

Comparison of Scaling Behavior Between Fuzzy c -Means Based Classifier with Many Parameters and LibSVM

Hidetomo Ichihashi, Katsuhiro Honda, Akira Notsu
Department of Computer Science and Intelligent Systems
Osaka Prefecture University
1-1 Gakuen-cho, Nakaku, Sakai, Osaka 599-8531, Japan
Email: {ichi, honda, notsu}@cs.osakafu-u.ac.jp

Abstract—This paper reports the scaling behavior of the fuzzy c -means based classifier (FCMC) with many parameters. FCMC is a classifier based on clustering approaches. The classification accuracy on test sets (i.e., the generalization capability) is not necessarily improved by increasing the number of clusters. Especially when the number of training samples is relatively small, not only the classification boundary over-fits the data, but also covariance matrices and cluster centers are computed incorrectly, since the number of samples in each cluster becomes smaller. Hence, the test set accuracy deteriorates.

The performance of FCMC with two clusters in each class and the number of training samples less than 1000, was reported in the literature. This paper reports the scaling behavior of FCMC by testing with variously-sized training samples. The number of clusters of FCMC is increased up to eight. The number of clusters used in this paper is not very large but the number of parameters is relatively large. So, the parameters are optimized to training sets.

LibSVM is one of the widely known state of the art tools for support vector machines (SVM). The test set accuracy, training time and testing time (i.e., the detection time) of FCMC are compared with LibSVM by varying the size of training sets. FCMC shows a good generalization capability, though the parameters are optimized to training sets. When the number of training samples is increased by 10 times, the training time of FCMC increases by 10 times, but that of LibSVM increases by a factor of 100. The testing time is also much shorter than LibSVM when the size of the training set is large.

Index Terms—classifier, clustering, large data set, scaling behavior.

I. INTRODUCTION

Fuzzy c -means based classifier (FCMC) [1], [2], [3], [4] is a classifier based on clustering approaches. We put FCMC in practical use for the vehicle detection system at several parking lots in Japan [4].

In this paper, for testing scaling behaviors of FCMC, the number of clusters is increased up to 8 for each class. Large-sized training sets with samples from 100,000 to 1,000,000 are used and the size of the training set is variously varied. Mixing proportions of clusters are parameterized. So, FCMC has many parameters, though the number of clusters is not very large. The parameters are optimized to training sets and the test set accuracy is evaluated on the test set.

When the number of training samples is relatively small, the classifier tends to over-fits the data. The obtained covariance matrices and cluster centers become incorrect, because the number of samples in each cluster is small.

For parameter optimization, PSO [5], [6], [7] is implemented just because PSO is as simple as the random search method. The selection of the optimization method is not essential because small improvement in training time on average is not significant in practice.

Support vector machines (SVMs) deliver superb performance in many real-world applications. SVMs are the standard tools for machine learning and data mining. We also give results with LibSVM: a library for support vector machines by Chang and Lin [8] for the sake of comparison. LibSVM incorporates SMO [9], which is a standard in SVM resolution for large size data sets and LibSVM is a state-of-the-art approach to SVM.

We need to compare the performance of FCMC from the following two points of view.

1) Comparison of scaling behaviors by varying the number of training samples. The number of clusters in each class is increased up to 8, and the mixing proportions of clusters are parameterized for FCMC. The parameters are optimized to training sets and are not regarded as free parameters.

2) Comparison of classification performances on very large datasets, when the number of clusters in each class is increased as many as possible for FCMC. The number of free parameters is set to less than or equal to 3 and the range for parameter search is limited for both FCMC and LibSVM.

This paper reports the results of 1). Free parameters (hyper-parameters) of LibSVM and FCMC, which are optimized to test sets or validation sets, are assumed to be known from the knowledge about the given set of data. Our results indicate that FCMC surpasses LibSVM with properly chosen parameters in classification accuracy when the size of training set is reduced by sampling. So, we can reduce the training time by a simple sampling of training data. For FCMC, the training time and the testing time are proportional to the size of training set and the size of testing set respectively.

The choice of the free parameters of SVM does not only

influence greatly the accuracy but also the training and testing time. These computation times are usually difficult to estimate in advance because the time largely depends on the number of support vectors, which is not known in advance. Our experiments on LibSVM show similar behaviors as those reported in the literature concerning training size and performance.

The advantages of FCMC with many parameters are the high generalization capability and the fast testing time.

