

# Hidden Space Smooth Support Vector Machine with C Means Clustering

Jinjin Liang<sup>1, a</sup>, Wenhao Xie<sup>1, b</sup> and Xiaoyan Wang<sup>1, c</sup>

<sup>1</sup> School of Mathematical Sciences, Xi'an Shiyou University, Xi'an 710065, China

<sup>a</sup>myonlyonly@qq.com, <sup>b</sup>1609632928@qq.com, <sup>c</sup>18512262@qq.com

**Keywords:** Hidden Space; Smooth technique; Support Vector Machine; FCM

**Abstract.** A piecewise smooth model is proposed and called HSSVM-CM for short in the hidden space. Mapping the training data to the hidden space with a hidden function, HSSVM-CM divides the original data into several subclasses by C means; derives the smooth differentiable unconstrained model by utilizing the entropy function to approximate the plus function of the slack vector, and introduces linking rules to combine classification results of various subclasses. Simulations on benchmark data demonstrate that HSSVM-CM maintains good classification accuracies, reduces the training time and hardly varies with kernel parameters.

## Introduction

Support vector machine (SVM)[1] is developed as an prominent method for the small-sample data and has gained wide attention. However, the kernel must satisfy the rigorous Mercer condition; only the symmetric and positive definite function can be used as the kernel which of course rejects some usable functions, such as the compact support kernel function.

Hidden space support vector machine (HSSVM) extends the set of usable kernels [2], which only requires the hidden function to be symmetry. Researches on HSSVM include constructing sparse or ensemble algorithms [3-4], applying it new area [5]. However, HSSVM needs to solve a constrained program and requires long training time for large scale data.

Smooth support vector machine (SSVM) has mathematical properties such as strong convexity and infinitely often differentiability, which attracts many scholars to research from various aspects [6-8]. But they can not get rid of the restriction on kernel function to be positive definite.

Inspired by the advantages of HSSVM and smoothing techniques [9-10], this paper presents a new smooth model called HSSVM-CM in hidden space with C Means clustering. Firstly, the training data are mapped into the hidden space and then partitioned into several subclasses by CM. Secondly, a smooth support vector machine model is derived by replacing the plus function with the entropy function of the slack vector. Finally, linking rules are proposed to combine classification results of all the subclasses.

## Introduction of Hidden Space

Let  $T=(X,Y)=\{(x_i,y_i)\}_{i=1}^l$ , with  $X=\{x_i\}_{i=1}^l (x_i \in R^n)$  is the independently and identical distributed data and  $Y=\{y_i\}_{i=1}^l (y_i \in \{1,-1\})$  is the label. Define a vector  $j(x)$  made up of a real valued function set

$$j(x)=[j_1(x), j_2(x), \mathbf{L}, j_d(x)] \quad (1)$$

The hidden space  $Z$  is defined as

$$Z=\{z|z=[j_1(x), j_2(x), \mathbf{L}, j_d(x)]^T, x \in X\} \quad (2)$$

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

$$x \xrightarrow{k} z=[k(x_1,x), k(x_2,x), \mathbf{L}, k(x_l,x)] \quad (3)$$

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

$$Z = \{z \mid z = [k(x_1, x), k(x_2, x), \mathbf{L}, k(x_l, x)]^T, x \in X\} \quad (4)$$

**Lemma 1** The distance between  $x$  and  $y$  in the hidden space is computed as follows

$$d_H(x, y) = \|x - y\|_H = \sqrt{\sum_{t=1}^l [k(x_t, x) - k(x_t, y)]^2} \quad (5)$$

**Proof:** The transformed formulation of  $x$  and  $y$  are  $z(x) = [k(x_1, x), k(x_2, x), \mathbf{L}, k(x_l, x)]$  and  $z(y) = [k(x_1, y), k(x_2, y), \mathbf{L}, k(x_l, y)]$ , where  $k(x, y)$  is the symmetric hidden function.

