1-1-2010

Efficient SVM training with reduced weighted samples

Son Lam Phung
University of Wollongong, phung@uow.edu.au

Giang Hoang Nguyen
University of Wollongong, giang_nguyen@uow.edu.au

Abdesselam Bouzerdoum
University of Wollongong, bouzer@uow.edu.au

Efficient SVM training with reduced weighted samples

Abstract

This paper presents an efficient training approach for support vector machines that will improve their ability to learn from a large or imbalanced data set. Given an original training set, the proposed approach applies unsupervised learning to extract a smaller set of salient training exemplars, which are represented by weighted cluster centers and the target outputs. In subsequent supervised learning, the objective function is modified by introducing a weight for each new training sample and the corresponding penalty term. In this paper, we investigate two methods of defining the weight based on cluster vectors. The proposed SVM training is implemented and tested on two problems: (i) gender classification of facial images using the FERET data set; (ii) income prediction using the UCI Adult Census data set. Experiment results show that compared to standard SVM training, the proposed approach leads to much faster SVM training, produces a more compact classifier while maintaining generalization ability.

Keywords

reduced, weighted, training, svm, samples, efficient

Disciplines

Physical Sciences and Mathematics

Publication Details

Nguyen, G. Hoang, Phung, S. & Bouzerdoum, A. (2010). Efficient SVM training with reduced weighted samples. WCCI 2010 IEEE World Congress on Computational Intelligence (pp. 1764-1768). USA: IEEE.

WCCI 2010 IEEE World Congress on Computational Intelligence
July, 18-23, 2010 - CCIB, Barcelona, Spain IJCNN

Efficient SVM Training with Reduced Weighted Samples

Giang Hoang Nguyen, Son Lam Phung, *Member, IEEE*, and Abdesselam Bouzerdoum, *Senior Member, IEEE*

***Abstract*— This paper presents an efficient training approach for support vector machines that will improve their ability to learn from a large or imbalanced data set. Given an original training set, the proposed approach applies unsupervised learning to extract a smaller set of salient training exemplars, which are represented by weighted cluster centers and the target outputs. In subsequent supervised learning, the objective function is modified by introducing a weight for each new training sample and the corresponding penalty term. In this paper, we investigate two methods of defining the weight based on cluster vectors. The proposed SVM training is implemented and tested on two problems: (i) gender classification of facial images using the FERET data set; (ii) income prediction using the UCI Adult Census data set. Experiment results show that compared to standard SVM training, the proposed approach leads to much faster SVM training, produces a more compact classifier while maintaining generalization ability.**

I. Introduction

With the advance of technology, the ability of collecting large and high dimensional data sets in domains such as finance forecasting, geosciences, biomedical, network intrusion detection, credit card fraud detection, and medical diagnosis increases. While these fast growing data sets provide an opportunity to build high-quality predictive models, they also impose several difficulties in computation, storage, processing and learning from such large and quite often, imbalanced data.

Support vector machine (SVM) is a powerful technique for pattern classification [1]. It improves generalization ability of the classifier by maximizing the margin between the two classes. SVM training involves solving a quadratic programming (QP) problem, which is computation intensive, especially for a large data set. In standard SVM training, all training samples are treated equally and this may lead to poor performance on imbalanced data sets, where there are more samples from one class than another. This class imbalance problem is severe in many real-world applications where the emphasis is to detect patterns from the minority class.

To reduce computation load in training the SVM, several techniques have been proposed which can be divided into two categories: (i) modifying the standard SVM training so that it could be applied to large data sets, and (ii) selecting a small number of representative training samples from the original, large data set so that the standard SVM training could handle.

The first SVM training approach involves dividing the original quadratic programming (QP) problem into smaller sub-problems, thereby reducing the size of each QP problem. Examples of this approach include chunking, decomposition [2], [3], [4], and Sequential Minimal Optimization (SMO) [5], [6].

