Communications of the Korean Statistical Society
2009, Vol. 16, No. 5, 881–889

Visualizing SVM Classification in Reduced Dimensions

Myung-Hoe Huh[1,a], Hee Man Park[a]

[a]Department of Statistics, Korea University

Abstract

Support vector machines(SVMs) are known as flexible and efficient classifier of multivariate observations, producing a hyperplane or hyperdimensional curved surface in multidimensional feature space that best separates training samples by known groups. As various methodological extensions are made for SVM classifiers in recent years, it becomes more difficult to understand the constructed model intuitively. The aim of this paper is to visualize various SVM classifications tuned by several parameters in reduced dimensions, so that data analysts secure the tangible image of the products that the machine made.

Keywords: Support vector machine(SVM), dimensional reduction, model visualization.

1. Background and Aim

Suppose that we have n observations $(x_1, y_1), \ldots, (x_n, y_n)$ each of which belongs to one of two classes, Class 1 or Class 0, where x_i's are $p \times 1$ vectors of feature variables on the continuous scale and y_i's are either 1 (for Class 1) or -1 (for Class 0) according to the associated group. Support vector machines(SVMs) are known as very flexible and efficient classifier of multivariate observations into membership groups (Vapnik, 1999; Hastie *et al.*, 2001). SVM classifiers are constructed by finding a hyperplane in the given feature space or in the transformed feature space that separates two groups of observations by the maximal margin.

We start with the linear SVM classifier, which predicts unclassified unit x to Class 1 if $f_L(x) \geq 0$, or Class 0 otherwise, where

$$f_L(x) = w^t x + b, \tag{1.1}$$

x is a $p \times 1$ feature vector, w is a $p \times 1$ vector of coefficients and b is a constant. In the linear SVM classifier, w and b are found by solving

$$\min_{w,b} \frac{\|w\|^2}{2} + C \sum_{i=1}^{n} \xi_i, \quad \text{subject to } y_i(w^t x_i + b) \geq 1 - \xi_i,\ i = 1, \ldots, n, \tag{1.2}$$

where $\xi_1 \geq 0, \xi_2 \geq 0, \ldots, \xi_n \geq 0$ are slacks and C is a parameter that affects the cost due to positive slacks. Using Lagrange multipliers $\lambda_1 \geq 0, \ldots, \lambda_n \geq 0$ attached to n constraints in (1.2), w can be expressed as

$$w = \sum_{i=1}^{n} \lambda_i y_i x_i.$$

Hence the classification (1.1) is determined by a subset of observations (x_i, y_i) with $\lambda_i > 0$, called by "support vectors".

[1] Corresponding Author: Professor, Department of Statistics, Korea University, Anam-Dong 5-1, Sungbuk-Gu, Seoul 136-701, Korea. E-mail: stat420@korea.ac.kr

The linear SVM has been extended to a more flexible nonlinear machine that predicts unclassified unit x to $y = 1$ if $f_N(x) \geq 0$, or -1 otherwise, where

$$f_N(x) = w^t \Phi(x) + b, \tag{1.3}$$

where $\Phi(x)$ is a transform of x. Also in the nonlinear SVM classifier, w and b are found by solving

$$\min_{w,b} \frac{\|w\|^2}{2} + C \sum_{i=1}^{n} \xi_i, \quad \text{subject to } y_i(w^t \Phi(x_i) + b) \geq 1 - \xi_i,\ i = 1, \ldots, n.$$

Then, by the "kernel trick", the classification function (1.3) is simplified to

$$f_N(x) = \sum_{i=1}^{n} \lambda_i y_i K(x_i, x) + b, \tag{1.4}$$

where typical kernel functions $K(x, x')$ are

Linear: $K(x, x') = x^t x'$,

The d^{th} degree polynomial: $K(x, x') = (x^t x' + 1)^d$,

Gaussian radial basis function(RBF): $K(x, x') = \exp(-\gamma \| x - x' \|^2)$, $\gamma > 0$,

Sigmoid: $K(x, x') = \tanh(\gamma x^t x' + c)$, $\gamma > 0$, $c < 0$.

Several researchers proposed visualization methods of SVM or nonlinear classifications. For instance, Wickham *et al.* (2006) produced R package `explore` linked to `GGobi` to visualize separating manifolds dynamically on the two-dimensional subspace. Also, Huh and Lee (2008) and Huh (2009) proposed the so-called conditional predictive graphs which contain a number of traces of the classification function for one explanatory variable varying on the prediction interval with the other $p - 1$ variables fixed at observed values.

