International Journal of Grid and Distributed Computing
Vol. 9, No. 9 (2016), pp.129-136
http://dx.doi.org/10.14257/ijgdc.2016.9.9.12

Multiple Smooth Support Vector Machine with FCM Clustering in Hidden Space

Xian-wei Zhang[1], and *Jinjin Liang[2]

[1]*School of Computer Sciences, Xi'an Shiyou University, Xi'an 710065, China*
[2]*School of Mathematical Sciences, Xi'an Shiyou University, Xi'an 710065, China*
[1]*E-mail: xwzhang@xsyu.edu.cn,*[2]*E-mail: myonlyonly@126.com*

Abstract

A multiple smooth model is proposed by smoothing technique and piecewise technique for large scale data. Mapping the training data to the hidden space with a hidden function, the proposed model divides the original data into several subclasses by Fuzzy C Means (FCM), whose initial cluster centers are selected by samples with large density indexes; derives the smooth differentiable model by utilizing the entropy function to replace the plus function of the slack vector, and introduces linking rules to combine results of subclasses. Simulations demonstrate that the obtained algorithm maintains good classification accuracies, reduces the training time and hardly varies with kernel parameters.

Keywords*: Hidden space; smooth; piecewise; density indexes; Fuzzy C Means*

1. Introduction

Smooth support vector machine(SSVM) is a differentiable model[1], which adopts unconstrained optimization algorithms with great efficiency and attracts scholars' attentions. Some proposed models with various smooth functions [2,3], some generalized to the forecasting area[4], and some proposed the exact model [5]. The smooth algorithms have two disadvantages. One, the kernel function must satisfy the rigorous Mercer condition, which rejects some usable kernel functions. Two, long training time is needed for large scale data.

Hidden space support vector machine (HSSVM) overcomes the first disadvantage of SSVM [6], which extends the usable kernel set by mapping the input data to a hidden space with the symmetry hidden function. Researches on HSSVM include finding sparse algorithm [7] or ensemble algorithm [8], and applying it to novelty detection [9] or feature fusion area[10]. HSSVM figures out the solution to the convex quadratic program by dual technique and has two disadvantages. One, it restricts of the efficient unconstrained optimization algorithms. Two, long training time is needed for large scale data.

To construct a machine with short training time, a piecewise technique was utilized [11], whose final decision surface was constructed by some linear function in subspaces. It had strong adaptability and good classification ability, but it is only applied to SVM; also, the equidistance partitioning technique of the sample characteristic will induce certain subspace only contain one class, thus limited the accuracies and then the training space should be divided again. Wu[12] and Ye[13] proposed smooth models by replacing the plus function with a differentiable piecewise function; but it is not in accordance with the original ideology of piecewise learning.

This paper presents a smooth model in the hidden space using piecewise technique, the multiple smooth support vector machines with Fuzzy C Means clustering (MSSVM-FCM). Firstly, the data are mapped into the hidden space and then partitioned into several subclasses by FCM. Secondly, a smooth model is derived by replacing the plus function

ISSN: 2005-4262 IJGDC

with the entropy function of the slack vector, with the optimal solutions solved by Newton algorithm. Finally, linking rules are proposed to combine classification results of all the subclasses.

2 MSSVM-FCM: Multiple Smooth Support Vector Machine with FCM Clustering in Hidden Space

Let $T=(X,Y)=\{(x_i,y_i)\}_{i=1}^{l}$, where $X=\{x_i\}_{i=1}^{l}, x_i \in R^n$ is the training sample and $Y=\{y_i\}_{i=1}^{l}(y_i \in \{1,-1\})$ is the label.

2.1 Hidden Space

Let X be the independently and identical distributed data, we define a vector $\varphi(x)$ made up of a real valued function set

$$\varphi(x)=[\varphi_1(x),\varphi_2(x),\cdots,\varphi_d(x)] \tag{1}$$

The input point is mapped into a new space of dimension d

$$x \xrightarrow{\varphi} z=[\varphi_1(x),\varphi_2(x),\cdots,\varphi_d(x)] \tag{2}$$

The function set $\{\varphi_i(x)\}$ plays a role similar to a hidden unit in the forward neural networks (FNNS), and is thus referred to as "the hidden function".