The second SVM training approach uses mixture of models and Bayesian committee machine (BCM) for large data sets. A parallel mixture of SVMs is proposed by Collobert [7], in which several SVMs models are trained on small subsets of samples and then combined using a gater such as linear hyperplane or multi-layer perceptron. Schwaighofer and Tresp [8] apply BCM to SVM training. The BCM divides data into M sets of approximately equal size and M models are built on those sets. The final prediction is the combination of all individual models using a Bayesian-based weighting scheme. However, the computational complexity of the BCM scales linearly with the number of training samples, and the optimal number of models must be determined.

Recently, selective sampling or active learning have been used to reduce the number of training data points [9], [10]. Active learning optimizes learning from a minimum number of data points, by intelligently selecting training samples from the entire data set. It is an iterative process that avoids redundant or non-informative samples. When applied to SVM training, a SVM model is constructed on an initial small subset of samples. The trained model is then used to query new samples to add to the existing training set and this step is repeated until convergence. At each iteration, this approach essentially selects data points close to the decision boundary, because they have a higher chance of being the support vectors. Examples of SVM training based on active learning include probabilistic active support vector learning algorithm [12] and confident-based active learning [11].

In this paper, we propose to combine supervised and unsupervised learning to address the computation load and the class imbalance problem when training with SVMs. The key idea of our proposed approach is to reduce the original training set to a manageable size via unsupervised clustering. To compensate for information loss during clustering, we introduce a weight factor that is associated with each new labeled training sample, and modify the objective function to include the weight factor in each penalty term. The paper is organized as follows: Section II describes our proposed training approach for SVMs and discusses two methods of assigning the weight factors. Section III presents experimental results and analysis of the proposed approach, and compares it with other approaches. Section IV gives the conclusion and directions for further work.

II. Proposed Approach

We propose a reduced, weighted SVM approach (RW-SVM) that combines supervised and unsupervised learning for

Authors are with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, Australia (email: giang@uow.edu, phung@uow.edu.au, a.bouzerdoum@ieee.org)

designing SVM classifiers. Our approach involves a pre-processing step to reduce the number of training samples to a manageable size. To this end, we apply unsupervised clustering on the original data set to extract cluster centers which form a more compact representation. To reduce information loss, the centroids of clusters and associated weights are used for SVM training. This approach is suitable for many real-world applications where the saliency of training data varies significantly from sample to sample, or there is significant data redundancy.

A. Reduced, weighted SVM training

Consider the SVM training problem that involves a training set D of M samples,

$$(\mathbf{x}_1, y_1), (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_M, y_M),$$

where each training sample $\mathbf{x}_i$ in the N-dimensional space is given a label y_i in $\{-1, +1\}$. In standard SVM training, samples are given equal weighting. However, we consider the case when training samples have unequal weights. Suppose that training sample $\mathbf{x}_i$ is associated with a weight p_i, where

$$p_i \in (0,1) \quad \text{and} \quad \sum_{i=1}^{M} p_i = 1. \tag{1}$$

There are many ways to define the weights and they will be discussed in Section II-B. Here, we focus on deriving a training algorithm for SVM that takes into account the sample weights.

To derive the training algorithm, an intuitive approach is to treat the weight as the "frequency" of observing a sample in the training set. Then, SVM training can be formulated as

$$\text{minimize} \quad \Phi(\mathbf{w}, \xi) = \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{M} p_i \xi_i, \tag{2}$$

subject to

$$\begin{aligned} y_i(\mathbf{w}^T \mathbf{x}_i + b) &\geq 1 - \xi_i, \quad i = 1, \ldots, M && (3) \\ \xi_i &\geq 0, \qquad i = 1, \ldots, M. && (4) \end{aligned}$$

Here, ξ_1,ξ_2,...,ξ_M are the error margins and C is a cost parameter that determines the trade-off between maximizing the class margin and minimizing the training error. To solve this constrained optimization problem, we introduce nonnegative Lagrangian multipliers α_i and β_i. The optimization problem becomes minimizing

$$\begin{aligned} Q(\mathbf{w}, b, \alpha, \beta) &= \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{M} p_i \xi_i \\ &\quad - \sum_{i=1}^{M} \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i) \\ &\quad - \sum_{i=1}^{M} \beta_i \xi_i. && (5) \end{aligned}$$