The aim of this paper is to visualize various SVM classifications tuned by several parameters in reduced dimensions, so that data analysts secure the tangible image of the products that the machine made without much effort of their side. However, this study does not intend to cover optimal choice of the tuning parameters in SVM, so that the parameters adopted in numerical examples of this paper are chosen arbitrarily. Nevertheless, visual displays may be useful in judging whether the specific SVM classification is over- or under-fitted.

In Section 2, we visualize linear SVM classifications (1.1) constructed from Monte-Carlo and real datasets in 2D subspace. In Section 3, we visualize nonlinear classifications (1.4) constructed from Monte-Carlo and real datasets in reduced dimensions. In Section 4, we generalize our methods to the case of three or more classes.

2. Visualizing Linear SVM Classifications

In linear SVM classifications, the classification boundary is specified as

$$f_L(x) = 0 \quad \text{or} \quad w^t x + b = 0,$$

which is perpendicular to $p \times 1$ unit vector $w/\|w\|$ $(= v^{[1]})$ in p-dimensional feature space. Hence, to visualize predicted class labels effectively at n observed units, we propose to plot dots at $v^{[1]t} x_1, \ldots, v^{[1]t} x_n$ in different colors depending on the predicted class ("blue" for Class 1 and "red" for Class 0).

Since two-dimensional graph can be drawn without much additional effort, we add the second axis, which is orthogonal to the first axis determined by w or $v^{[1]}$. Therefore we proceed as follows.

First, project $p\times 1$ vectors $x_1, \ldots, x_n$ on the directional vector $v^{[1]}$ $(= w/\| w \|)$ and compute residual vectors $x_1^{[1]}, \ldots, x_n^{[1]}$. That is,

$$x_i^{[1]} = x_i - v^{[1]} v^{[1]t} x_i, \quad i = 1, \ldots, n.$$

Here we assume that the rows of $X^t = (x_1, \ldots, x_n)$ are standardized to have mean 0 and standard deviation 1.

Second, compute the principal component direction vector $v^{[2]}$ that carries maximally dispersed projections of $x_1^{[1]}, \ldots, x_n^{[1]}$. That is, $v^{[2]}$ is the eigenvector corresponding to the largest eigenvalue of $X^{[1]t} X^{[1]}$, where $X^{[1]t} = (x_1^{[1]}, \ldots, x_n^{[1]})$.

Third, n observations are dotted at

$$z_i^{[1]} = v^{[1]t} x_i, \qquad z_i^{[2]} = v^{[2]t} x_i \quad (i = 1, \ldots, n),$$

on the first and second axis, respectively, in different colors depending on their membership class ("blue" for Class 1 or "red" for Class 0). Accordingly, on the first and second axis, the feature variables are plotted with arrow heads at p components of

$$y^{[1]} = v^{[1]}, \qquad y^{[2]} = v^{[2]}.$$

On the 2D projection plane, we may superimpose probability contours as follows. For $u^t = (u_1, u_2)$ of the 2D projection plane, define the inverse projection by $x = u_1 v^{[1]} + u_2 v^{[2]}$ and attach to x the probability $p_x(1)$ being classified to Class 1. Then, we superimpose the probability contours to the 2D display of SVM classification.

For the propensity probability, we simply adopted the `libsvm` (Chang and Lin, 2001)'s `probability`, which is known as the implementation of Platt (2000) and Wu *et al.* (2004). Recently, Wang *et al.* (2008) improved the method for computing propensity probability following SVM classification.

Example 1. **(Monte-Carlo Simulated Dataset)** This simulation study for two class problem, that will be continued in section 3 and modified in Section 4, is intended to show the effectiveness of the reduced dimensional display of SVM classification.

Four hundred (X_1, X_2) observations are generated uniformly on the inside of 2D-sphere with radius 2, and the third variable X_3 is independently generated from Uniform $(-2, 2)$:

$$x_1^2 + x_2^2 \le 4, \quad -2 \le x_3 \le 2.$$

Class labels 1 or 0 are assigned to (x_1, x_2, x_3)'s by the following rule.

If $(x_1 - 1)^2 + x_2^2 \le 1$, it is assigned to Class 1.

Elsewhere, it is assigned to Class 0.