II. FUZZY c -MEANS BASED CLASSIFIER

The first phase of FCMC is clustering or partitioning, which is modified from [10], [11]. See [3] for details of the derivation of the algorithm. The objective function of FCM with regularization by Kullback-Leibler divergence (KLFCM) [12], [10] is linearized and the update rules are derived by using Lagrangian multiplier method. Let $x_k \in \mathcal{R}^p$ be a feature vector and

$$D(x_k, v_i; S_i) = (x_k - v_i)^\top S_i^{-1} (x_k - v_i) \quad (1)$$

be squared Mahalanobis distance from x_k to $v_i \in \mathcal{R}^p$. S_i is a covariance matrix of data samples of the i -th cluster written as:

$$S_i = \frac{\sum_{k=1}^N u_{ki} (x_k - v_i)(x_k - v_i)^\top}{\sum_{k=1}^N u_{ki}}. \quad (2)$$

$$v_i = \frac{\sum_{k=1}^N u_{ki} x_k}{\sum_{k=1}^N u_{ki}} \quad (3)$$

is a cluster center and the mixing proportion of i -th cluster is

$$\alpha_i = \frac{\sum_{k=1}^N u_{ki}}{\sum_{j=1}^c \sum_{k=1}^N u_{kj}} = \frac{1}{N} \sum_{k=1}^N u_{ki}. \quad (4)$$

c denotes the number of clusters and N denotes the number of samples. The above updating rule is repeated and called as the generalized hard c -means [12], [11].

Initial locations of cluster centers or membership values are usually given randomly in the clustering approach. Since the classification performance is sensitive to the initialization, our approach does not use random initializations and use the principal component analysis (PCA) instead. Let $f_k, k = 1, \dots, N$ be PCA scores of the data set $X = (x_1, \dots, x_N)^\top$ of a class. f_k are associated with the largest singular value of mean corrected X . First, initial membership values of the two clusters of a class are given by

$$\begin{aligned} u_{k1} &= \begin{cases} 1 & (f_k \leq 0) \\ 0 & (\text{otherwise}) \end{cases} \\ u_{k2} &= \begin{cases} 0 & (f_k \leq 0) \\ 1 & (\text{otherwise}). \end{cases} \end{aligned} \quad (5)$$

This bisection method is repeated on each cluster, and the number of clusters (c) is increased. The resultant leaf nodes of the binary tree are the clusters. In this paper, the number of clusters is less than or equal to 8 for each class. The number of parameters to be optimized by PSO increases with the number

of clusters (c). Moreover, when c is large, the number of training samples in each cluster becomes small, and covariance matrices and cluster centers become inaccurate. This leads to the deterioration of classification accuracy. We can not use many more clusters for testing scaling behaviors.

One of the impediments for FCMC is the singularity of covariance matrices, which frequently occurs when feature dimension is relatively high and the number of samples in each cluster is small. So, the low-rank approximation of covariance matrices in the mixture of probabilistic principal component analysis (MPCA) [13] and the character recognition [14] is applied. By reducing the number r of basis vectors in the approximation of S_i , the algorithm becomes stably convergent. When $r=0$, S_i is a diagonal matrix, hence is non-singular, and $D(x_k, v_j; S_j)$ is reduced to Euclidean distance. The clustering is done on a per class bases.

The classification (i.e., the second phase) is performed by computing fuzzy memberships. Let π_q denote the mixing proportion of class q , i.e., the *a priori* probability of class q . Let α_{qj} be α_i in (4) for cluster j of class q . The class membership of k -th data x_k to class q is computed as:

$$u_{qjk}^* = \alpha_{qj} |S_{qj}|^{-\frac{1}{\gamma}} (D(x_k, v_{qj}; S_{qj}) + \nu)^{-\frac{1}{m}}, \quad (6)$$

$$\tilde{u}_{qk} = \frac{\pi_q \sum_{j=1}^c u_{qjk}^*}{\sum_{s=1}^Q \pi_s \sum_{j=1}^c u_{sjk}^*}, \quad (7)$$

where c denotes the number of clusters of each class and Q denotes the number of classes. We selected the functional form of u^* based on the membership functions derived from the generalized FCM objective function [12] and that of FCM with regularization by Kullback-Leibler divergence [12], [10].

At the completion of clustering for all classes in the first phase of FCMC, we then compute Mahalanobis distances by (1) for all the samples in the training sets. The second phase of FCMC is the supervised classification and the parameters, i.e., $m \in [0, 2], \gamma \in [0, 20], \nu \in [0, 200]$ and the change rates of $\alpha_{qj} \in [-0.5, 0.5]$ (see [3]), are selected to minimize error rate on the training sets. Since we have Q classes and c clusters for each class, the number of α_{qj} is $Q \times c$. The distances by (1) for all the samples in the training set are fixed and the best setting of the parameters are searched by PSO [5], [6], [7]. The free parameters (C, γ) of LibSVM in Section III are chosen to fit the given test set. But, the parameters ($m, \gamma, \nu, \alpha_{qj}$) of FCMC are optimized to training set since the number of parameters is much larger than LibSVM. After the optimization, the test set accuracy on the test set is evaluated.

