Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong, 30-31 Aug. 2008

AN IMPROVED ALGORITHM FOR INCREMENTAL LEARNING LEARN++

AI-JUN LI

Shanxi University of Finance & Economics , Taiyuan, Shanxi, 030006, China
E-MAIL: Liaijun12@tom.com

Abstract:

An Improved Algorithm for Incremental Learning LEARN++ is proposed. Based on the LEARN++ algorithm, The Improved Algorithm modifies instances selection technique in Boosting and provides an effective method of selecting the individual networks. The Algorithm allows the new classifiers trained with new database to learn past knowledge reducing the forgetfulness , and ensures generalization performance by selecting individual networks. The experiments show that the performance of Improved Algorithm is better than that of LEARN++ and its memory capacity and computing cost. are acceptable.

Keywords:

Incremental learning; Boosting; LEARN++

1. Introduction

LEARN++[1] is inspired by active AdaBoost algorithm[2,3]. AdaBoost iteratively updates the distribution by assigning appropriate weights to each instance such that the weak learner, which is trained with a subset training data drawn from this distribution, is forced to focus on increasingly harder instances. Thus the weak leaner is challenged to learn the difficult parts of the instance space. Considering an ensemble neural network that has been trained with an initial dataset that come from a distribution $\mathcal{D}_1 \subseteq \mathcal{D}$ for which an initial set of hypotheses is generated, where $\mathcal{D}$ is the original distribution. Now suppose that we are given a new dataset, drawn from a distribution $\mathcal{D}_2$. If the original training dataset is a good representative of the entire instance space $\mathcal{D}$, then the ensemble network will perform well on the new dataset. If the ensemble network does not perform well on the new dataset, then the first dataset is not a good representative of the entire instance space $\mathcal{D}$, and $\mathcal{D}_2 \not\subset \mathcal{D}_1$. In other words, the ensemble network is not trained with that part of the instance space from which the new data are drawn. Therefore, we can interpret these instances as hard examples of $\mathcal{D}$, and force the ensemble network to learn instances coming from $\mathcal{D}_2$ by generating new hypotheses for this dataset and combine all hypotheses generated for classification of unknown instances.

In LEARN++, let the distribution of instances to be learned is $\mathcal{D}$ including $\mathcal{D}_k, k = 1,2,\cdots,K$ .where , instances from $\mathcal{D}_1$ is initial dataset to train ensemble network , and instances from $\mathcal{D}_k, k = 2,3,\cdots,K$,are new datasets to be learned incrementally . All hypotheses are then combined by weighted majority voting. LEARN++ updates the AdaBoost in two key aspects. First aspect is distribution update. For each iteration t during current dataset $\mathcal{D}_k$, training (TR_t) and testing (TE_t) subsets are randomly generated from the current dataset. These subsets are provided to individual, which returns the hypothesis h_t .Unlike AdaBoost, the error, ε_t ,is computed from the misclassified patterns of $TR_t + TE_t$. If $\varepsilon_t > 1/2$， h_t is discarded, and new TR_t and TE_t are generated. Then distribution update is performed by the overall hypothesis H_t and its scaled error B_t .

When learn dataset from $\mathcal{D}_k$, the update rule of distribution is as followed

Let $B_t = E_t/(1-E_t)$， update distribution D_t：

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} B_t, & while\ H_t(x_i) = y_i \\ 1, & otherwise \end{cases} \quad (1)$$

where E_t is overall error generated by the t th individual network， D_t is the sample distribution of the t th individual， $Z_t = \sum_i D_t(i)$ is a normalization constant to ensure that D_{t+1} is a distribution.

The second key update is the method of combining hypotheses generated by individuals. A final hypothesis of LEARN++ can be obtained by combining all hypotheses that have been generated by using the weighted majority voting rule. The final output is formulated like this:

978-1-4244-2239-5/08/$25.00 ©2008 IEEE

$$h_{final}(x) = \arg\max_{y \in Y} \sum_{k} \sum_{h_t(x)=y} \log(1/\beta_t) \quad (2)$$

where β_t is individual error.

Without using old samples and changing previous structure, LEARN++ is able to learn incrementally and retain previous knowledge. But it can be improved in some aspects. On the basis of this method, we proposed an Improved algorithm for feed-forward Neural Network to solve the recognition rate decreasing problem caused by neural network ensemble which unable to learn with single-class or small instances set. The improved algorithm ensures generalization by the number selection of individual networks. The experiments show that the performance of Improved Algorithm is better than that of LEARN++ and its memory capacity and computing cost. are acceptable.