Figure 1 visualizes the linear SVM classification ($C = 1$) constructed from the simulated dataset. The observation plot(left) with the classification boundary reveals the poor performance of the linear classification, while the variable plot(right) shows that X_1 is more important than X_2 or X_3 in determining the classification. We will present a better classification in the next section.

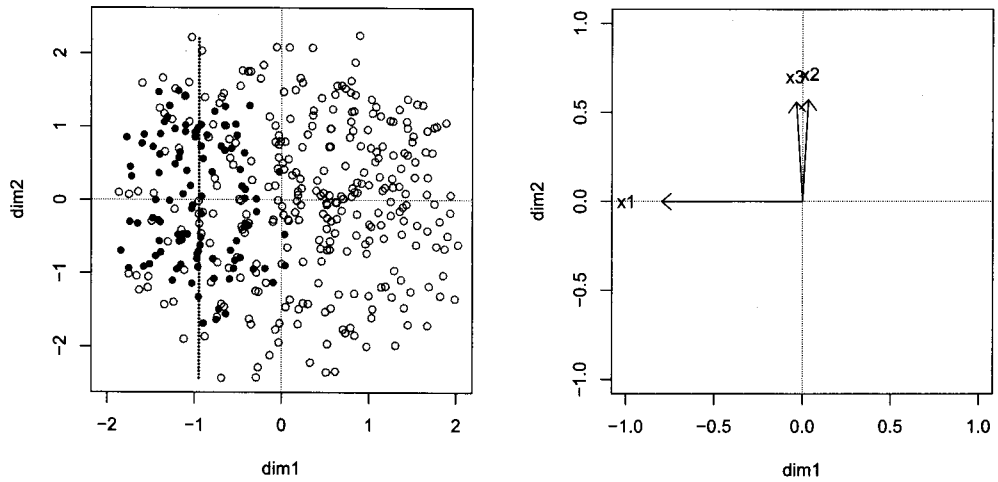

Figure 1: *Visualization of the linear SVM classification for the Monte-Carlo dataset. The dotted line is the classification boundary.*

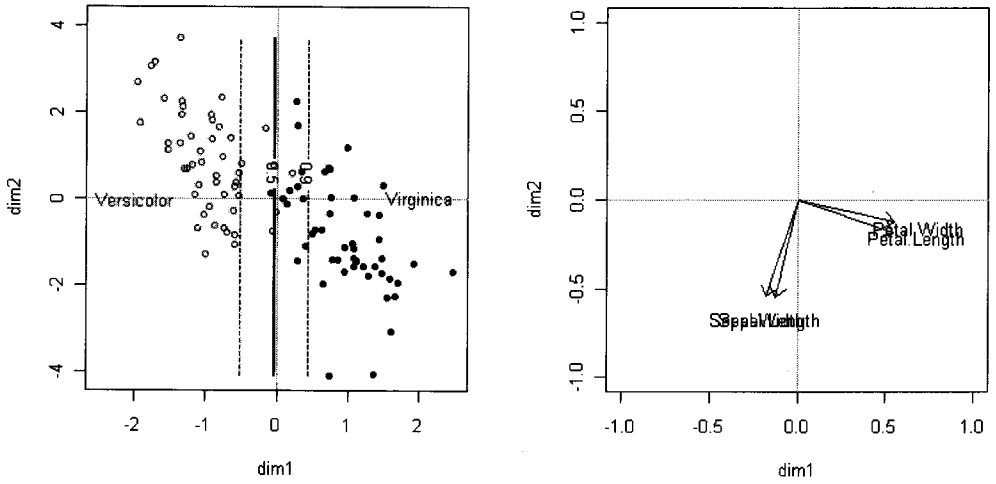

Figure 2: *Visualizing the linear SVM classification for the two-class Iris data (Versicolor vs. Virginica). The solid line is the classification boundary. Dotted lines denote probability contours at the 0.9 level.*

Example 2. **Iris data (Versicolor *vs.* Virginica)** Fisher's iris data consists of 150 observations in three known species (setosa, versicolor, virginica). Four measurements (sepal length, sepal width, petal length, petal width) are recorded for each observation. Here we consider the classification between two species, versicolor versus virginica.

Figure 2 shows the observation plot(left) and the variable plot(right) of the linear SVM classification ($C = 1$) with the classification boundary and two probability contours at the 0.9 level being classified to the dominant class. We find several observations near the boundary, so that there can happen misclassfications in the future. Variable plot shows that the first dimension is determined more or less by two petal variables and that second dimension is by two sepal variables.