Let $\xi = (\xi_1, \xi_2, \ldots, \xi_M)^T$, $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_M)^T$ and $\beta = (\beta_1, \beta_2 \ldots, \beta_M)^T$. The optimal solution must satisfy the following KarushKuhnTucker (KKT) conditions:

$$\begin{aligned} \frac{\partial Q(\mathbf{w}, b, \alpha, \beta)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{M} \alpha_i y_i \mathbf{x}_i = 0 && (6) \\ \frac{\partial Q(\mathbf{w}, b, \alpha, \beta)}{\partial b} &= -\sum_{i=1}^{M} \alpha_i y_i = 0 && (7) \\ \frac{\partial Q(\mathbf{w}, b, \alpha, \beta)}{\partial \xi} &= p_i C - \alpha_i - \beta_i = 0. && (8) \end{aligned}$$

Therefore, substituting these conditions into (5), we obtain the following dual problem: Maximize

$$Q(\alpha) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \tag{9}$$

subject to

$$\sum_{i=1}^{M} y_i \alpha_i = 0, \text{ and } 0 \leq \alpha_i \leq p_i C, \quad i = 1, 2, \ldots, M. \tag{10}$$

The KKT conditions of reduced weighted SVM become

$$\begin{aligned} \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i) &= 0, \; i = 1, 2, \ldots, M && (11) \\ (p_i C - \alpha_i) \xi_i &= 0, \; i = 1, 2, \ldots, M && (12) \end{aligned}$$

B. Defining the sample weight

Given an original training set, we apply unsupervised clustering on all training samples that belong to a particular class. That is, unsupervised clustering is performed independently on each class. There are numerous clustering techniques including the K-means [13], fuzzy C-means [14], hierarchical clustering [15], and self-organizing maps [16]; for a detailed review, the reader is referred to [15]. Any of these clustering techniques can be applied in our approach.

Once clustering is completed, we use each cluster centroid as a new training sample. The label of the new training sample is derived from the label of the corresponding cluster. Note that only new training samples are used in subsequent SVM training. Although it is possible to give all new training samples equal weights, to retain useful information we propose to assign a custom weight to each new training sample.

Suppose there are S clusters. For cluster i, let $\mathbf{x}_i^n$ be the cluster centroid and z_i be the cluster size. That is, z_i is the number of original samples in the cluster i. Clearly, we have

$$\sum_{i=1}^{S} z_i = M. \tag{13}$$

Here, we describe two ways of defining the weight of the new training sample.

- *Method 1: Weight proportional to class size.* Let R be number of classes in the training set, N_j is the size of class j. Let $\gamma_{i,j}$ is the degree membership of cluster i to class j, that is,

$$\gamma_{i,j} = \begin{cases} 1, & \text{cluster } i \text{ belongs to class } j \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

The weight for cluster centroid $\mathbf{x}_i^n$ is defined as

$$p_i = \frac{z_i}{R\sum_{j=1}^{R} N_j \gamma_{i,j}}. \quad (15)$$

- *Method 2: Weight proportional to cluster size.* The weight of cluster centroid $\mathbf{x}_i^n$ is proportional to the cluster size. That is,

$$p_i = \frac{z_i}{M}, \quad i = 1, 2, \ldots, S. \quad (16)$$

This weight definition does not address the imbalanced distribution among classes, where one class has more samples than the other.

III. Experiments and Analysis

We evaluate the performance of the proposed SVM training approach on two problems: (i) gender classification of facial images on the standard FERET data set [17], and (ii) income prediction on the UCI Adult Census data set [18]. Our aim is to analyze the capability of the proposed method in handling large and imbalanced data sets in term of training time, and generalization performance. The well-known LibSVM software is used in our experiments [19]. All experiments are done in a Intel Corel Quad 2.66GHz machine with 3G memory.

A. Gender classification of facial images on FERET database

The FERET database consists of $14,051$ gray-scale images of human faces, stored in several data sets. There are two data sets for frontal faces: (i) data set fa has 1762 images and (ii) data set fb has 1518 images. Since there is a significant overlap between these two data sets, we only use the images in data set fa in this paper. In our experiments, the extracted face patterns were histogram-equalized and then scale to the range $[-1, 1]$. A five-fold cross validation was performed on the entire data set of 1762 face patterns. For each fold, 1408 patterns were used for training and 354 patterns were used for testing. The final classification rates were obtained by averaging over the five folds.

