Tree-shaped Decision Schema for Multi-classification

Ping LING\textsuperscript{1, a,*}, Xiang-Sheng RONG\textsuperscript{2}, Yong-Jia JIANG\textsuperscript{2}, De-Ping YANG\textsuperscript{2}, Ming XU\textsuperscript{2}

\textsuperscript{1}College of Computer Science and Technology, Jiangsu Normal University, Xuzhou, 221116, China
\textsuperscript{2}Air Force Logistics of P.L.A, Xuzhou 221000, China
a\textsuperscript{lingiceling@163.com}
*Corresponding author

Keywords: SVCMR, Kernel, Tree-shaped decision frame, Self-tuning, Informative metric.

Abstract. This paper presents a novel \( M \)-class classification schema. SVCMR creates a tree-shaped decision frame which contains \( M/2 \) three-separation classifiers as decision nodes. In an informed order, a series of basic classifiers are trained on the reduced dataset, and this order ensures the less number of basic models to cover decisions on all classes. SVCMR is characterized with individual parameterization of Kernel scale and penalty coefficient. Experiments on real experiments demonstrate the fine performance of SVCMR.

Introduction

Support Vector Machine (SVM) \cite{1} behaves excellently in binary classification. However, it cannot present an easy solution for multi-classification. Many corresponding methods include: one-versus-one method (1-vs-1) using winner-takes-all strategy \cite{2}; one-versus-rest (1-vs-r) method implemented by max-wins voting \cite{3}; DAGSVM \cite{4} and error-correcting codes \cite{5}. For a \( M \)-class classification problem, 1-vs-r model constructs \( M \) SVMs for each class against rest classes. 1-vs-1 method needs \( M(M-1)/2 \) SVMs and each SVM is trained for a pair of classes. Recently, K-SVCR was proposed in \cite{6} by combining SVM and support vector regression (SVR). Then, \( \nu \)-K-SVCR was given by \cite{7}, which adopts parameter \( \nu \) into the optimization. \( \nu \)-K-SVCR builds \( M(M-1)/2 \) basic classifiers, and the final label assignment is decided by voting strategy. In this paper, this model is named as SVCMR.

SVCMR employs SVMR as basic classifier and constructs a decision tree that contains \( M/2 \) decision nodes. Then label assignment amounts to traveling a path in this tree. A class-selection rule is proposed to sort class pairs in a wise order. And basic classifiers are trained on pairs in the turn specified by this order. Support Vector Clustering (SVC) \cite{8, 9} procedure is adopted to extract working set on which consequent operations are conducted. The schema includes SVC, SVM and SVR, so is named as SVCMR. SVCMR is characterized by two points: 1) self-tuning Kernel scale and penalty coefficient; 2) employing a weighted-voting kNN that is encoded with an informative metric to handle the rejected query. Experimental results comparison between SVCMR and state-of-the-arts demonstrates its fine performance and efficiency.
Related Work

SVMR

Assume \( \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \in X \times Y \) be training set, where \( Y = \{CL_1, CL_2, \ldots, CL_M\} \). For a pair of classes \( (CL_i, CL_j) \in Y \times Y \), let \( CL_i = \{x_i, \ldots, x_{i_1}\} \), \( CL_j = \{x_{i_1}, \ldots, x_{i_2}\} \), and the rest classes named as \( CL_o \), with \( CL_o = \{x_{i_1}, \ldots, x_i\} \). Set \( l_1 = l + l_2 \) and \( l_3 = l - l_2 \). SVMR solves:

\[
\begin{align*}
\min & \quad \frac{1}{2} \|w\|^2 + D_1 \cdot (-v_1 \cdot \rho + \frac{1}{l_{12}} \sum_{i=1}^{l_{12}} \xi_i) + D_2 \cdot (-v_2 \cdot \epsilon + \frac{1}{l} \sum_{l_{12}+1}^l (\phi_i + \phi_i^*)). \\
\text{s.t.} & \quad y_i(w \cdot x_j + b) \geq \rho - \xi_i, \quad i = 1, \ldots, l_{12}. \\
& \quad w \cdot x_j + b \leq \epsilon + \phi_i, \quad i = l_3 + 1, \ldots, l.
\end{align*}
\]

Here \( v_1, v_2 \in (0,1] \) are ratio pre-specified. \( D_1, D_2 \geq 0 \) are penalty parameters.