3. Visualizing Nonlinear SVM Classifications

In nonlinear SVM classifications, the classification function is given in (1.4) as

$$f_N(x) = \sum_{i=1}^{n} \lambda_i y_i K(x_i, x) + b.$$

Hence the gradient of $f_N(x)$ can be written as

$$\nabla f_N(x) = \sum_{i=1}^{n} \lambda_i y_i \nabla K(x_i, x).$$

For the Gaussian RBF $K(x, x') = \exp(-\gamma \| x - x' \|^2)$,

$$\nabla f_N(x) = -2\gamma \sum_{i=1}^{n} \lambda_i y_i \exp\left(-\gamma \| x_i - x' \|^2\right)(x_i - x).$$

Since constructed SVM classification depends only on $L(\leq n)$ support vectors

$$X_s = \{x_{s_1}, x_{s_2}, \ldots, x_{s_L}\},$$

we restrict our attention to the gradient vectors at X_s: For Gaussian RBF,

$$\nabla f_N(x_{s_l}) = -2\gamma \sum_{i=1}^{n} \lambda_i y_i \exp\left(-\gamma \| x_i - x'_{s_l} \|^2\right)(x_i - x_{s_l}), \quad l = 1, \ldots, L.$$

We propose the 2D display for nonlinear SVM classifications as follows.

First, obtain $p \times L$ matrix G^t consisting of gradient vectors as columns:

$$G^t = (\nabla f_N(x_{s_1}), \ldots, \nabla f_N(x_{s_L})).$$

Here we assume that the rows of $X^t = (x_1, \ldots, x_n)$ are standardized to have mean 0 and standard deviation 1.

Second, compute principal component direction vectors $v^{[1]}$ and $v^{[2]}$ that carries maximally dispersed projections of $\nabla f_N(x_{s_1}), \ldots, \nabla f_N(x_{s_L})$. That is, $v^{[1]}$ and $v^{[2]}$ are the eigenvectors corresponding to the largest and second largest eigenvalues of G^tG. Thus $\| v^{[1]} \| = 1$, $\| v^{[2]} \| = 1$, $v^{[1]t} v^{[2]} = 0$.

Third, n observations are dotted at

$$z_i^{[1]} = v^{[1]t} x_i, \qquad z_i^{[2]} = v^{[2]t} x_i \ \ (i = 1, \ldots, n)$$

on the first and the second axis, respectively, colored differently depending on their membership class ("red" for Class 0 or "blue" for Class 1). Accordingly, on the first and second axis, the feature variables are plotted with arrow heads at p components of

$$y^{[1]} = v^{[1]}, \qquad y^{[2]} = v^{[2]}.$$

Fourth, we may superimpose probability contours as in the case of linear SVM classifications.

Example 3. **Continuation of Example 1** Figure 3 visualizes the Gaussian RBF SVM classification ($C = 1$, $\gamma = 0.33$) constructed for the simulated dataset of Example 1. The variable plot(right)

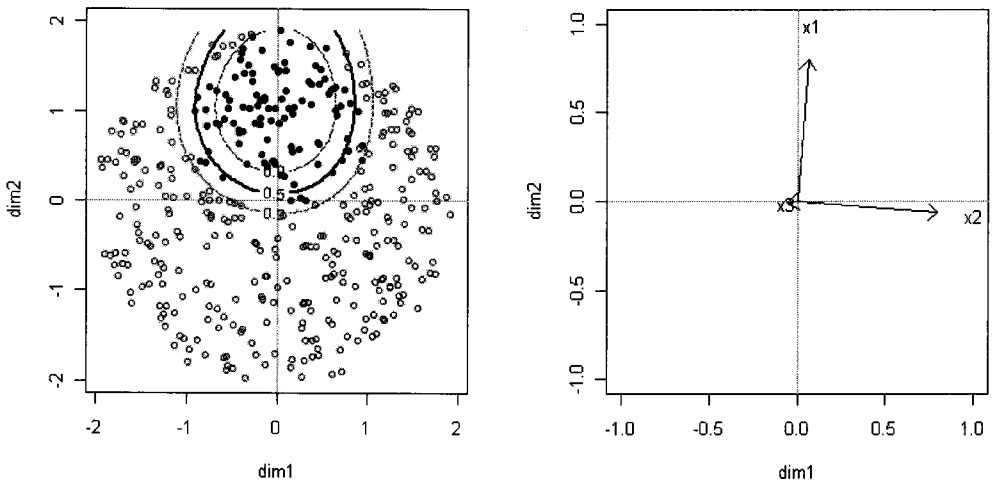

Figure 3: *Visualization of RBF SVM classification for the Monte-Carlo dataset. The central solid curve is the classification boundary. Dotted curves denote probability contours at the 0.9 level.*