Moghaddam and Yang [20] used SVMs with the RBF kernel for gender classification and evaluate their classifier on a set of 1755 FERET face images (1044 males faces and 713 female faces). They achieved a classification rate of 96.6%, and their SVM approach is considered as the state-of-the-art in gender classification. Moghaddam and Yang found that the difference between classification rates when using low-resolution (21×12 pixels) and high-resolution (84×48 pixels) image is only 1%. Therefore, we only use image size of 21×12 pixels in our experiments, and hence each training samples has 252 attributes.

B. Income prediction on UCI Adult Census data set

The UCI Adult data set is a well-known and widely used benchmark data set in pattern classification and data mining. The task is to predict whether a person income is greater than 50K per year or not. Each sample has 14 attributes that include ages, work class, education, occupation, race, relationship and sex. Six attributes are continuous and eight attributes are symbolic. For the symbolic attributes, we assigned each name of symbolic attribute with a number. For examples, in the sex attribute we assign a value of 1 to 'female' and 2 to 'male' attribute. Then we normalized and scaled all the attributes values to the range of $[-1, 1]$. The data set has some missing attributes; we replaced each missing attribute with the mean value of the attribute from the entire data set. A five-fold cross-validation was performed on the entire data set of $45,222$ samples. For each fold, $33,916$ samples were used for training and $11,306$ samples were used for testing.

We observe that there is an imbalanced distribution between the two income classes. In the training set, $25,510$ samples belong to the majority class, and $8,406$ samples belong to the minority class. In the test set, the majority class has $8,504$ samples whereas the minority class has $2,802$ samples.

C. Comparison of techniques for training data reduction

We implemented two approaches of selecting training data from a large data set so that the standard SVM training could handle. The first approach selects the training samples randomly from the original set. The second approach finds representative training samples via clustering. In this study, we adopt the K-means clustering algorithm; this algorithm requires little parameter tuning and is quite effective in handling large data sets [13]. We also study the effects of replacing the original data by the cluster centroids and custom weights. Overall, four techniques are compared:

- Random-SVM: standard SVM training using samples that are randomly selected from the original data set.
- Cluster-SVM: standard SVM training using the cluster centroids; no information on the weights is used.
- RW-SVM1: proposed SVM training which takes into account both cluster centroids and custom weights. Method 1 of defining the sample weight is used.
- RW-SVM2: proposed SVM training which takes into account both cluster centroids and custom weights. Method 2 of defining the sample weight is used.

Each technique is applied to train SVMs with the RBF kenel. The SVM classifier has two key parameters: the penalty parameter *C* and γ of the RBF kernel. These parameters are determined though cross validation.

The classification rates (CRs) of the different training techniques on the gender classification problem are presented in Table I and Fig. 1. The classification rates for different sizes of the training set are given in the table.

Clearly, using unsupervised clustering to select training samples (Cluster-SVM, RW-SVM1 and RW-SVM2) achieves higher classification rates compared to selecting training samples randomly (Random-SVM). Furthermore, the proposed SVM training approach, RW-SVM, achieves the highest CR, among the four tested techniques. Compared to other techniques, the improvement in the classification rate of RW-SVM is most significant when the number of training

TABLE I

COMPARISON OF FOUR SVM TRAINING TECHNIQUES ON GENDER CLASSIFICATION TASK.