Support Vector Clustering

SVC is to find a smallest hyper sphere containing all data. Let \( \Phi \) be the nonlinear map, \( a \) is the center of sphere, \( R \) is the radius of sphere. The objective function with slack variables \( \xi_i \) is:

\[
\begin{align*}
\min_{R, \xi} & \quad R^2 + C \sum_i \xi_i \\
\text{s.t.} & \quad \|\Phi(x_i) - a\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0.
\end{align*}
\]

Here \( C \) is penalty parameter to tradeoff radius and slack variables. Transfer it to the Lagrangian function, then to the Wolfe dual, with Kernel trick, leading to:

\[
\begin{align*}
\max_{\gamma} & \quad \sum_{i} \sum_{j} \gamma_i \gamma_j \Phi(x_i) K(x_i, x_j) - \sum_{i} \gamma_i \phi_i \Phi(x_i) K(x_i, x) \quad \text{s.t.} \quad \sum_{i} \gamma_i = 1, \quad 0 \leq \gamma_i \leq C.
\end{align*}
\]

Spectral Analysis

Spectral analysis includes following steps: 1) Normalize affinity matrix \( H \) into \( H' \); 2) Eigen-decompose on \( H' \), and select the top \( M \) eigenvectors to form spectral matrix \( S \); 3) Run K-means on rows of \( S \). Label \( i \)th point according to \( i \)th row’s cluster membership.

SVCMR Schema

SVCMR Decision Frame

Let \( \{CL_1, \ldots, CL_M\} \) be the class series sorted in a certain order. If \( M \) is even, the decision frame of SVCMR is shown in Fig. 1. The decision frame contains \( M/2 \) three-child nodes, \( M \) two-child nodes and \( M \) leaf nodes. The three-child node is basic classifier, SVMR. Its three children correspond to three solutions. Two-child node investigates whether point meets the decision criterion required by some class. Leaf node includes three types. One does label assignment for a point. The second is to solve the rejection case, that is, point cannot meet radius constrains of all classes. The third type makes decision in the last remaining class when \( M \) is odd.
Dataset Reduction with Tuning SVC

Our method is identifying SVs that act as representatives. And the Gaussian Kernel scale is tuned data-dependently. That is, a scale factor $\sigma_x$ is learned for point $x$: $\sigma_x = ||x - x_r||$. $x_r$ is the $r$th point in the ascending sort of distance of $x$ to other points. $r$ is set: 1) Sort each row of distance matrix $D(i,j)$ in an ascending order. 2) $r_i = \max_j \{D(i,j+1) - D(i,j)\}$.

Class-selection Rule

Denote Gaussian Kernel matrix data representatives as $K$. Normalize $K$ as: $K' = D^{-1/2}KD^{-1/2}$, where $D$ is diagonal-shape with $D_{ij} = \Sigma_{j=1}^{n} (K)_{ij}$. Eigen-decompose $K'$, and form matrix $S$ by stacking the top $p$ eigenvectors as columns. Let $x_s$ as spectrum coordinates of $x$. $CL_i$ and $CL_j$ are:

$$\text{Sim}(CL_i, CL_j) = \frac{\sum_{x_s \in CL_i, y_s \in CL_j} k(x_s, y_s)}{|CL_i \times CL_j|}. \quad (5)$$

Two classes with the least similarity are selected as the first pair on which the first basic classifier is trained. Then among remaining classes, the second pair of classes is found according to the current-smallest similarity.