The hidden space Z is defined as

$$Z=\{z \mid z=[\varphi_1(x),\varphi_2(x),\cdots,\varphi_d(x)]^T, x \in X\} \tag{3}$$

Take the symmetric function $k(x,y)=k(y,x)$ as a special kind of hidden function, the kernel mapping becomes

$$x \xrightarrow{k} z=[k(x_1,x),k(x_2,x),\cdots,k(x_l,x)] \tag{4}$$

Accordingly, the hidden space can be expressed as follows based on any symmetric kernel with dimension l

$$Z=\{z \mid z=[k(x_1,x),k(x_2,x),\cdots,k(x_l,x)]^T, x \in X\} \tag{5}$$

The commonly used kernel functions includes positive SVM kernel, such as the Polynomial kernel, Gaussian radial basis kernel and Sigmoid kernel function, as well as the compact support kernel function in (6).

$$k(x,y)=\begin{cases}\cos(p\|x-y\|^2), \\ \qquad p\|x-y\|^2 \in [-\frac{\pi}{2},\frac{\pi}{2}] \\ 0, \qquad other\end{cases} \tag{6}$$

2.2 FCM clustering

FCM is easy to implement. It is introduced to partition all the training data into several disconnected regions by a similarity measure.

Denote by $\{Z_j\}_{j=1}^{c}$ the c $(2 \le c \le l)$ fuzzy subclasses, that represent Z 's natural substructure satisfying $Z = \bigcup_{j=1}^{c} Z_j$ and $Z_i \bigcap Z_j = \phi$ $(i \ne j)$. The training of FCM equals the following

$$\min_{(U,V)} J_m(U,V;Z) = \sum_{i=1}^{c}\sum_{j=1}^{l}(u_{ij})^m d_{ij}^{\ 2}$$

$$s.t. \quad \sum_{i=1}^{c} u_{ij} = 1, \ \forall j = 1,2,\cdots,l \tag{7}$$

Here, J_m is the cost function, $V = (v_1, v_2, \cdots, v_c)$ is a vector of cluster centers, or called the prototype; $U = (u_{ij})_{c\times l}$ is the membership matrix, where $u_{ij} \in [0,1]$ describes the membership of sample x_j belonging to the cluster center v_i; $m \ge 1$ is a weighted index number controlling the fuzziness extent of clusters, the fuzziness increases with m; d_{ij} is the Euclidean distance between v_i and x_j.

$$d_{ij} = d(i,j) = \sqrt{\| x_i - v_j \|^2} \tag{8}$$

Optimal partitions U^* of Z are taken from pairs (U^*, V^*) that are local minimizes of J_m. Approximate optimization of J_m by FCM is based on iteration, according to the following two necessary conditions

$$v_i^{(r)} = \frac{\sum_{j=1}^{n}(u_{ij}^{(r)})^m z_j}{\sum_{j=1}^{n}(u_{ij}^{(r)})^m} (i = 1,2,\cdots,c) \tag{9}$$

and

$$u_{ij}^{(r+1)} = \frac{1}{\sum_{k=1}^{c}(\frac{d_{ij}}{d_{kj}})^{\frac{2}{m-1}}} \tag{10}$$

by setting partial derivative of J_m to be zero with respect to each parameter.

2.3 Multiple Smooth Models in Hidden Space

Let $\xi = (\xi_1, \xi_2, \cdots, \xi_l)$ be the slack, and let $0 < C \in R^1$ be the penalty parameter that makes good compromise between the margin and the misclassified error.

2.3.1 Smooth support vector machine: The training of smooth support vector machine (SSVM) equals the following program

$$\min \frac{1}{2}(w^T w + b^2) + C\sum\nolimits_{i=1}^{l}\xi_i^2$$

$$s.t. \ \ y_i(w^T x_i + b) \ge 1 - \xi_i$$

$$\xi_i \ge 0, i = 1,\cdots,l. \tag{11}$$

where $w \in R^n$ is the weight and $b \in R^1$ is the bias of the separating hyper plane.