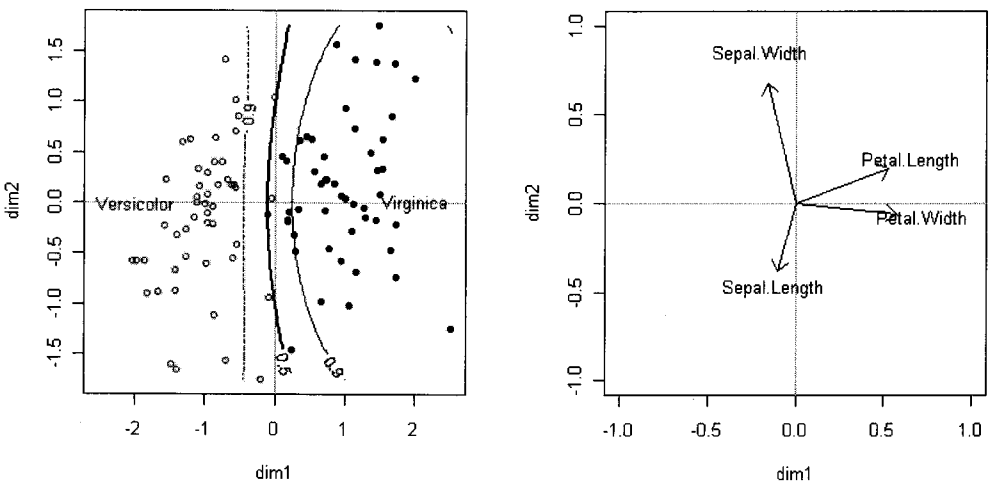

Figure 4: *Gaussian RBF SVM classification for the two-class Iris data (Versicolor vs. Virginica). The central solid curve is the classification boundary. Dotted curves denote probability contours at the 0.9 level.*

shows correctly that two variables, X_1 and X_2, are crucial and the third variable X_3 is irrelevant in determining the classification. Moreover, unlike the linear case, the observation plot(left) with classification boundary reveals the excellent performance of the nonlinear classification.

Example 4. **Continuation of Example 2** Figure 4 shows the observation plot(left) and the variable plot(right) of the Gaussian RBF SVM classification ($C = 1, \gamma = 0.25$) with classification boundary and two probability contours of the 0.9 level being classified to the dominant class. The first axis of nonlinear SVM classification is more or less similar to that of the linear SVM classication, visualized in Figure 2, while the second axes of linear and nonlinear SVM classifications are quite different.

4. *m*-class Problem

For classification problem for more than $m \geq 2$ classes, SVM classification consists of all pairwise classifications: Inside the m-class SVM classification, there exist ${}_mC_2$ classification functions of the form $f^{[k_1,k_2]}(x)$, which is expressed with $L_{[k_1,k_2]}(\leq n)$ support vectors, that distinguishes Class k_1 against Class k_2. Thus we need to gather the gradient vectors into the matrix G_{k_1,k_2} of $f^{[k_1,k_2]}(x)$ for $1 \leq k_1 < k_2 \leq m$ at the support vectors for respective classification, where

$$G^t_{k_1,k_2} = \left(\nabla f^{[k_1,k_2]}(x_{s_1}), \ldots, \nabla f^{[k_1,k_2]}\left(x_{s_{L_{[k_1,k_2]}}}\right)\right)$$

and join all $G^t_{k_1,k_2}$'s side by side:

$$G^t_{k_1,k_2} = \left(G^t_{1,2}, \ldots, G^t_{1,m}, \ldots, G^t_{m-1,m}\right). \tag{4.1}$$

Then, compute principal component direction vectors $v^{[1]}$ and $v^{[2]}$ that carry maximally dispersed projections of all column vectors of G^t. That is, $v^{[1]}$ and $v^{[2]}$ are the eigenvectors corresponding to the largest and second largest eigenvalues of G^tG. Thus $\| v^{[1]} \| = 1$, $\| v^{[2]} \| = 1$, $v^{[1]t}v^{[2]} = 0$.