Radius Criterion

For class $CL_i$, to further confirm whether one point meets its membership criterion or not, its radius criterion is defined as: $R_i(x) = R_i <= \theta_i$. $R_i$ is the hyper sphere radius obtained in SVC. $R_i(x)$ computes the distance from $x$ to the sphere center:

$$R_i^2(x) = K(x, x) - 2\sum_{i = CL_i} y_i K(x_i, x) - \sum_{i = CL_i} y_i y_j K(x_i, x_j). \quad (6)$$

$R_i$ is generated by introducing a nbSV into (9). Let $\theta_i = v_1$. $R_i$. That allows some subtle perturbation. There $v_1$ controls a ratio of radius, whose setting is discussed in Section 4.1.
Multi Penalty Coefficients of Basic SVMR

Here instead of using a single uniform setting, we learn the individual penalty coefficient for each point from its neighborhood. Firstly we update the objective to a new version that is encoded with multi penalty coefficients:

\[
\min \frac{1}{2} \| w \|^2 + (-D_1 \cdot v_1 \cdot \rho + \frac{1}{|i|} \sum_{i \in |i|} D_i \cdot \xi_i ) + (-D_2 \cdot v_2 \cdot \epsilon + \frac{1}{|i|} \sum_{i \in |i|} D_i \cdot (\varphi_i + \varphi_i^*)).
\]

Correspondingly, only conditions concerned with \( r_i \) and \( y_i \) need to be updated as:

\[
0 \leq r_{ij} \leq \frac{D_i}{R_i}, \quad i = 1, \ldots, l_2, \quad 0 \leq r_i \leq \frac{D_i}{R_i}, \quad i = l_2 + 1, \ldots, l_3.
\]

According to this thinking, for point \( x_i \), its \( D_i \) is:

\[
|| \_ || \max || \_ || 1 \quad \frac{\text{래}_{	ext{C}_{	ext{L}}_{	ext{B}}}}{\text{래}_{	ext{C}_{	ext{L}}_{	ext{B}}}} \quad \frac{\text{래}_{	ext{C}_{	ext{L}}_{	ext{B}}}}{\text{래}_{	ext{C}_{	ext{L}}_{	ext{B}}}} \quad \frac{\text{래}_{	ext{C}_{	ext{L}}_{	ext{B}}}}{\text{래}_{	ext{C}_{	ext{L}}_{	ext{B}}}} \quad \frac{\text{래}_{	ext{C}_{	ext{L}}_{	ext{B}}}}{\text{래}_{	ext{C}_{	ext{L}}_{	ext{B}}}}
\]

Here, \( \text{Cen}_{B} = \text{ave}(\{ sv | sv \in SV, sv \in CLB \}) \), is the center of class that \( x_i \) belongs to. \( D_1 \) and \( D_2 \) work respectively for bi-classification between class \( CL_i \) and \( CL_j \), and regression over the rest classes \( CL_0 \). So according to their working context, \( D_1 \) and \( D_2 \) are set as:

\[
D_1 = \text{ave}(\{ D_j | i \in CL_i \cup CL_j \}), \quad D_2 = \text{ave}(\{ D_j | i \in CL_0 \})
\]

Here the average neighborhood size of data representatives is used to tune \( v_1 \) and \( v_2 \) them:

\[
v_1 = \frac{\min(r_i, r_j)}{N_I + N_J + N_O}, \quad v_2 = \frac{r_O}{N_I + N_J + N_O}
\]

Here \( r_i = \text{ave}(|r_i | x \in CL_i) \) , \( r_j = \text{ave}(|r_j | x \in CL_j) \) and \( r_O = \text{ave}(|r_j | x \in CL_0) \) . \( r_x \) is the neighborhood size in Section 3.2. \( N_I, N_J, N_O \) are sizes of \( CL_i, CL_j \) and \( CL_0 \).