Denote by $A=[x_1, x_2, \cdots, x_l]$ the matrix form of the training data, and use the entropy function

$$P_\beta(x)=x+\beta^{-1}\ln[1+\exp(-\beta x)] \tag{12}$$

to approximate the plus function of the slack vector

$$\xi=\max[e-D(Aw+be),0]=[e-D(Aw+be)]_+ \tag{13}$$

where e is a column vector of ones, D is the diagonal matrix with ones or negative ones along the diagonal corresponding to the label of x_i. The training of smooth support vector machine (SSVM) equals follows

$$\min \frac{1}{2}(w^T w+b^2)+C \| P_\beta[e-D(Aw+be)] \|^2 \tag{14}$$

and Newton algorithm is proposed to figure out the optimal solution. For more detail, refer to reference [8].

2.3.2 Multiple smooth models: The training samples are divided into c subclasses with FCM in the hidden space. For intuitive understanding, denote by $z_k(i)$ the samples of the i -th fuzzy subclass and supply the labels $y_k(i) \in \{-1,1\}$. Rank $(z_k(i), y_k(i))$ in ascending order according to the footnote k, and obtain the binary training data set

$$T_H(i)=(Z_i, Y_i)=\{(z_k(i), y_k(i))\}_{k=i_1}^{i_s} \tag{15}$$

where $Z_i=\{z_k(i)\}_{k=i_1}^{i_s}, Y_i=\{y_k(i)\}_{k=i_1}^{i_s}(i_1<i_2<\cdots<i_s)$.

The specific formulation of the hidden space is as follows

$$Z_i=\{z_k \mid z_k=[k_{(x_{i_1},x)}(i), k_{(x_{i_2},x)}(i), \cdots, k_{(x_{i_s},x)}(i)]^T, x \in X_i\} \tag{16}$$

Here $X_i=\{x_k(i)\}_{k=i_1}^{i_s}$ is the training data, and $k_{(x_p,x_q)}(i)=\phi(x_p)^T\phi(x_q)$ is the hidden function.

Denote by $\xi(i)=\{\xi_k(i)\}_{k=i_1}^{i_s}(0 \le \xi_k(i) \in R^1)$ the slack to $x_k(i)$, and denote by $0<C(i) \in R^1$ the penalty proportional to the amount of slacks $\sum_{k=i_1}^{i_s}\xi_k^2(i)$; the smooth model is

$$\min \frac{1}{2}[w(i)^T w(i)+b(i)^2]+C(i) \| P_\beta\{e(i)-D(i)[A(i)w(i)+b(i)e(i)]\} \|^2 \tag{17}$$

The gradient vector and the Hessian matrix are computed as

$$\nabla F_\beta\begin{pmatrix} w(i) \\ b(i) \end{pmatrix}=\frac{1}{2C(i)}\begin{bmatrix} w(i) \\ b(i) \end{bmatrix}+\begin{bmatrix} -Q(i)^T \\ -Y(i)^T \end{bmatrix} Diag\{P_\beta[t(i)]\}Diag[e(i)+v(i))^{-1}]e(i) \tag{18}$$

$$\nabla^2 F_\beta\begin{pmatrix} w(i) \\ b(i) \end{pmatrix}=\frac{1}{2C(i)}\begin{bmatrix} I(i) & 0 \\ 0 & 1 \end{bmatrix}+2\begin{bmatrix} -Q^T(i) \\ -Y^T(i) \end{bmatrix} Diag[e(i)+v(i)]^{-2}M[-Q(i)-Y(i)] \tag{19}$$

where $Q(i)=D(i)A(i)$, $Y(i)=D(i)e(i)$, $t(i)=e(i)-D(i)[A(i)w(i)-e(i)b(i)]$, $v(i)=\exp[-\beta t(i)]$ and $M=$ $\{I(i)+\beta Diag[(v(i)]\}\cdot Diag\{P_\beta[t(i)]\}$.

Evidently the Hessian matrix (19) is positive definite, so the program (17) is convex and has unique solution.

2.4 Linking Rule

For the resulted c decision functions, one for each subclass, how to make reasonable use of them is an important procedure to predict the label of any test data. A natural way is considering results of all the subclasses and making compromise among them.

