it's clear that for each dataset CS-SCA is giving better accuracy and requires less training time compare to SVM based classifier.

5 Concluding Remarks

A binary neural network based Semi-supervised classifier is constructed using the concept of geometrical expansion, which classify semi-labeled data. The classification is performed using two processes, first is clustering and second is labeling. Various benchmark datasets used to demonstrate the performance of CS-SCA in terms of accuracy and number of hypersphere etc. After that same datasets is applied in SVM based classifier to compare its performance with developed classifier. It's found that CS-SCA gives better performance in terms of accuracy, training time etc.

References

1. Wang, D., Chaudhari, N.S.: A Constructive Unsupervised Learning Algorithm for Clustering Binary Patterns. In: Proceedings of International Joint Conference on Neural Networks (IJCNN 2004), Budapest, July 2004, vol. 2, pp. 1381–1386 (2004)
2. Wang, D., Chaudhari, N.S.: A Novel Training Algorithm for Boolean Neural Networks Based on Multi-Level Geometrical Expansion. Neurocomputing 57C, 455–461 (2004)
3. Kim, J.H., Park, S.K.: The geometrical learning of binary neural neworks. IEEE Transaction. Neural Networks 6, 237–247 (1995)
4. Joo Er, M., Wu, S., Yang, G.: Dynamic Fuzzy Neural Networks. McGraw-Hill, New York (2003)
5. Kwok, T.Y., Yeung, D.Y.: Constructive algorithms for structure learning in feedforward neural networks for regression problems. IEEE Trans. Neural Networks 8, 630–645 (1997)
6. Chaudhari, N.S., Tiwari, A., Thomus, J.: Performance Evaluation of SVM Based Semi-supervised Classification Algorithm. In: International Conference on Control, Automation, Robotics and Vision, Hanoi, Vietman, December 17-20 (2008)
7. `http://www.ics.uci.edu/kmlearn/MLRepository.html`