Training algorithms	Number of training samples				
	424	494	564	634	706
Random-SVM	93.42	94.78	94.61	94.84	95.29
Cluster-SVM	95.35	95.12	95.91	96.03	95.46
RW-SVM1	95.29	95.06	95.86	96.14	95.57
RW-SVM2	95.35	95.06	95.86	96.20	95.63

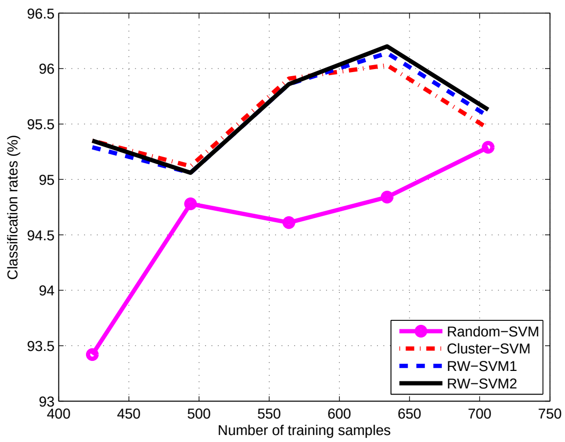

Fig. 1. The classification rates of four SVM training techniques when different numbers of training samples are used.

samples is small. For examples, on 424 training samples, RW-SVM2 has a CR of 95.35% with 95% confident interval of [94.37, 96.33], whereas Random-SVM has a CR of only 93.42%. For Random-SVM, a random subset of the original samples is used for training. These results show that combining clustering and the new objective function provides extra information in the extracted training samples.

D. Generalization performance and computational time

In this section, we compare the generalization performances of the reduced, weighted SVM training and the standard SVM training. Standard SVM training is performed using the LibSVM software package: It uses the entire original training set, that is 1408 samples for the gender classification task, and $33,916$ samples for the UCI Adult census data set. The reduced, weighted SVM training uses only a fraction of the original data size: 634 cluster centroids for the gender classification task, and 2374 cluster centroids (about 7% of the original size) for the UCI Adult census data set. The classification rates of different training algorithms are shown in Table II and Table III for the gender classification and income prediction task, respectively.

The standard SVM training and the proposed SVM training achieve almost similar classification rates. For gender classification task, the classification rates of different algorithms are: LibSVM = 96.48%, RW-SVM1 = 96.14%, and RW-SVM2 = 96.20%. For income prediction task, the classification rates of different algorithms are: LibSVM = 84.34%, RW-SVM1 = 84.23%, and RW-SVM2 = 84.44%. This is remarkable because the proposed SVM training uses only a fraction number of training examples. Furthermore, the amount of memory to store the RW-SVM model (i.e. the support vectors) is significantly less compared to LibSVM. For examples, in gender classification, RW-SVM classifier has $82,152$ stored parameters (325 support vectors $\times$ 252 features) whereas LibSVM has $189,252$ stored parameters (751 support vectors $\times$ 252 features).

The improvement is even more significant on the income prediction task. For the reduced, weighted SVM training, the number of support vectors needed is only 1098 for RW-SVM1, and 1270 for RW-SVM2. In comparison, for standard SVM training, the SVM classifier (trained with LibSVM) requires $13,462$ support vectors to form the class boundary. For the UCI Adult Census data set, we observe that the number of samples are not balanced for the two classes, and the class ratio is roughly one-to-three. Using Method 1 of defining the custom weights, RW-SVM1 has increased the classification rate for the minority class to 59.11%, whereas standard SVM training can only achieve a CR of 55.44% for the minority class.

In term of processing speed, on the UCI Adult Census data set, the RW-SVM takes on average 291 seconds to learn the entire training data and evaluate the test data. It is 10 times faster than LibSVM.

In summary, the experimental results show that, compared to standard SVM training, the RW-SVM can achieve similar classification rates. However, RW-SVM produces a much smaller number of support vectors and takes much shorter time to train and test. In addition, RW-SVM with different methods of defining the custom weights can improve classification performances for the minority classes in applications involving class-imbalanced data.

IV. CONCLUSIONS

In this paper, we combined unsupervised and supervised learning to develop an efficient SVM training to tackle the problems of class imbalance and large scale data. Through empirical experiments, we demonstrated that traditional SVM training has difficulties in constructing an effective classifier model from large and imbalanced data sets. We showed that the reduced, weighted SVM training method can improve the classification performance and reduce the training time. It also creates a more compact classifier model, which reduces memory storage and computation time significantly. We plan to extend the reduced, weighted SVM to multi-class problems and explore new methods of defining the sample weight.

REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[2] E. Osuna, R. Freund, and F. Girosit, "Training support vector machines: an application to face detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 130–136.