For the test data x, we use decision function of each subclass to predict its label. If the majority of the decision functions judge that the test data belongs to the positive class, the label is set to be positive one; otherwise the label is set to be negative one.

The decision function is as follows

$$y = \operatorname{sgn} \sum_{i=1}^{c} y_x(i) = \operatorname{sgn}\{\sum_{i=1}^{c} \operatorname{sgn}[g_x(i)]\} \tag{20}$$

3. Experiments and Comparisons

The effectiveness of MSSVM-FCM is demonstrated now on artificial and benchmark datasets. All the experiments are carried out on a PC with P4 CPU, 3.06 GHz, 1GB Memory. The programs are written in pure MATALAB 7.01 Language.

3.1 Performances Variances with the Class Number

Generate 300 binary-classification normal distribution data by five normal distribution clusters, in which the positive and negative data are of equal sizes and 5% labels are changed to make overlap. The training process is carried out on all the 300 normal distribution data, and so is the testing process.

To facilitate comparison, same parameters are adopted for various algorithms: the penalty is selected as $C = 1$, the symmetry positive definite Gaussian radial basis kernel function $k(x, y) = \exp(-\| x - y \|^2 / \sigma^2)$ is used with kernel width $\sigma = 0.1$ .The weighting rule is adopted. The performances are illustrated in table 1 under different values of class number c ; the time and accuracy are the training time and training accuracies on the whole data, $c = 1$ means that no "FCM" is applied before training.

Table 1. Performances with Various Algorithms

Algorithm		**Accuracies**	**Time**
SVM		94.76%	6.51s
SSVM		94.53%	3.18s
HSSVM		93.91%	7.23s
MSSVM-FCM	$c = 1$	94.72%	3.15s
	$c = 2$	94.96%	2.33s
	$c = 5$	95.69%	1.17s
	$c = 8$	94.12%	0.57s

Data in the table 1 lead to the following conclusions.

(1) MSSVM-FCM has the highest training accuracies, which vary slightly with the class number.

MSSVM-FCM has accuracies higher than 94%, which are identical with those of SVM, SSVM and HSSVM. It achieves the highest accuracy 95.69% at $c = 5$, which is 0.73, 0.97 and 3.57 percent higher than that at $c = 1$, $c = 2$ and $c = 8$; the maximum magnitude is not larger than 3.57%.

(2) MSSVM-FCM has the lowest training time, which varies significant slightly with the class number.

At $c=1$, MSSVM-FCM has a training time of 3.15s which is respectively 48.38% and 43.56% of SVM and HSSVM, and is 0.03 higher than SSVM. At $c=8$, MSSVM-FCM has a training time of 0.57s, which is respectively 7.88%, 8.75% and 17.92% of HSSVM, SVM and SSVM. The results testify that, figuring out the optimal solution using smoothing technique is more convenient and effective than traditional dual technique.

MSSVM-FCM obtains the lowest training time 0.57s at $c=8$, which is 18.09%, 24.46% and 48.72% of that at $c=1$, $c=2$ and $c=5$. In other words, the training time decreases greatly with the increase of c ; the variance is significant.

3.2 Performances Variances with the Kernel Width

Breast Cancer data is composed of 458 "benign" examples and 241 "malignant" examples, nine attributes for each sample. Set $C=1$ for all algorithms, the accuracies of MSSVM-FCM is compared with SVM, SSVM and HSSVM under Gaussian kernel width σ. Fifty percent are randomly selected as the training set, leaving the others as the testing set. Performances are illustrated in table 4 of ten randomly sampling.

Table 2. Accuracies with Kernel Width

Kernel width	Accuracies			
	SVM	SSVM	HSSVM	MSSVM-FCM
$\sigma=0.01$	66.76%	63.59%	60.37%	93.26%
$\sigma=0.2$	89.93%	86.33%	84.29%	93.75%
$\sigma=0.5$	93.22%	93.20%	93.09%	96.33%
$\sigma=0.8$	91.08%	91.71%	90.30%	96.37%
$\sigma=1$	89.81%	91.06%	85.73%	93.96%

Based on the above table, we draw the following conclusions.

(1) MSSVM-FCM has the highest accuracies.