III. EXPERIMENTS

In this Section, we implement FCMC and LibSVM, and illustrate the scaling behavior of FCMC on several benchmark data sets. Table I summarizes the characteristics of the data sets used. The parking lot image data were used for the vehicle detection system in [4]. Checkerboard data are artificial data. Other data sets are downloaded from <http://www.cse.ust.hk/~ivor/cvm.html> (Ivor W. Tsang, Andras Kocsor, James T. Kwok [15]). FCMC is implemented in

Matlab. FCMC has several to dozens of parameters, which are optimized by PSO. But FCMC has other parameters that are not automatically optimized. They are the number c of clusters and the number r of basis vectors for the low-rank approximation of covariance matrices. c is chosen from $\{2, 4, 8\}$ and $r \in \{0, 2, 5, 20, 40, 30\}$ and they also need to be selected. But in this paper, we assume that these two parameters are the free parameters and their optimum values are known from the knowledge about the given data sets. The free parameters (C, γ) of LibSVM are also assumed to be known. We use the Gaussian kernel $k(x, y) = \exp(-\gamma|x - y|^2)$ for nonlinear classification. The kernel parameter γ and the regularization parameter C are the free parameters of LibSVM.

For medium-sized problems, cross validation might be the most reliable way for parameter selection. But, for large-sized problems as in Table I, it is hard to implement cross validation procedure since it takes very long computation time. Therefore, in this paper PSO in FCMC is used for optimizing training accuracy on training set. Then, we evaluate the accuracy on the given test set. The free parameters of LibSVM are also chosen manually. We tested various parameter values including those reported in the literatures of SVM, and we report the best results.

Experiments are performed on DELL T5400 3.16GHz, 4GB. Each experiment reports the error rate on test set, the training time on training set, and the testing time (i.e., the detection time).

TABLE I
DATA SETS USED FOR PERFORMANCE COMPARISON

data	features	training	testing
Parking lot image	50 (1,024)	135,000	135,000
USPS zero-one	50 (676)	266,079	75,383
Checkerboard	2 (2)	1,000,000	2000
KDD-CUP 1999	5 (127)	2,449,216 (4,989,431)	311,029
MNIST	50 (576)	60,000	10,000

Parking Lot Image Data

First, we use the set of image data with 50 feature dimensions. The image data are used for the vehicle detection system in [4]. The camera images of a parking lot are first stored in JPEG format of 704×576 . Vehicle/parking space sub-windows are cropped from each picture. Each RGB image of the sub-windows is thinned as $32 \times 32 \times 3$ and transformed into HSV and then into gray scale image of 32×32 (=1,024). 50 PCA (principal component analysis) basis vectors were computed using a subset of sample images. PCA for image compression is also known as the Hotelling, or Karhunen and Leove (KL), transform. All the collected gray scale image data were compressed by multiplying PCA basis vectors (i.e., inner product). The obtained 270,000 vectors with 50 feature dimensions were used. The first half (135,000 image data)

is used for training and the latter half (135,000) is used for testing. The best accuracy by LibSVM is obtained with $C = 1.0$ and $\gamma = 5.0$. We found this parameter values by testing many times. We make the training size varies. Fig. 1 shows the results for experiments where the training size is growing. The error rate of FCMC on the test set is smaller than LibSVM as shown in Fig. 1, though the training time (dashed line) is slightly longer than LibSVM with the proper values of C and γ . The straight line with circles show the result of FCMC with the best settings of the parameters, while the dashed line shows the result by PSO for each reduced set of training samples. It should be noted that to increase the numbers of iterations and particles of PSO does not necessarily improve the testing accuracy, since PSO optimize the parameters to training sets. The straight line with circles shows the results when the parameters are fixed to the best setting, i.e., the change rate of $\alpha_{qj} = (0.2035, 0.7642, 0.3540, 0.1885, 0.7342, 0.7632, 0.4914, -0.1085, 0.6106, 0.1501, 0.7356, 0.5202, 0.4685, 0.9049, 0.2748, 1.1440)$, $m=0.1143$, $\gamma=8.2049$, and $\nu=173.8178$. The testing time of LibSVM is much longer than that of FCMC when the size of training set is large as shown in Fig. 3. For FCMC, we set as $c=8$, and $r=30$. The number of particles in PSO is 50 and the number of iterations is 10 (denoted as 50×10).

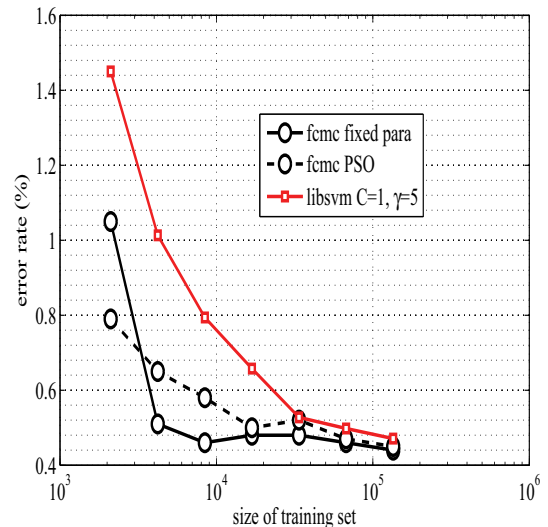


Fig. 1. Error rates (%) on the test sets of the parking lot image data (FCMC: PSO 50×10 , $c=8$, $r=30$). Note that the size of the training set is in log scale.