2. The Improved Algorithm

LEARN++ can be improved in some aspects. We improve LEARN++ aiming at two problems. One is that neural network does not learn single-class and small instances, and the other is the selection of individuals.

2.1. Small sample and single-class learning

A neural network must be PAC-learnability that is determined by the number of training instances. Haussler [4] states the upper bound of the number of training instances, and Blumer[5] states the lower bound of the number of training instances with Vapnik-Chervonenkis dimension. Based on positive and negative instances with different classes, the learning of neural network must have enough instances. The conclusion stated by Blume shows that if the number of instances is less, neural network can not be PAC-learnability. This means the error will be more than 0.5. Furthermore, if instances to be learned are in the same class, even if the number of instances is big enough, the neural network is still not PAC-learnability. Experiments can also verify this conclusion. The simulation result on Iris classification shows that neural network is convergence when it is trained with single-class instances. But it recognizes instances in other classes all as the trained class, and so is ensemble network. Learning on single-class instances needs negative instances to obtain an effective learner. Learning on small dataset and single-class instances has great effect on the performance of incremental learning. The experiment result of Learn++ also shows that the accuracy rate of new classes is obviously lower than that of previous classes and the accuracy of previous classes decreases after adding the new individual networks.

We exploit appending appreciate instances into small dataset or single-class dataset such that all individual networks are PAC-learnability and need less memory space and computational cost. For a classification problem, neural network try to find the best boundaries distinguishing different classes. If some negative instances can be combined into new dataset, the representation regions of new instances can be formed. In pattern classification, those misclassified instances are generally distributed on side of boundaries. If multi-classes are adjacent to each other, as one of boundary changes, the instances beside of boundary may be divided into different regions as different class. Now we consider Boosting algorithm. Because new individual network is forced to learn misclassified instances and forms the dataset for generating new individual network by judging the performance of previous networks, the misclassified instances are just filtered to consist a negative set S_{neg} . With negative set and new set $S_{neg} + S_{new}$, the performance of incremental learning is improved. Then, the procedure of Boosting creating new dataset is used to generate the negative instances. The process of generating negative instances is embedded in the learning step of Boosting, the computational cost can almost be ignored. Meanwhile, the memory space needed is very limited because the number of instances beside the boundary is far less than that of all instances. The Improved Algorithm learns hard learning instances to improve the generalization of incremental learning system.

2.2. Selection of individuals

The generalization of ensemble networks is determined by diversity of individuals. The more the diversity is, the better the generalization is. Although the individuals generated by Boosting have some diversity, it becomes more and more difficult to ensure the diversity of new individuals with more and more individuals. New individuals are selected by comparing the new individuals with previous individuals to improve the performance of incremental learning system.

In practice, the distribution estimation of test set is often used to calculate the diversity. The typical method is Kappa statistic. κ-tatistic is defined as

$$\kappa = \frac{\Theta_1 - \Theta_2}{1 - \Theta_2} \quad (3)$$

Where Θ_1 is the consistent probability estimation of

two individuals, Θ_2 is occasional consistent probability estimation of two individuals. κ=0 means that occasional coherence is occasional expect; κ=1 means that two individuals classify each instance into the same class, which means there is no diversity between two individuals. κ<0 means that occasional coherence is less than occasional expect, which means there is diversity between two individuals.

In learning process, the new individual must be compared to existing individuals, and if κ<0, the new individual will be hold.

Opitz[3] shown that the number of individuals generated by Boosting are not the more the better . The performance improvement is made in first individuals. the performance of incremental learning system will be improved by controlling the number of individuals. The main idea of controlling the number of individuals in the incremental learning algorithm is to set a error threshold α ,and stop the individual generation process when the error of ensemble network is less than α .

2.3. An Improved Algorithm

The steps of the Improved Algorithm are as followed

Input: For each dataset drawn from $D_k, k = 2,3,\cdots,K$

·Sequence of m instances $S = [(x_1, y_1),(x_2, y_2),\cdots,(x_m, y_m)]$

·individual network learning algorithm（for example MLP）

·Integer T_t specifying the number of the individuals

· D_{novel} specifying the set for saving instances beside boundary ， initial value is empty. **

Do for $k=1,2,...,K$:

