An Empirical Comparison between SVMs and ANNs for Speech Recognition

Jixin Li JIXLI@PAUL.RUTGERS.EDU
Department of Computer Science, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ 08854 USA

Abstract

While Artificial Neural Networks (ANNs) are widely used, Support Vector Machines (SVMs) are a comparatively new and efficient pattern recognition tool. In this paper, we examined the performance of SVMs and ANNs with different architectural and parameter settings for both binary and multi-class classification problems based on a vowel speech dataset. We adopted the commonly used one-against-all and all-against-all method for SVM. Our results show that both approaches had similar performance in binary classification but SVM outperformed ANN greatly in multi-class classification problems.

1. Introduction

1.1 Support Vector Machine

Support Vector Machines (Vapnik, 1995; Burge, 1998) are a comparatively new approach to the problems of classification, regression, ranking, etc. As a binary classifier, it tries to find an optimal hyperplane that maximizes the margin between data samples in two classes in a higher dimensional feature space derived from the original data space through a kernel function, while reducing the training errors.

In principle, the only parameter used in SVM, besides the kernel function, is a parameter *C*, which determines the trade-off between two conflicting goals: maximizing the margin and minimizing the training errors.

1.2 Artificial Neural Network

ANNs consist of multilayered perceptrons, connected with sets of weights. The Back-propagation algorithm is usually used to set the weights to minimize the squared error cost function over a training set.

1.3 Pros and Cons

Neither ANNs nor SVMs are perfect. SVMs are fast in training and guarantee a global optimum if the kernel satisfies Mercer's condition (Vapnik, 1995), but requires an appropriate choice of kernel function. ANNs are slow in training and can only guarantee local optima, but are fast in classifying and robust to noise.

A number of comparisons between SVMs and ANNs have been undertaken since the introduction of SVMs. For instance, both have been used in multi-class protein fold recognition (Ding & Dubchak, 2001), where SVMs were found to be more effective than ANNs. Another example is intrusion detection (Mukkamala, Janoski & Sung, 2002), where both learning methods delivered great accuracy with SVMs showing slightly better performance. Shawkat and Abraham (2002) did an empirical comparison on six databases and showed that SVMs were much faster than ANNs and in general had a better performance than ANNs.

In this paper, the performance of SVMs and ANNs were compared on a vowel speech data set, with various parameter settings for each learning methods. The experiments show that both have similar performance in binary classification, but SVM outperformed ANN in multi-class classification.

2. Experiments

2.1 Data Set

2.1.1 SOURCE OF DATA

The comparison was done on vowel recognition data collected by Deterding (1989). It is available from the UCI Depository (Blake & Merz, 1998)

Eleven words were spoken in British English by each of 15 speakers 6 times. Each word contains one distinct vowel. Voices were recorded and turned into 10-input vectors. This yielded 11x15x6=990 examples in 10-dimensional data space. Data from 4 male speakers and 4 female speakers were used as training data, and the other 4 male speakers and 3 female speaks as test data, resulting in 528 training examples and 462 test examples.

The eleven vowels are represented by the following words: *heed, hid, head, had, hard, hud, hod, hoard, hood, who'd, heard*.

2.1.2 PREPROCESSING

In most of our experiments, I used the data provided as they were, since these data were already well processed and maintained. However, I did normalize the data by row in some of the experiments so that each data is located in a hypersphere of radius 1 and the normalization did show an impact on the performance of the classifiers. The results of this modification are presented in later sections.

2.2 Design of Experiments

A binary classification problem and a multi-class classification problem were designed to compare the performance of SVM and ANN. In the experiments, only the accuracy on the test data was evaluated. We used the linear kernel $k(x, y) = x \cdot y$ and Gaussian kernel $k(x, y) = e^{-\gamma \|x-y\|^2}$ with different values of γ for SVM. The trade-off parameter *C* is selected by the default of the SVM software.

2.1.3 BINARY CLASSIFICATION

A binary classification problem was formed to evaluate the classification performance of the SVMs and ANNs. Classes were formed based on the gender of the speakers. In other words, male speakers were in one class, and the female speakers were in the other. The experiment was done with both the original data and the normalized data

We used three-layered ANNs, with 10 nodes in the input layer and 1 output node. The number of nodes in the hidden layer is 3, 5 or 11.

2.1.4 MULTI-CLASS CLASSIFICATION

Classes were created based on 11 vowels. One-against-all and all-against-all methods are used for classification. The experiment was done without data normalization.

2.1.4.1 ONE-AGAINST-ALL

In the one-against-all methods (Scholkopf & Smola 2002), for SVM, to classify M classes, M classifiers are trained to separate one class from the rest. The classification is made by taking the maximum of the real-valued output of the hyperplane boundary functions of each classifier. This method is quite heuristic since these values are outputs of different classifiers' discriminant function, which are not directly relevant to each other. There are some methods that transform the real-valued outputs into probabilities, which then can be compared to choose the class. But in this experiment, I used the heuristic method.

Though ANN does not need one-against-all method for multi-class classification, this method was also used for ANN in this experiment for comparison purpose.