Take $\sigma=0.01$ for example; MSSVM-FCM has a training accuracy of about 24.66%, 27.83% and 31.05% higher than SVM, SSVM and HSSVM with weighting rule; while it has a training accuracy of 26.5%, 29.67% and 32.89% higher than SVM, SSVM and HSSVM.

(2) MSSVM-FCM is insensitive to kernel width, while SVM, SSVM and HSSVM have obvious variances with kernel width.

At $\sigma=0.01$ and $\sigma=0.5$, MSSVM-FCM has a training accuracy of 91.42% and and 95.08% with weighting rule, while it has a training accuracy of 93.26% and 96.33% with absorbing rule 3. Increasing the kernel width from $\sigma=0.5$ to $\sigma=1$, the accuracies of MSSVM-FCM firstly increases and then decreases, but the variances range does not exceed 1.6%.

At $\sigma=0.01$, $\sigma=0.5$ and $\sigma=1$, SVM has a training accuracy of 66.76%, 93.22% and 89.81%. SVM has an accuracy increase of about 26.46% as the kernel width is increased from $\sigma=0.01$ to $\sigma=0.5$, and has an accuracy decrease of about 3.41% as the kernel width is increased to $\sigma=1$. The accuracies variances are significant.

Same results apply for SSVM and HSSVM by simple comparisons. In other words, the accuracies increase proportional with the kernel width until σ reaches a certain value and then decrease.

3.3 Performances on Large Scale Data

The number of subclasses c is determined by the validity function (19). The absorbing rule 3 is used since it has been proved to have better performances in former experiments. The Gaussian radial basis kernel is used throughout the section. The penalty parameter and the kernel width parameter are determined by the ten-fold cross validation.

The first experiment is carried out on two dimensional normal distribution data, whose training sizes vary from 200 to 2000, the mean is zero point four and variance is zero point six. Select the penalty from $C \in \{0.01, 0.1, 0.3, 0.5, 0.8, 1\}$ and the kernel width from $\sigma \in \{0.01, 0.1, 0.3, 0.5, 0.8, 1\}$, MSSVM-FCM is compared with SVM, SSVM and HSSVM. All the algorithms achieve accuracies higher than 95%, but their training time differ greatly with the kernel width as is illustrated in Figure 3.

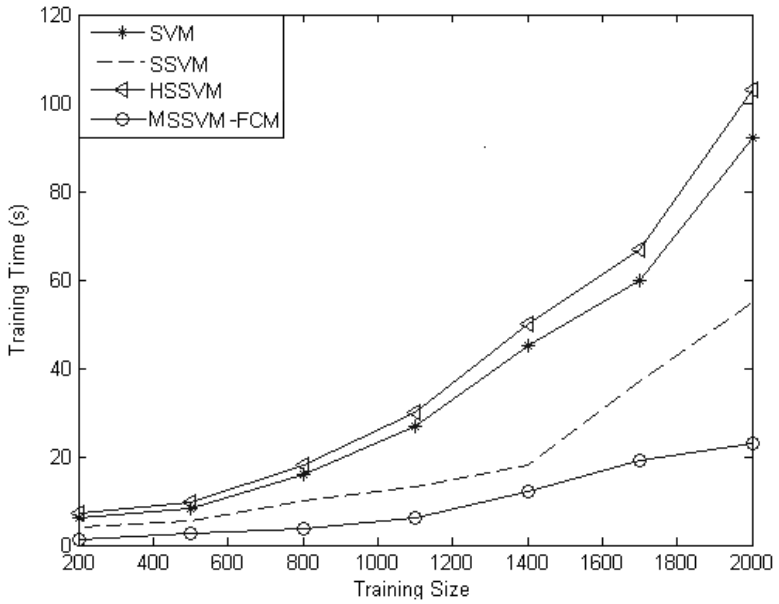


Figure 1. Training Time of Various Algorithms

It is clear that, MSSVM-FCM has the lowest training time; it decreases greatly with the training size.

Significantly, MSSVM-FCM has the highest accuracies as well as the lowest training time and iteration. The decrease of the training time is more obvious on large scale data. When the training size is 1000, MSSVM-FCM has a training time 47.37% of SSVM, 20.81% of HSSVM, 41.45% of SHSSVM and 52.68% SDWNSVM. When the training size is 5000, MSSVM-FCM has a training time 25.56% of SSVM, 5.9% of HSSVM, 26.62% of SHSSVM and 28.26% SDWNSVM.