Extended USPS Zero-one Data

In this experiment, our task is to classify digits zero from one in an extended version of the USPS data set. The original training set has 1,005 zeros and 1,194 ones, while the test set has 359 zeros and 264 ones. To better study the scaling behavior, Tsang, Kwok and Cheung [15] extend this data set by first converting the resolution from 16×16 to 26×26 , and then generate new images by shifting the original ones in all directions for up to

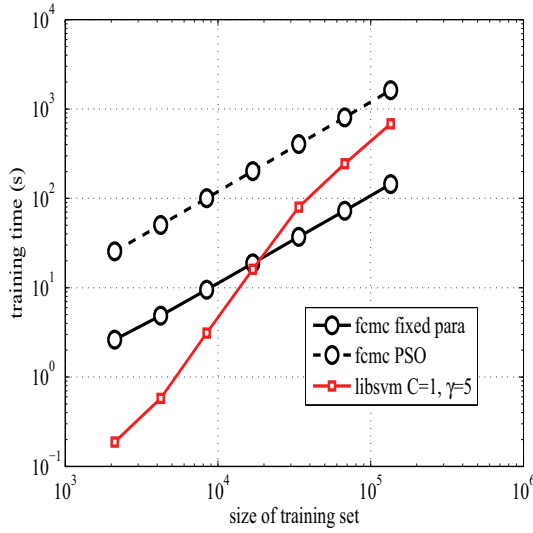


Fig. 2. Training time (s) on the parking lot image data (FCMC: PSO 50×10 , $c=8$, $r=30$). Note that the training time is also in log scale.

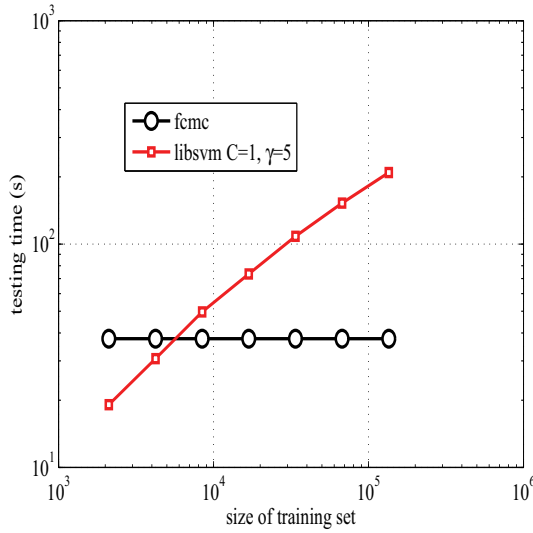


Fig. 3. Testing time (s) on the test sets of the parking lot image data (FCMC: PSO 50×10 , $c=8$, $r=30$)

five pixels. Thus, the resultant training set has a total of $(1005+1194) \times 112 = 266,079$ samples while the extended test set has $(359+264) \times 112 = 75,383$ samples (<http://www.kernel-machines.org/data/usps.mat.gz>). 50 PCA basis vectors were computed using a subset of image data. All the image data were compressed by multiplying PCA basis vectors. The obtained training and testing samples with 50 feature dimensions were used. As shown in Fig. 4, the accuracy of LibSVM deteriorates with the decreasing number of training samples. The straight line with circles show the result of FCMC with the best settings of the parameters, while the dashed line shows the result by a single trial of PSO (20×10) for each reduced set of training samples. Therefore, the training time can not be reduced for LibSVM by a simple sampling of the training data, though FCMC's accuracy is high for a wide range of training

set sizes. When the size of training set is larger than 30,000, the best set of the parameters in terms of testing accuracy is $C = 1, \gamma = 0.03$ for LibSVM, and the training time is nearly equal to FCMC with $c = 8$, and $r = 5$. This, however, is true only when we know the best parameter values. If we test LibSVM with $C = 1, \gamma = 0.1$, it takes 10 times longer than FCMC as shown in Fig. 5. So, the total training time including parameter search may become extremely long for LibSVM. And, the testing time is longer than that of FCMC by more than an order of magnitude as shown in Fig. 6.

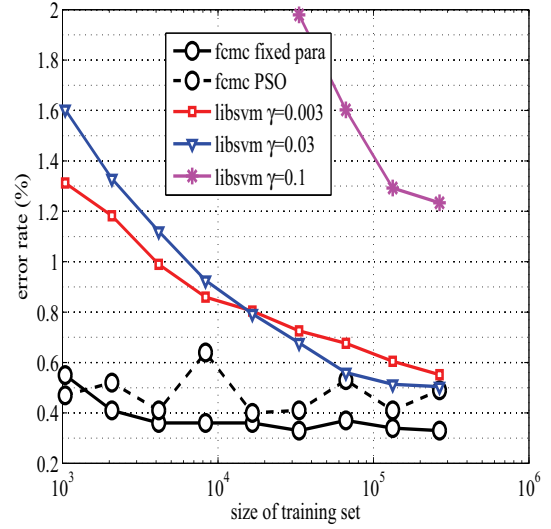


Fig. 4. Error rates (%) on the test sets of the extended USPS digits data (LibSVM: $C=1$, FCMC: PSO 20×10 , $c=8$, $r=5$).