One of the problems of the one-against-all method is the imbalance of the data. Ding and Dubchak (2001) used duplication to make the numbers of examples of two classes approximately match. Particularly for the case of SVM, LibSVM (Hsu, Chang & Lin 2003) uses separate trade-off variables C_- and C_+ for training errors of the two classes to deal with imbalanced data. In my experiment I did not adopt any of the methods, since the outcome remains unknown to me.

2.1.4.2 ALL-AGAINST-ALL

For M classes, classifiers are trained for each pair of classes, which results in M(M-1)/2 classifiers. The test data then go through each of the classifiers and are assigned to the class with the largest number of votes. In case of a draw, there are many methods available to deal with this situation. In my experiment, the class with the smaller index was chosen, which is also used in LibSVM.

2.1.4.3 11-OUPUT CODING OF ANN

Though one-against-all method was used, an 11-output ANN was also designed. Each output corresponds to one class and the data from the ith class correspond to the output configuration that the ith output is 1 and the other outputs are 0. In the test phase, the output with the largest value was considered and the corresponding class was assigned.

2.2 Software

SVM^{light} 4.0 (Joachims, 1999) was used for SVM and NevProp 1.16 (Fahlman, 1988) for ANN.

3. Results and Analysis

3.1 Binary Classification

Table 1 shows the results of binary classification by SVM with a linear kernel, SVM with Gaussian kernel ($\gamma = 1$ [1]) and ANN with 3, 5 and 11 nodes in the hidden layer. As we can see from the table, an ANN with a hidden layer of 5 nodes achieved the best test accuracy of 80.1%. An SVM with a Gaussian kernel ranked second with an accuracy of 74.7%

Table 2 shows the results of binary classification with normalized data. $\gamma = 20$ is chosen for an SVM with a Gaussian kernel. In this case the accuracy of the SVM with Gaussian kernel rose to 81.7%.

In all, SVM and ANN showed similar performance, with a best accuracy of around 80%. Most of the classifiers showed accuracies higher than 70% as apposed to 4/7=57.1% using the strategy of always guessing "male".

[1] The value of gamma was selected from several candidates, for instance, 0.5, 3, 20, etc. and the one that resulted in the best performance was chosen. This principle also applies to other situations wherever a Gaussian kernel was used in our experiments

Table 1. Binary classification results on test data. M-M: male and classified as male; M-F: male but classified as female; F-F: female and classified as female; F-M: female and classified as male.

	SVM LINEAR	SVM GAUSSIAN	ANN-3	ANN-5	ANN-11
M-M	166	151	157	190	167
M-F	98	113	107	74	97
F-F	171	194	126	180	171
F-M	27	4	72	18	27
ACCURACY (%)	72.9	74.7	61.3	80.1	73.2

Table 2. Binary classification results with normalized data.

	SVM LINEAR	SVM GAUSSIAN	ANN-3	ANN-5	ANN-11
M-M	170	184	149	178	153
M-F	94	80	115	86	111
F-F	174	192	179	177	183
F-M	24	6	19	21	15
ACCURACY (%)	74.5	81.7	71.0	76.8	72.7

3.2 Multi-class Classification

Table 3 and Table 4 show the classification accuracy for each class of the 11 vowels and the overall accuracy. The shaded numbers are used to stress the best accuracy for a class.

As shown in the tables, SVM with Gaussian Kernels outperformed ANN, and ANN outperformed SVM with linear kernel.

It is also shown that the all-against-all method improved the performance of SVM compared to one-against-all, but not much. For ANN, there is no significant difference between the one-against-all method and a simple 11-output method.

The classification results should provide some feedback to the data set. If the results of a classifier reflect better the nature of the data set, it would be potentially more useful and powerful. Table 5 and 6 shows matrices of the classification results of an SVM with a Gaussian kernel using the one-against-all method and the classification results of ANNs with 3 nodes of hidden layer and 11 output nodesr not using the one-against-all method. In both tables, the ith row and the jth column is the number of samples that belong to Class i but classified as Class j. Significantly large classification errors (larger than 15) are emphasized with shaded numbers.

Table 3. Multi-class classification results, using one-against-all methods for both SVMs and ANNs. $\gamma = 1$ for the Gaussian kernel

ACCURACY (%)	SVM LINEAR	SVM GAUSSIAN	ANN-5	ANN-11
1: HEED	85.7	47.6	71.4	71.4
2: HID	4.8	78.6	42.9	45.2
3:HEAD	14.3	73.8	35.7	66.7
4:HAD	85.7	81.0	47.6	64.3
5:HARD	31.0	38.1	21.4	35.7
6:HUD	7.1	66.7	28.6	45.2
7:HOD	16.7	71.4	12.0	42.9
8:HOARD	66.7	83.3	50.0	40.5
9:HOOD	33.3	40.5	76.2	47.6
10:WHO'D	54.8	50.0	33.3	26.2
11:HEARD	40.5	76.2	9.5	21.4
ALL	40.0	64.3	39.0	46.1

Table 4. Multi-class classification results cont'd, using all-against-all methods for SVM and 11-output configuration for ANN. $\gamma = 0.2$ for Gaussian kernel