The square of the distance between them is

$$\begin{aligned} d_H^2(x, y) &= \|x - y\|_H^2 = \\ &= \|z(x) - z(y)\|^2 = \\ &= \{[k(x_1, x) - k(x_1, y)]^2 + [k(x_2, x) - k(x_2, y)]^2 + \mathbf{L} + [k(x_l, x) - k(x_l, y)]^2\}^{\frac{1}{2}} = \\ &= \sum_{t=1}^l [k(x_t, x) - k(x_t, y)]^2 \end{aligned} \quad (6)$$

Thus, we complete the proof.

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.

$$k(x, y) = [(x^T y) + 1]^q, q \in N \quad (7)$$

$$k(x, y) = \exp(-\|x - y\|^2 / S^2) \quad (8)$$

$$k(x, y) = \tanh(p_1 \cdot x^T y + p_2), p_1, p_2 \in R \quad (9)$$

$$k(x, y) = \begin{cases} \cos(p \|x - y\|^2), \\ p \|x - y\|^2 \in [-\frac{p}{2}, \frac{p}{2}] \\ 0, & other \end{cases} \quad (10)$$

## Smooth Support Vector Machine

Let  $x = (x_1, x_2, \mathbf{L}, x_l)$  be the slack, and  $0 < C \in R^1$  be the penalty parameter that makes compromise between the margin and the misclassified error. The training of smooth support vector machine (SSVM) equals the following program

$$\begin{aligned} \min & \frac{1}{2}(w^T w + b^2) + C \sum_{i=1}^l x_i^2 \\ \text{s.t.} & y_i(w^T x_i + b) \geq 1 - x_i \\ & x_i \geq 0, i = 1, \dots, l. \end{aligned} \quad (11)$$

In equation (11),  $w \in R^n$  is the weight and  $b \in R^1$  is the bias of the separating hyper plane

$$g(x) = w^T x + b = 0 \quad (12)$$

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

$$P_b(x) = x + b^{-1} \ln[1 + \exp(-bx)] \quad (13)$$

to approximate the plus function of the slack vector

$$x = \max[e - D(Aw + be), 0] = [e - D(Aw + be)]_+$$

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 SSVM equals follows

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

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

### Implementation of HSSVM-CM

HSSVM-CM firstly maps the training data  $T = (X, Y)$  into the hidden space  $T_H = (Z, Y)$ , employs CM to divide  $Z$  into  $c$  disconnected regions  $\{Z_i\}_{i=1}^c$ , derives smooth models in each divided space, and finally uses linking rules to predict label of test data.

HSSVM-CM consists of three steps, mapping to the hidden space, smooth support vector machine training, and linking rules. The principle of HSSVM-CM is illustrated in Figure 1.

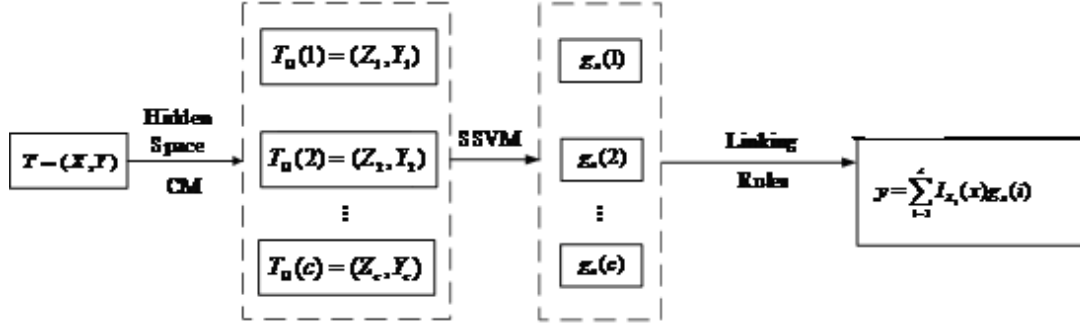


Figure 1 Classification principle of HSSVM-CM

### C Means Clustering

C Means clustering is used to partition all the training data into several disconnected regions. Denote by  $\{Z_j\}_{j=1}^c$  the  $c$  ( $2 \leq c \leq l$ ) fuzzy subclasses, that represent  $Z$ 's natural substructure satisfying