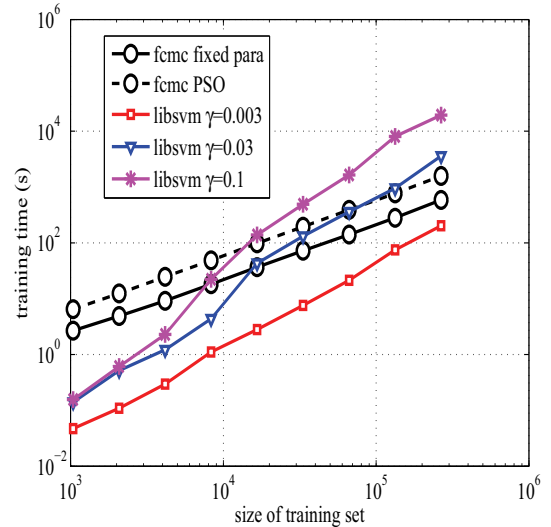


Fig. 5. Training time (s) on the extended USPS digits data (LibSVM: $C=1$, FCMC: PSO 20×10 , $c=8$, $r=5$)

Checkerboard Data

We experiment on the 4×4 checkerboard data commonly

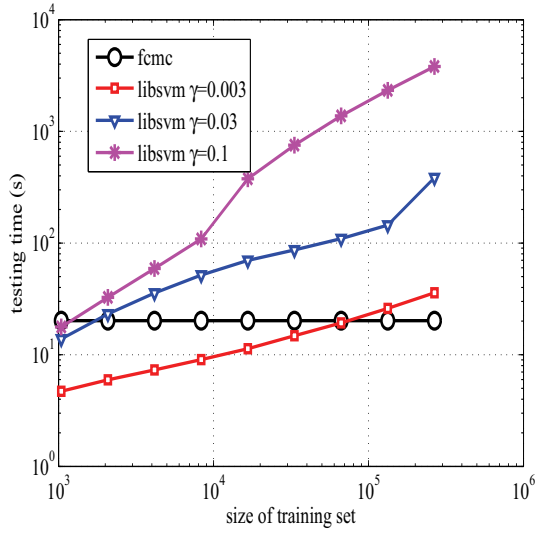


Fig. 6. Testing time (s) on the test sets of the extended USPS digits data (LibSVM: $C=1$, FCMC: PSO 20×10 , $c=8$, $r=5$)

used for evaluating large-scale non-linear SVM implementations [15], [16]. We use training sets with a maximum of 1 million points and 2,000 independent points for testing.

Fig. 7 shows that the accuracy of both approaches deteriorates as the training samples are reduced by sampling. For relatively small training sets, with less than 100,000 samples, the error rate of LibSVM is twice larger than FCMC. But, LibSVM is faster as shown in Fig. 8. When the number of training samples is 1,000,000, the training time is nearly the same. This, however, is true when we know the best parameter values of LibSVM. If we test LibSVM with $C = 10, \gamma = 0.001$, it takes 10 times longer than FCMC. So, the total training time of LibSVM for parameter search may become extremely long. The testing time for LibSVM with the best parameter setting is 10 times longer than FCMC as shown in Fig. 9. As we know the clusters are all squares of the same size on the checkerboard and the number of clusters is 8 for each class, we set as $c = 8, r = 0$

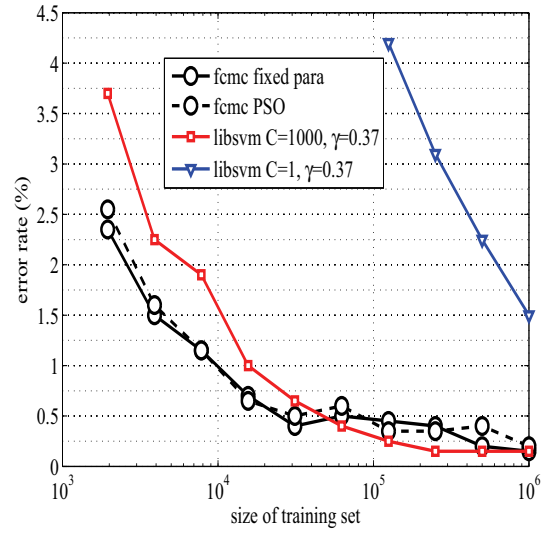


Fig. 7. Error rates (%) on the test sets of the checkerboard data (FCMC: PSO 50×10 , $c=8$, $r=0$)