Hence, n observations are dotted at

$$z_i^{[1]} = v^{[1]t}x_i, \qquad z_i^{[2]} = v^{[2]t}x_i \ \ (i = 1, \ldots, n)$$

on the first and second axis, respectively, colored differently depending on their membership class ("blue" for Class 1, "green" for Class 2,..., "red" for Class m). Accordingly, on the first and second axis, the feature variables are plotted with arrow heads at p components of

$$y^{[1]} = v^{[1]}, \qquad y^{[2]} = v^{[2]}.$$

We may superimpose probability contours as for the case of two-class SVM classifications.

Example 5. **Monte-Carlo Simulated Dataset** The dataset is the same as that in Example 1 except class labels assigned to the units and the number of classes. Class labels 1, 2 or 3 are determined by the following rule.

If $(x_1 - 1)^2 + x_2^2 \leq 1$, it is assigned to Class 1.

If $(x_1 + 1)^2 + x_2^2 \leq 1$, it is assigned to Class 2.

Elsewhere, it is assigned to Class 3.

Figure 5 visualizes Gaussian RBF SVM classification for this three-class Monte-Carlo dataset. As in the nonlinear case with two classes, the variable plot(right) shows correctly that X_1 and X_2 are crucial and that X_3 is irrelevant in forming the classification. Moreover, the observation plot with classification boundary(left) manifests the excellent performance of the nonlinear classification.

Example 6. **Olive Oils data from four South areas of Italy** The Olive Oils data consists of eight fatty acid composition measurements (palmitic, palmitoleic, stearic, oleic, linoleic, linolenic, arachidic, and eicosenoic) in three regions (North, South, Sardina), which can be divided into nine areas (Cook and Swayne, 2007). In this example, we restrict our attention to the data from South region which is divided into four areas (Calabria, North-Apulia, Sicily, South-Apulia). Thus the number of classes is four.

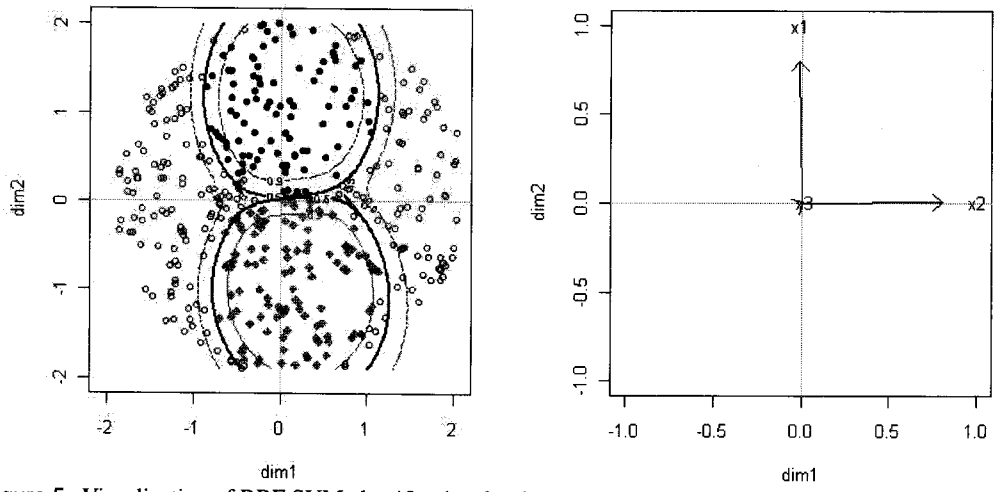

Figure 5: *Visualization of RBF SVM classification for the Monte-Carlo dataset. Dark solid curve is the classification boundary. Dotted curves denote probability contours at the 0.9 level.*