$Z = \bigcup_{j=1}^c Z_j$  and  $Z_i \cap Z_j = \emptyset$  ( $i \neq j$ ). The training of CM equals the following

$$\begin{aligned} \min P(U, Z) &= \sum_{i=1}^c \sum_{x_k \in X_i} u_{ij} d_{ij}^2 \\ \text{s.t. } \sum_{i=1}^c u_{ij} &= 1, \forall j = 1, 2, \dots, n \end{aligned} \quad (15)$$

Here,  $P(U, Z)$  is the cost function,  $Z = (v_1, v_2, \dots, v_c)$  is a vector of cluster centers, or called the prototype;  $U = (u_{ij})_{c \times n}$  is the membership matrix, where  $u_{ij} = 0$  indicates that sample  $x_j$  doesn't belong to the cluster center  $v_i$ , while  $u_{ij} = 1$  indicates that sample  $x_j$  belongs to the cluster center  $v_i$ ;  $d_{ij}$  is the Euclidean distance between  $v_i$  and  $x_j$ .

$$d_{ij} = d(i, j) = \sqrt{\|x_i - v_j\|^2} \quad (16)$$

For the selected clustering centers, CM arranges the training data to its nearest cluster, then computes the center of the new cluster, and repeat the iteration until the objective function achieves its minimum.

### 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.

**Step 0** Initialization. Input the test data  $x$  and the obtained the hyper plane of each subclass.

**Step 1** Judgment. Use the following rule to judge the label

$$y = \sum_{i=1}^c I_{Z_i}(x) g(x) \quad (17)$$

where  $g_x(i)$  is the decision function  $g(x) = \text{sign}(w^T x + b)$  and  $I_{Z_i}(x)$  is the indicative function.

$$I_{Z_i}(x) = \begin{cases} 1, & z(x) \in Z_i \\ 0, & z(x) \notin Z_i \end{cases} \quad (18)$$

The footnote of  $Z_i$  is computed with

$$i = \arg(\min d_j) \quad (19)$$

where

$$d_j = \|x - v_j\|_H = \sqrt{\sum_{t=1}^l [k(x_t, x) - k(x_t, v_j)]^2} \quad (20)$$

and then classify the test data  $x$  to the  $i$ -th fuzzy group  $Z_i$  and adopt the decision function.

### Complexity analysis

For the  $l$  training data, SVM, SSVM, HSSVM and SDWNSVM all train on the whole data; they have space complexity of  $O(l^2)$ . HSSVM-CM divides the training data into  $c$  regions, computes the solution on each individual subclass and combines their results. Suppose the number of samples is  $\frac{l}{c}$  in

each subclass, HSSVM-CM has a total space complexity of  $O(\frac{l}{c^2})$ . Since  $O(\frac{l}{c^2}) \ll O(l^2)$ .

HSSVM-CM has much lower space complexity than the aforementioned algorithms. It is expected that HSSVM-CM has low training time and is more suitable for large scale data.

### Experiments and comparisons

Effectiveness of MSSVM-CM is demonstrated on UCI datasets. All the experiments are carried out on a PC with P4 CPU, 3.06 GHz, 1GB Memory with pure MATLAB Language.

**Performances variances with the kernel width** Breast Cancer is composed of 458 “benign” and 241 “malignant” examples with nine attributes. Set  $C = 1$  for all algorithms, HSSVM-CM is compared with SVM, SSVM and HSSVM with Gaussian kernel. Fifty percent are randomly selected as the training set, leaving the others as the testing set. Performances are illustrated in table 1.