$D_t(i) = D_t(i) \cup D_{novle}$ **

initialize $D_t(i) = 1/m,\ \forall i$

Do while $t<=T_k$ and $E_t > \alpha,\ \alpha > 0$

1． Randomly choose training TR_t and testing TE_t subsets from D_t

2． Call individual network learning algorithm， providing it with TR_t from D_t

3．Generate a hypothesis $h_t : \mathbf{X} \to \mathbf{Y}$，and calculate the error of $h: \varepsilon_t = \sum_{t:h_t(x_i)\neq y_i} D_t(i)$ on $TR_t + TE_t$. If $\varepsilon_t > 1/2$， set $T = t-1$ ， discard h_t and go to step 1.otherwise, call diversity degree algorithm and compare it with first T individuals，if $k_i < 0,\ \forall i,\ i = 1,2,\cdots,T-1$，hold the new individual. calculate scaled error as $\beta_t = \varepsilon_t/(1-\varepsilon_t)$. **

4. Call weighted majority algorithm, obtain the over all hypothesis:

$H_t = \arg\max_{y\in Y} \sum_{t:h_t(x)=y} \log\frac{1}{\beta_t}$ and calculate the overall error:

$error\ \ E_t = \sum_{t:H_t(x_i)\neq y_i} D_t(i)$

if $E_t < \alpha,\ \alpha > 0$,stop，go to 6 **

5．set $B_t = E_t/(1-E_t)$，and update distribution D_t：

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} B_t, & while\ H_t(x_i) = y_i \\ 1, & otherwise \end{cases}$$

where $Z_t = \sum_i D_t(i)$ is a normalization constant such that D_{t+1} will be a distribution，and save the instances which weight value is 1 to D_{novel}，discard the redundancy **

END while

6．Call weighted majority and output the final hypothesis：

$$h_{final}(x) = \arg\max_{y\in Y} \sum_k \sum_{h_t(x)=y} \log(1/\beta_t)$$

END for

The symbol ** indicates the major modifications made to LEARN++ given in[1]. We can obtain that the computational cost of modification is very small by steps of the Improved Algorithm.

3. Experiment results

The Improved Algorithm was tested on a variety of datasets. In order to compare with LEARN++, two experiments are presented here. One experiment employs synthetic dataset distributing in circular regions, the data have two dimension and in five classes. The experiment employs Glass dataset. Each experiment shows the generalization performance of the Improved Algorithm and LEARN++.

3.1. Synthetic dataset in circular regions

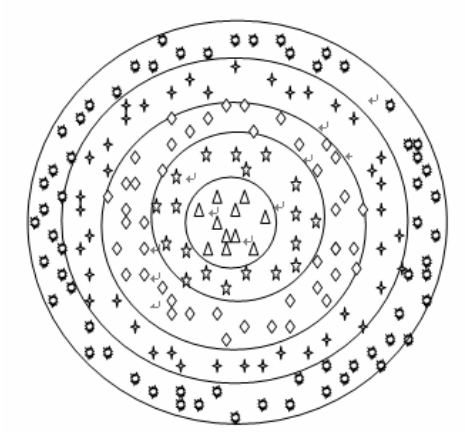

Figure 1 dataset in circular regions

Synthetic dataset in circular regions consists of concentric circles. Innermost circle is labeled as class 1 and outermost is labeled as class 5. Six training datasets, $S_1 \sim S_6$, are generated from this dataset, where S_1 and S_2 has instances from classes 1,3,and 5, S_3 S_4 has instances from class 4, S_5 and S_6 has instances from classes 2. An additional dataset, TEST, is generated from all classes to test performance of incremental learning system. These datasets are presented to the Improved Algorithm and Learn++ algorithms starting with S_1, generate a series of individuals to form a ensemble network on present dataset. The ensemble network is tested on training set and testing data. Table 1 and 2 present the experiment results.

Table 1 Performance of LEARN++ on synthetic data

Set	TS1	TS2	TS3	TS4	TS5	TS6
S_1	100%	99.7%	97.7%	96.7%	94.3%	92.7%
S_2		96.3%	93.7%	92.3%	90.0%	88.3%
S_3			97.5%	97.0%	94.5%	93.5%
S_4				93.5%	91.5%	90.0%
S_5					89.0%	87.5%
S_6						89.0%
TEST	56.0%	59.2%	72.0%	74.8%	81.2%	88.4%

Each column represents the accuracy rate of final ensemble network obtained by adding a new ensemble network in all current and previous training sessions. Each row shows the accuracy rate of final ensemble network on a particular dataset. The results in tables show that the accuracy rate increases gradually with the increase of the number of ensemble network. Meanwhile, accuracy rates of the Improved Algorithm and LEARN++ are both decreasing on particular dataset with the ensemble networks increasing. But accuracy of the Improved Algorithm is obviously better than that of LEARN++ on TEST dataset.