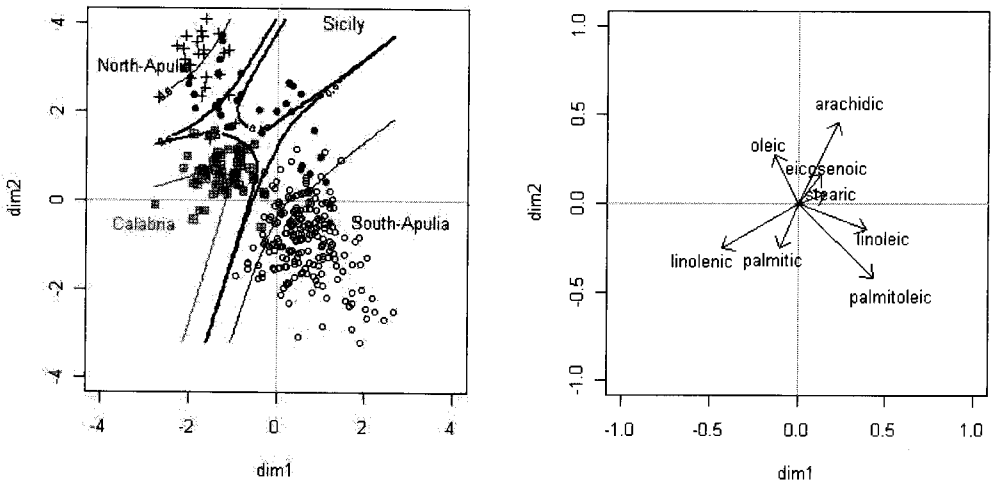

Figure 6: *Visualization of the linear SVM classification for Italian Olive Oils data from Four Regions (Calabria, North-Apulia, Sicily, South-Apulia). Dark solid curve is the classification boundary. Dotted curves denote probability contours at the 0.9 level.*

Figure 6 shows classification map of the linear SVM classification. From the map, we can see that Sicily oils are duplicated with other area oils. In their graphical analysis of the same dataset, Cook and Swayne (2007, p.99) commented that Sicily oils used borrowed olives from neighboring areas. Setting aside Sicily oils, North-Apulia, Calabria, and South-Apulia oils are lined up from Upper-Left to Lower-Right, corresponding to contrasting directions of "oleic" and "palmitoleic", respectively.

5. Concluding Remarks

For the m-class SVM classification, the matrix G^t of (4.1) formed with all $p \times 1$ gradient vectors of $m(m-1)/2$ classification functions which are determined by $\sum_{k_1,k_2} L_{[k_1,k_2]}$ support vectors, could have

$m(m-1)/2 \cdot n$ columns. Hence, in the case of large n, we need a big storage for G^t. Such scalability problem, however, does not affect the computation much since we compute eigenvalue-eigenvector decomposition of G^tG which is $p \times p$.

In all plots of this paper, we projected SVM classifications onto the 2D subspace for brevity of visual presentations. But our methods can be easily extended to yield the 3D display, that may be needed when the explained proportion by the first two dimensions is not large enough. The explained proportions in Figures 1, 2, 3, 4, 5 and 6 are 100%, 100%, 99.1%, 97.9%, 97.3%, 76.7%, respectively. The 3D display for the linear SVM classification for the four-class Olive Oils data would have explained 94.5% of all gradient vectors.

References

Chang, C. C. and Lin, C. J. (2001). LIBSVM: a library for support vector machines. Software available at `http://www.csie.ntu.edu.tw/~cjin/libsvm`.

Cook, D. and Swayne, D. (2007). *Interactive and Dynamic Graphics for Data Analysis*, Springer, New York.

Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York.

Huh, M. H. (2009). Visualizing multi-variable prediction functions by segmented k-CPG's, *Communications of the Korean Statistical Society*, **16**, *185–193.*

Huh, M. H. and Lee, Y. G. (2008). Simple graphs for complex prediction functions, Communications of the Korean Statistical Society, **15**, *343–351.*

Platt, J. (2000). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In Advances in Large Margin Classifiers, *Edited by A. Smola, P. Bartlett, B. Scholkopf and D. Dchuurmans. Cambridge, MA.*

Vapnik, V. N. (1999). The Nature of Statistical Learning Theory, *Springer, New York.*

Wang, J., Shen, X. and Liu, Y. (2008). Probability estimation for large-margin classifiers, Biometrika, **95**, *149–167.*

Wickham, H., Caragea, D. and Cook, D. (2006). Exploring high-dimensional classification boundaries, Unpublished manuscript.

Wu, T. F., Lin, C. J. and Weng, R. C. (2004). Probability estimates for multi-class classification by pairwise coupling, Journal of Machine Learning Research, **5**, *975–1005.*

Received June 2009; Accepted July 2009