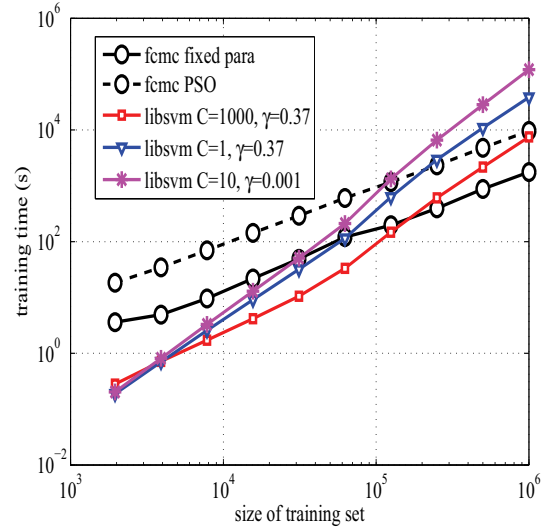


Fig. 8. Training time (s) on the checkerboard data (FCMC: PSO 50×10 , $c=8$, $r=0$)

KDDCUP-99 Intrusion Detection Data

This intrusion detection data set has been used for the Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99. Each original sample has 34 continuous features and 7 symbolic features (<http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>).

The original training set contains 4,898,431 connection records, which are processed from about four gigabytes of compressed binary TCP dump data from seven weeks of

network traffic. Another two weeks of data produced the test data with 311,029 samples. The data set includes a wide variety of intrusions simulated in a military network environment. There are a total of 24 training attack types, and an additional 14 types that appear in the test data only. The task is to separate normal connections from attacks. We used 2 million samples for training. Each sample is compressed with PCA and has only 5 feature (PCA score) dimensions because of not enough memory. As the test results show, the accuracy does not deteriorate by the significant reduction of the feature dimensionality. This is the case only for the KDD data. We computed 5 PCA basis vectors by using first 300,000 training samples and then multiply them with all training and testing samples. We used the compressed data for both FCMC and LibSVM.

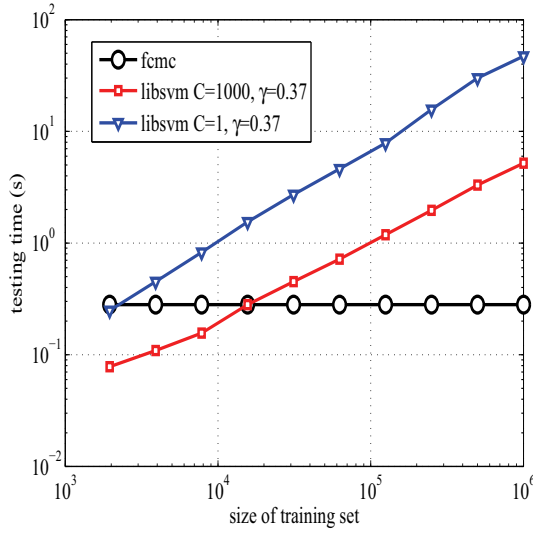


Fig. 9. Testing time (s) on the test sets of the checkerboard (FCMC: PSO 50×10 , $c=8$, $r=0$)

When $C = 1,000,000$ and $\gamma = 0.1$, LibSVM attains the best accuracy, but the differences between LibSVM with some other parameter settings and FCMC are slight for a wide range of sizes of training samples as shown in 10. The training time for FCMC is shorter than LibSVM when the size is larger than 1,000,000 as shown in Fig. 11. The testing time is shorter than LibSVM with $C = 1, \gamma = 0.0001$ by two orders of magnitude as shown in Fig. 12.

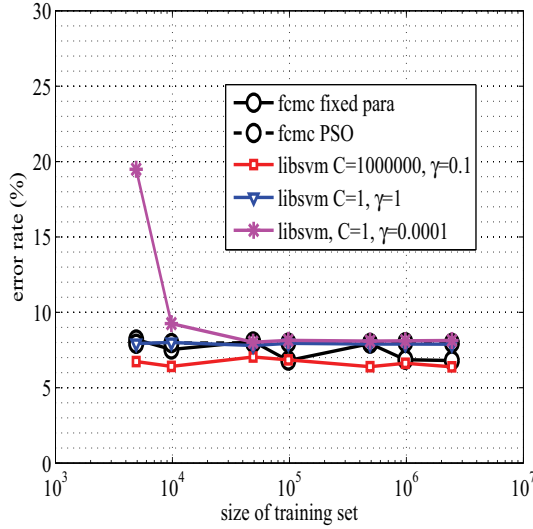


Fig. 10. Error rates (%) on the test sets of the KDD data (FCMC: PSO 20×10 , $c=2$, $r=2$)

MNIST Data

The MNIST database of handwritten digits (10-class problem), has a training set of 60,000 samples, and a test set of 10,000 samples. It is a subset of a larger set available from NIST by LeCun research group at AT&T lab [17]. A 576-dimensional directional feature based