4. Conclusions

This paper deals with multiple smooth model in the hidden space. MSSVM-FCM broadens the usable kernel functions by mapping all the data to the hidden space; it has short training time due to the training on the divided subclasses by FCM piecewise technique; it has satisfying accuracies as well as good robustness that vary little with the kernel width. Simulations on various data demonstrate the effectiveness. Future work

includes designing new piecewise technique to partition the samples, or exploiting new linking rules to combine results of subclasses.

Acknowledgment

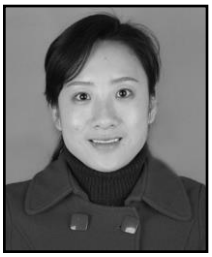
The Work is supported by the Special Research Program of Shaanxi Provincial Department of Education (No. 15JK87).

References

[1] Lee YuhJye, Mangasarian O. L.. SSVM: a smooth support vector machine for classification[J]. Computational Optimization and Applications, 2001, 22(1): 5-21.
[2] Qin Chuandong, Liu Sanyang. Fuzzy smooth support vector machine with different smooth functions[J]. Journal of Systems Engineering and Electronics, 2012, 23(3): 460-466
[3] Shen Jindong, Peng Xiaojun. A New smooth support vector machine with 1-norm penalty term [J]. International Journal of Online Engineering, 2013, 9(4): 54-58.
[4] Yuan Yubo. Forecasting the movement direction of exchange rate with polynomial smooth support vector machine [J]. Mathematical and Computer Modelling, 2013, 57(3-4): 932-944.
[5] Liang Jinjin, Wu De. Smooth diagonal weighted newton support vector machine [J]. Mathematical Problems in Engineering, 2013, v2013.
[6] Zhang Li, Zhou Weida, Jiao Licheng. Hidden space support vector machines [J]. IEEE Transactions on Neural Networks, 2004, 15(6): 1424-1434.
[7] Wang, Ling; Bo, Liefeng; Liu Fang; Jiao Licheng. Sparse hidden space support vector machine[J]. Journal of Xidian University, 2006, 33(6): 896-901.
[8] Dhanalakshmi, P.; Palanivel, S.; Ramalingam, V. Classification of audio signals using SVM and RBFNN[J]. Expert Systems with Applications, 2009, 36(3): 6069-6075.
[9] Zhang Li, Wang Bangjun, Li Fanzhang, *et al.* Support vector novelty detection in hidden space [J]. Journal of Computational Information Systems, 2011, 7(15): 5581-5590.
[10] Cheng Jun, Xie Can, Bian Wei, *et al.* Feature fusion for 3D hand gesture recognition by learning a shared hidden space[J]. Pattern Recognition Letters, 2012, 33(4): 476-484.
[11] Ren Shuangqiao, Yang degui, Li Xiang, Zhuang Zhaowen. Piecewise support vector machines [J]. Chinese Journal of Computers, 2009, 32(1): 77-85.
[12] Wu Qing, Wang Wenqing. Piecewise smooth support vector machine for classification[J]. Mathematical problems in Engineering, 2013, v2013.
[13] Ye Qixiang, Han Zhenjun, Jiao Jianbin, *et al.* Human detection in images via piecewise linear support vector machines [J]. IEEE Transactions on Image Processing, 2013, 22(2): 778-789.

Author

Xian wei Zhang, he was born in 1973. He received his Bachelor's degree from Xidian University in 2000. He obtained his Master's degree from the same school in 2005. He obtained his Doctor's degree in 2013 from Xian University of science and technology. He is now a senior engineer in School of computer sciences in Xi'an Shiyou University. His main research areas include Grid computing and Cloud computing.



Jin jin Liang, was born in 1983. She received her B. E. degree in applied mathematics from Hernan Normal University in 2004. She obtains her Ph. D. degree in applied mathematics in Xidian University. She is now a lecture in Xi'an Shiyou University. Her main research areas are optimization methods with applications, data mining and support vector machine.