Table 2 Performance of the Improved Algorithm on synthetic data

Set	TS1	TS2	TS3	TS4	TS5	TS6
S_1	99.7%	99.7%	99.3%	99.0%	98.7%	98.7%
S_2		99.7%	99.7%	99.3%	99. %	98.3%
S_3			89.5%	92.0%	93.5%	95.5%
S_4				93.5%	95.5%	97.0%
S_5					88.5%	91.5%
S_6						89. %
TEST	55.6%	59.2%	73.2%	76.4%	88.8%	94.4.0%

The performance of the Improved Algorithm is better than that of LEARN++. LEARN++ learning on single-class data set generates corresponding ensemble networks that can misclassify instances from dataset of other classes. During training, S_3 S_4 only includes the fourth-class instances, and S_5 S_6 only includes the second-class instances. So the final error increases by misclassification. The Improved Algorithm utilizes instances beside boundary of the previous datasets when leanring the new training dataset and recognizes most instances in previous datasets accurately. Therefore, the performance of new ensemble network will not be decreased heavily for previous instances.

The performance on TEST dataset increases gradually because TEST dataset which includes instances from all classes was never learned during any training session. After training on S_1 S_2 which include only instances from the first-class, third-class and fifth-class datasets, the learner misclassifies the instances from the second-class and fourth-class datasets. Then it recognizes accurately the

Table 3 Distribution of GLASS dataset

B-W-F-P	B-W-N-F-P	V-W-F-P	V-W-N-F-P	Containers	Tableware	Headlamps
70	76	17	0	13	9	29

instances on TEST datasets from the first-class, third-class and fifth-class datasets. When training on S_3 S_4, the learner recognizes accurately the instances from the fourth-class dataset . after training on S_5和S_6, the learner recognizes accurately the instances from the second-class dataset .

3.2. GLASS dataset

There are seven classes and 214 instances in GLASS dataset with 9 attributes. Table 3 shows its distribution. In experiment, the dataset is divided into training set S and testing set *TEST* . Then the instances from classes B-W-N-F-P、V-W-F-P、Tableware and a part of instances from class B-W-F-P form the dataset S_1, $S_1 \in D_1$, residual instances from class B-W-F-P is the first dataset S_2 for incremental learning $S_2 \in D_2$, and instances from classes Containers and Headlamps is the second dataset S_3 for incremental learning $S_3 \in D_3$. Table 4 and 5 present the experimental results.

Table 4 The performance of LEARN+

Set	TS1	TS2	TS3
S1	94.6	95.7	93.5
S2		91.1	88.9
S3			83.7
TEST	46.5	67.4	76.7

Table 4 shows that the performance on GLASS set is similar to that on synthetic data. In LEARN++, the accurate rate on TEST set increases with ensemble networks increasing. But the accurate rate on previous learned dataset decreases. Table 5 presents that the accurate rate on previous dataset is improved by repeatedly learning the instances which are easy to be misclassified and the generalization performance of incremental learning system is also improved.

Table 5 The performance of the Improved Algorithm.

Set	TS1	TS2	TS3
S1	94.6	91.4	86.0
S2		91.1	86.7
S3			83.7
TEST	48.8	60.5	69.8

4. Conclusions

On the basis of LEARN++ , an Improved algorithm for feed-forward Neural Network is proposed. The Improved Algorithm can incrementally learn the new knowledge and hold the previous knowledge structure at the same time. It solves the problem that neural network is unable to learning on single-class or small samples set, and it can select the individuals effectively. The generalization performance of the Improved Algorithm is better than that of LEARN++, and its memory space and computing cost are both acceptable. It has universality which means not only can it be used for neural network but also for other learners.

Acknowledgements

This research is supported by the Shanxi Natural Science Foundation of China.

References

[1] Polikar, R.， Udpa,L., Udpa,S.,& Honavar, V. learn++ : an incremental learning algorithm for multilayer perceptron networks[C]. Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Proc, 2000,6: 3414-3417.

[2] Freund,Y., Schapire, R.E. A Decision Tree-Theoretic Generalization of On-Line Learning and an Application to Boosting[J]. Journal of Computer and Science, 1997, 55(1):119-139.

[3] Opitz, D., Maclin, R. Popular ensemble methods: An empirical study[J]. Journal of Artificial Intelligence Research, 1999, 11: 169- 198.

[4] Haussler, D. Quantifying inductive bias :AI learning algorithm and Valiant's learning framework [J]. Artificial Intelligence, 1988,36: 177-221.

[5] Blumer, A., Ehrenfeucht, A. Haussler, D., &Warmuth, M.K. Learnability and the Vapnik-Chervonenkis Dimension[J]. Journal of the Association for Computing Machinery, 1989, 36:929-965.