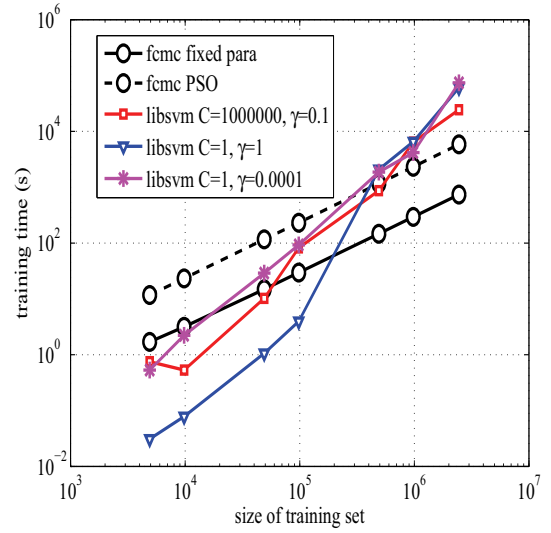


Fig. 11. Training time (s) on the KDD data with (FCMC: PSO 20×10 , $c=2$, $r=2$)

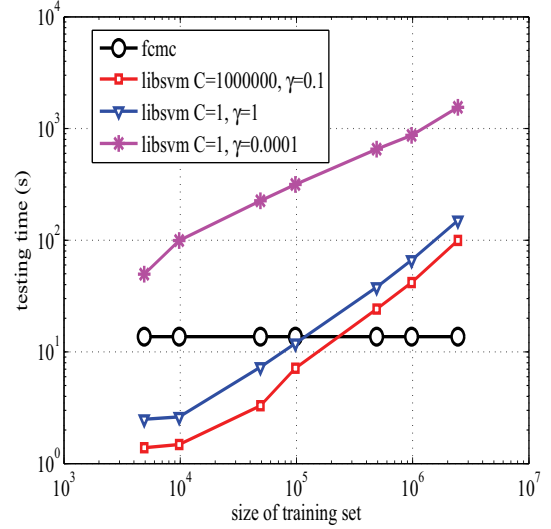


Fig. 12. Testing time (s) on the test sets of the KDD data (FCMC: PSO 20×10 , $c=2$, $r=2$)

on gradient of gray-level image was extracted by using Robert edge operator. The data set is available from <http://www.cenparmi.concordia.ca/~jdong/HeroSvm.html>.

Results are shown in Figs. 13-15. When the number of training samples is smaller, the accuracy of FCMC is better for $c = 2$, but when the number is larger than 20,000, $c = 8$ is better. LibSVM use the “one-against-one” approach in which $Q(Q-1)/2$ classifiers are constructed and each one trains data from two different classes. In classification a voting strategy is used: each binary classification is considered to be a voting where votes can be cast for all data points x - in the end point is designated to be in a class with maximum number of votes. LibSVM’s training time and testing time is short, since the number of training samples handled by each classifier is small. We used the same data set of 50 dimensional feature vectors for

both LibSVM and FCMC. It should be noted that the training time does not include the time for parameter search, hence in practice a long computation time is indispensable for LibSVM.

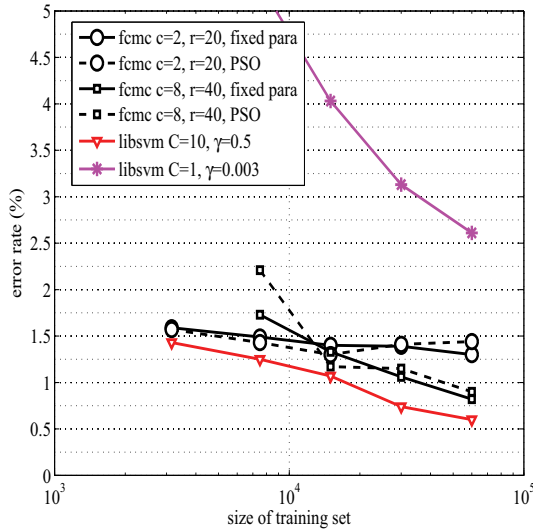


Fig. 13. Error rates (%) on the test sets of the MNIST 10-class data (FCMC: PSO 20×10)

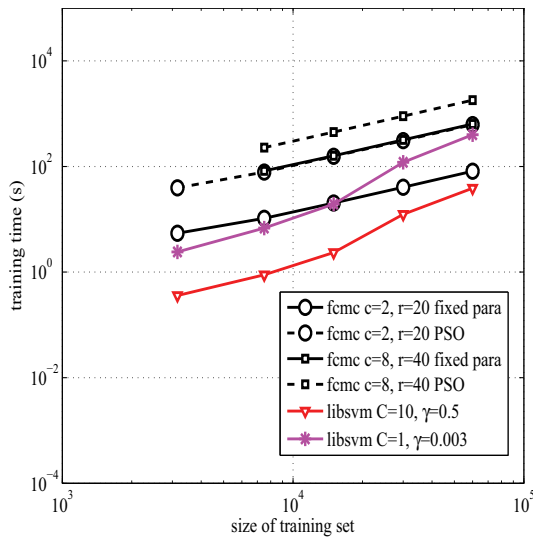


Fig. 14. Training time (s) on the MNIST 10-class data with (FCMC: PSO 20×10)

IV. CONCLUSION

We have demonstrated the performance of FCMC by comparing with LibSVM. We used five large sized data sets whose feature values are mainly continuous real numbers.