ACCURACY (%)	SVM LINEAR	SVM GAUSSIAN	ANN-3	ANN-5	ANN-11
1: HEED	76.2	59.5	71.4	45.2	71.4
2: HID	21.4	69.1	76.2	45.2	69.1
3:HEAD	33.3	61.9	59.5	42.9	64.3
4:HAD	61.9	81.0	50.0	42.9	54.8
5:HARD	50.0	47.6	26.2	31.0	26.2
6:HUD	40.5	71.4	4.8	57.1	35.7
7:HOD	26.2	66.7	42.9	50.0	11.9
8:HOARD	61.9	88.1	81.0	47.6	33.3
9:HOOD	31.0	50.0	31.0	4.76	40.5
10:WHO'D	38.1	50.0	23.8	42.9	54.8
11:HEARD	45.24	76.2	83.3	14.3	78.6
ALL	44.2	65.6	50.0	38.5	49.1

For the SVM case, the biggest error occurs between class pairs (1, 2), (2, 3) and (5, 6), which implies that vowels in pairs (heed, hid), (hid, head) and (hard, hud) are very similar to each other and hard to classify, which is very reasonable for British English. While for ANNs, the biggest error occurs between class pairs (2, 3), (6, 11) and (1, 10) implying that vowels in the pairs (hid, head), (hud heard) and (heed, who'd) would be very similar. But as apposed to the expectation, I found that hud and heard are not close to each other and neither are heed and who'd. In this sense, SVM with Gaussian kernel better modeled confusability.

Table 5. Multi-class classification results of SVM with Gaussian kernel using one-against-all method. $\gamma = 1$.

#	1	2	3	4	5	6	7	8	9	10	11
1	20	22	0	0	0	0	0	0	0	0	0
2	33	6	0	0	0	0	0	0	0	0	0
3	3	31	7	0	0	0	0	0	0	0	1
4	0	0	1	34	0	7	0	0	0	0	0
5	7	0	0	0	16	15	4	0	0	0	0
6	0	0	0	0	3	28	1	0	0	0	10
7	7	0	0	0	5	0	30	0	0	0	0
8	0	0	0	0	0	0	6	35	1	0	0
9	0	0	0	0	0	0	9	6	17	4	6
10	10	5	0	0	0	0	0	2	4	21	0
11	0	0	0	0	0	2	1	0	7	0	32

Table 6. Multi-class classification results from ANN with 3 nodes of hidden layer and 11 nodes of output layer.

#	1	2	3	4	5	6	7	8	9	10	11
1	32	12	0	0	0	0	0	0	0	0	0
2	8	32	0	0	0	0	0	0	0	0	2
3	1	25	6	0	5	0	0	0	0	0	5
4	0	0	2	21	0	6	0	0	0	0	13
5	0	0	0	2	11	2	14	0	0	0	13
6	0	0	1	0	4	2	0	0	1	0	34
7	0	1	1	0	1	1	18	8	3	0	9
8	0	0	0	0	0	0	7	34	0	1	0
9	0	2	0	0	0	0	0	13	13	9	5
10	21	0	0	0	0	0	0	8	1	10	2
11	0	0	0	0	0	0	0	0	6	1	35

4. Conclusions

Overall, SVM and ANN showed similar results in binary classification, but SVM with Gaussian kernel outperformed ANN greatly in multi-class classification. Also the multi-class classification results of SVM better reflected the nature of the data.

Acknowledgements

I would like to thank Dr. Littman and Yihua Wu for their careful reading and insightful advises. I would also like to thank in advance my reviewers for their precious time and helpful comments.

References

Blake, C. and Merz, C.J. (1998). UCI Repository of Machine Learning Database. http://www.ics.uci.edu/ mlearn/MLRepository.html, Ivrine, CA: University of California

Burges, C.J.C. (1998). A tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2, 1-43.

D. H. Deterding. (1989) University of Cambridge, "Speaker Normalisation for Automatic Speech Recognition", submitted for PhD

Ding, C. and Dubchak I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics* 2001, 17:349-358

Fahlman, P. (1988). *Faster-Learning Variations on Back-Propagation: An Empirical Study*, In proceeding of the 1988 Connectionist Models Summer school. Morgan Kaufmann, 1998

Hsu, C.-W., Chang, C.-C, Lin, C.-J. (2003). *A practical guide to support vector classification*

Mukkamala S., Janoski G., Sung A. H. (2002). *Intrusion Detection Using Neural Networks and Support Vector Machines*," Proceedings of IEEE International Joint Conference on Neural Networks, pp. 1702-1707.

Scholkopf, B., Smola, A. J. (2002) *Learning with Kernels*. MIT Express, 2002 pp. 211-212.

Shawkat Ali, Ajith Abraham, (2002). *An Empirical Comparison of Kernel Selection for Support Vector Machines*, 2nd International Conference on Hybrid Intelligent Systems, Soft Computing systems: Design, Management and Applications, IOS Press, The Netherlands, pp. 321-330, 2002.

T. Joachims. (1999). *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

V. Vapnik. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.