**Table 1 Accuracies with kernel width**

Kernel width	SVM	SSVM	HSSVM	HSSVM-CM
$s = 0.01$	66.76 %	63.59 %	60.37%	<b>93.26%</b>
$s = 0.2$	89.93 %	86.33 %	84.29%	<b>93.75%</b>
$s = 0.5$	93.22 %	93.20 %	93.09%	<b>96.33%</b>
$s = 0.8$	91.08 %	91.71 %	90.30%	<b>96.37%</b>
$s = 1$	89.81 %	91.06 %	85.73%	<b>93.96%</b>

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

HSSVM-CM has the highest accuracies that is insensitive to kernel width, while SVM, SSVM and HSSVM have obvious variances with kernel width.

**Performances on moderate scale data** Banana data consists of 400 training and 4900 testing data, two attributes for each sample. HSSVM-CM is compared with SSVM, HSSVM and SHSSVM, where the parameters are set  $C = 1$ ,  $s = 0.5$  and  $p = 0.1$ , “CS” and “RBF” are respectively the short for “compact support kernel function” and “Gaussian radial basis kernel function”, the accuracy is the averaged value of the accuracies on the training and testing set, the blank means that it cannot be computed. The performances are illustrated in table 2.

**Table 2 Comparisons with various algorithms**

Algorithm	Ker.	Accuracies	Time	Iteration
SSVM	RBF	89.67%	2.59s	3.1
	CS	\	\	\
HSSVM	RBF	88.82%	14.66s	\
	CS	90.71%	16.18s	\
SHSSVM	RBF	89.81%	2.93s	3.2
	CS	90.92%	3.01s	3.1
HSSVM-C M	RBF	92.18%	1.21s	1.9
	CS	93.33%	1.16s	2.0

The following conclusions are directly drawn from three aspects based on data in the above table.

(1) HSSVM-CM has the highest accuracies, possesses the lowest training time under positive definite kernel and the symmetric kernel, and has the least iteration among the three smooth algorithms, SSVM, SHSSVM and HSSVM-CM.

(2) HSSVM-CM has higher training accuracies, lower time and identical iteration with RBF kernel than CS kernel.

We also carry out the experiment on Iris data. The results are illustrated in table 3, where the best results are emphasized in bold.

**Table3 Comparisons with various algorithms**

Algorithm	Ker.	Accuracies	Time	Iteration
SSVM	RBF	96.67%	0.46s	2.9
	CS	\	\	\
HSSVM	RBF	95.47%	2.62s	\
	CS	95.55%	2.67s	\
SHSSVM	RBF	96.81%	0.46s	3.2
	CS	97.13%	0.46s	2.9
HSSVM-C M	RBF	<b>99.73%</b>	<b>0.21s</b>	<b>1.1</b>
	CS	<b>99.96%</b>	<b>0.17s</b>	<b>1.3</b>

It should be noted that the blank “\” means that it cannot be computed, since SSVM and HSSVM cannot use the non-positive definite kernel. It is obvious to see that, HSSVM-CM has the highest accuracies, lowest training time and least iterations among the four algorithms.

## Conclusions

HSSVM-CM broadens the usable kernel functions, has short training time, high accuracies and good robustness. Future work includes producing new piecewise technique to partition the samples, exploiting new linking rules to combine results of subclasses or finding efficient ways to set optimal value of parameters.

## Acknowledgements

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

## References

- [1] Cristianini N., ShaweTaylor J. An Introduction to Support Vector Machines[M]. Cambridge University Press, 2000.

- [2] Zhang Li, Zhou Weida, Jiao Licheng. Hidden space support vector machines [J]. IEEE Transactions on Neural Networks, 2004, 15(6): 1424-1434.
- [3] Wang, Ling; Bo, Liefeng; Liu Fang; Jiao Licheng. Sparse hidden space support vector machine[J]. Journal of Xidian University, 2006, 33(6): 896-901.
- [4] 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.
- [5] 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.
- [6] Wu Qing, Wang Wenqing. Piecewise smooth support vector machine for classification[J]. Mathematical problems in Engineering, 2013, v2013.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] Liang Jinjin, Wu De. A hidden space smooth support vector machine with particle swarm optimization [J]. Hsi-An Chiao Tung Ta Hsueh, 2013, 47 (12), 38-42.