When the number of clusters of FCMC is increased, in order for cluster covariance matrices and cluster centers to be correctly estimated, a very large-sized training set is necessary. Therefore, we tested the performance with the number of clusters less than or equal to 8, and the number of parameters $c \times Q + 3$. The results are summarized as follows:

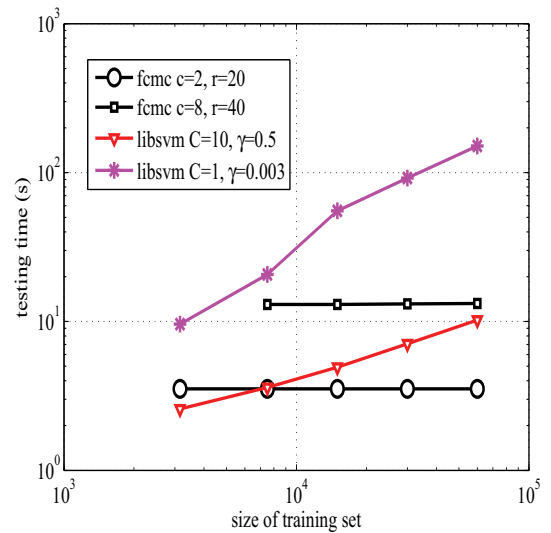


Fig. 15. Testing time (s) on the test sets of the MNIST 10-class data (FCMC: PSO 20×10)

1) When the number of training samples is large, the difference in the test set accuracy (i.e., the generalization capability) between FCMC and LibSVM is slight.

2) When the size of training set is reduced, the test set accuracy of FCMC surpasses LibSVM in many data sets.

3) Training time of FCMC proportionally increases with the size of training set (i.e., $O(N)$). But, the time needed for LibSVM increases more rapidly than FCMC (i.e., $O(N^2)$ for LibSVM). Thus, the training is faster than LibSVM, when the size of training set is large.

4) The testing time (detection time) is faster than LibSVM by an order of magnitude, when the size of training set is large.

There may be other types of benchmark data such as those with many categorical features. To confirm the discriminative power on various benchmark data is left for our future work.

REFERENCES

- [1] H. Ichihashi, K. Honda, A. Notsu and K. Ohta, "Fuzzy c -means classifier with particle swarm optimization," *Proc. of the IEEE International Conference on Fuzzy System, World Congress on Computational Intelligence*, Hong Kong, China, pp. 207-215, 2008.
- [2] H. Ichihashi, A. Notsu and K. Honda, "Triplet of FCM classifiers," *Proc. of 2009 IEEE International Conference on Fuzzy System*, Jeju, Korea, Aug. 20-24, 2009.
- [3] H. Ichihashi, A. Notsu and K. Honda, "Semi-hard c -means clustering with application to classifier design," *Proc. of the IEEE International Conference on Fuzzy System, World Congress on Computational Intelligence*, Barcelona, Spain, pp. 2788-2795, 2010.
- [4] H. Ichihashi, A. Notsu, K. Honda, T. Katada and M. Fujiyoshi, "Improvement in the performance of camera based vehicle detector for parking lot," *Proc. of 2010 IEEE International Conference on Fuzzy System*, Barcelona, Spain, Jul. 18-23, 2010.
- [5] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Proc. of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39-43, 1995.
- [6] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proc. of the IEEE International Conference on Neural Networks*, Piscataway, NJ, vol. 4, pp. 1942-1948, 1995.
- [7] M. Clerc, *Particle Swarm Optimization*, Wiley, 2006.
- [8] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

- [9] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization.," *Advances in Kernel Methods - Support Vector Learning*, Eds. B. Scholkopf, C. J. C. Burges, and A. J. Smola, Cambridge, MA, 1998. MIT Press, pp. 185-208, 1999.
- [10] H. Ichihashi, K. Miyagishi and K. Honda, "Fuzzy *c*-means clustering with regularization by K-L information," *Proc. of 10th IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, vol.3, pp.924-927, 2001.
- [11] S. Miyamoto, T. Yasukochi, R. Inokuchi, "A family of fuzzy and defuzzified *c*-means algorithms," *Proc. of the International Conference on Computational Intelligence for Modelling, Control and Automation*, Vienna, Austria, pp.170-176, 2005.
- [12] S. Miyamoto, H. Ichihashi, and K. Honda, *Algorithms for Fuzzy Clustering, Methods in c-Means Clustering with Applications*, Springer-Verlag, Berlin, 2008.
- [13] M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol.11, pp.443-482, 1999.
- [14] F. Sun, S. Omachi, and H. Aso, "Precise selection of candidates for hand written character recognition," *IEICE Trans. Information and Systems*, vol.E79-D, no.3, pp.510-515, 1996.
- [15] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines - Fast SVM training on very large data sets," *Journal of Machine Learning Research*, vol.6, pp. 363-392, 2005.
- [16] G. Loosli and S. Canu, "Comments on the 'Core vector machines: Fast SVM training on very large datasets'," *Journal of Machine Learning Research*, vol.8 , pp. 291-301, 2007.
- [17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.