[3] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

TABLE II

PERFORMANCE MEASURES ON GENDER CLASSIFICATION FERET DATA SET FOR THE PROPOSED AND STANDARD SVM TRAINING.

Training Algorithms	Number of data	(C, γ)	CR (%)	95% C.I. ($\pm$)	Number of SVs	Training time(s)	Clustering time(s)	Total time(s)
LibSVM	1408	(3.5, 0.029)	96.48	0.86	751	44.49	0	44.49
RW-SVM1	634	(8192, 0.018)	96.14	0.90	326	19.44	13.55	32.99
RW-SVM2	634	(4096, 0.018)	96.20	0.89	326	19.16	13.55	32.71

TABLE III

PERFORMANCE MEASURES ON INCOME PREDICTION UCI ADULT CENSUS DATA SET FOR THE PROPOSED AND STANDARD SVM TRAINING.

Training algorithms	(C, γ)	CR (%)	95% C.I. ($\pm$)	CR on majority	CR on minority	Number of SVs	Training time (s)	Clustering time(s)	Total time(s)
LibSVM	(3.5, 0.05)	84.34	0.335	93.86	55.44	13,462	2831	0	2831
RW-SVM1	(32768, 0.063)	84.23	0.336	92.51	59.11	1098	94.60	160	254.6
RW-SVM2	(16384, 0.25)	84.44	0.334	94.28	54.59	1270	131.2	160	291.2

[4] F. Zhou and B. Lu, “Learning concepts from large-scale data sets by pairwise coupling with probabilistic outputs,” in *Proceedings of International Joint Conference on Neural Networks*, Orlando, Florida, USA, 2007, pp. 524–529.

[5] J. C. Platt, “Fast training of support vector machines using sequential minimal optimization,” *Advances in Kernel Methods: Support Vector Learning*, pp. 185–208, 1999.

[6] S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, “Improvements to Platt’s SMO algorithm for SVM classifier design,” *Neural Computation*, vol. 13, no. 3, pp. 637–649, 2001.

[7] R. Collobert, S. Bengio, and Y. Bengio, “A parallel mixture of svms for very large scale problems,” *Neural Computation*, vol. 14, no. 5, pp. 1105–1114, 2002.

[8] A. Schwaighofer and V. Tresp, “The bayesian committee support vector machine,” in *Proceedings of the International Conference on Artificial Neural Networks*, G. Dorffner, H. Bischof, and K. Hornik, Eds. Springer Verlag, 2001, p. 411417.

[9] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[10] G. Schohn and D. Cohn, “Less is more: Active learning with support vector machines,” in *Proceedings of the 17th International Conference on Machine Learning*, 2000, pp. 839–846.

[11] M. Li and I. Sethi, “Confidence-based active learning,” *IEEE Transactions on Pattern Analysis and Machnie Intelligence*, vol. 28, no. 8, pp. 1251–1261, 2006.

[12] P. Mitra, C. Murthy, and S. Pal, “A probabilistic active support vector learning algorithm,” *IEEE Transactiosn on Pattern Analysis and Machnie Intelligence*, vol. 26, no. 3, pp. 413–418, 2004.

[13] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, “An efficient k-means clustering algorithm: analysis and implemetation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881 – 892, Jul. 2002.

[14] J. Yu, “General c-means clustering model,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, p. 1197 1211, 2005.

[15] R. Xu and D. Wunsch, “Survey of clustering algorithm,” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[16] F. Badran, M. Yacoub, and S. Thiria, “Self-organizing maps and unsupervised classification,” in *Neural Networks*. Springer Berlin Heidelberg, 2005, pp. 379–442.

[17] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi, “The feret evaluation methodology for face recognition algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1090–1104, 2000.

[18] A. Asuncion and D. Newman, “Uci machine learning repository,” 2007, http://www.ics.uci.edu/simmlearn/MLRepository.html.

[19] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” 2001, software available at http://www.csie.ntu.edu.tw/c̃jlin/libsvm.

[20] B. Moghaddam and M.-H. Yang, “Learning gender with support faces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 707–711, 2002.