Address the Rejected Case

Weighted-Voting kNN

The size of the neighborhood where kNN works is designed as an estimate of size of the natural dense region around \( Q \). Sort distance list of \( Q \) to other points \( Sd(Q, x_j) \) in the ascending order. Then neighborhood size \( NS = \max_j \{ Sd(Q, x_j) - Sd(Q, x_{j-1}) \} \). \( Sd(Q, x_j) \) is the new metric discussed next. Let occurring frequencies of \( M \) class be: \( t_1 \ldots t_M \), and \( J_{NEI} \) collects \( Q \)’s neighbors of class \( J \). Then \( Q \) is labeled as:

\[
\text{label}(Q) = \max_j \{ \mu_j t_j \}, \quad \text{with} \quad \mu_j = 1 - \frac{Sd(Q, J_{NEI})}{\sum_{j \in J_{NEI}} Sd(Q, J_{NEI})}, \quad Sd(Q, J_{NEI}) = \text{ave}\{Sd(Q, xJ) | xJ \in J_{NEI}\}. (11)
\]

Informative Metric Definition

The new metric is derived from the eigenvalues produced by SC because the magnitudes of them reveal the importance of spectrum dimensions, and consequently the discriminant information in the sense of classification. Let \( \lambda_i \) be \( i \)-th eigenvalue, \( p \) be the number of selected spectrum dimensions, which is determined by the max gap in the descending-list of eigenvalues. The new metric and the corresponding distance definition are developed as:

\[
\mu_j = \frac{\exp(A_j \lambda_i)}{\sum_{j=1}^p \exp(A_j \lambda_i)} \quad \text{and} \quad Sd(x^*, y^*) = \sqrt{(x^* - y^*)^T \mu(x^* - y^*)}
\]

\[
(12)
\]
\( x_j \) is the spectrum coordinates of \( x \). \( A \) controls the influence of \( \lambda_j \) on \( \mu_j \). The closest distance between two SVs, \( x_s, y_s \), coming from two classes acts as the setting value of \( A \):

\[
A = \exp\left(-\frac{d_f^*}{df}\right), \quad \text{with} \quad d_f^* = \min\{\| x_s - y_s \|\}, \quad df = \text{ave}\{\| x_s - y_s \|\}, \quad \text{label}(x_s) \neq \text{label}(y_s).
\]

Nystrom method [8] is used to yield spectrum coordinate of \( Q \). Assume the number of data representatives as \( m \), the matrix among existing data and \( Q \) as \( S_{m+1,m} \), then:

\[
V_i = \frac{m}{m+1} S_{m+1,m} V_i.
\]

Experiment Results

Firstly, plane points in Fig. 2 are tested. Fig. 3 shows the result produced by tuning SVC. The generated SVs are shown by ‘+’. SVC is run with a series scales: \( \{0.162, 0.262...0.962, 1.062...2.262\} \), but find the yielded SVs fail to describe contours correctly. Fig. 3 and 4 are two cases of them. This shows the suggested tuning approach can adapt to diverse distributions.

A class similarity matrix is formed in Fig. 6. Based on it, the list of class pairs is obtained: \((CL_2, CL_5)\), \((CL_1, CL_3)\), \((CL_4)\). Visually, it is true that class 2 and class 5 are least relevant, and class 1 and class 3 follows. Two types SVs are produced: one type that expresses classification interfaces; another type that expresses regression tube. They are shown by ‘+’ and ‘*’. In this dataset, other SV-based classifiers are tested to compare their time computation. They are: SVM11 (1-vs-1 SVM), SVM1r (1-vs-r SVM), and \( \nu \)-K-SVCR. Fig. 7 plots the changes of time as the increase of training data amount, where the advantage of SVCMR is obvious.

Then SVCMR is conducted on real datasets: Iris, Wine and Glass [10]. Some other classifiers are tested: quadratic multi-class SVM (SVMqp) and linear multi-class SVM (SVMlp) [11]. Their results come from [7]. Gaussian Kernel with scale \( \sigma \) being 0.2236 for Glass set. Set \( \nu_1 = \nu_2 = 0.01 \) for \( \nu \)-K-SVCR. And SVCMR is performed in a searching-version, that is, to search scale parameter space. This procedure is named as \( \sigma \)-SVCMR. Result of 10 runs is shown in Table 1.

From Table 1, for Iris and Glass, SVCMR is competitive to or outperforms other methods. For Wine set, the fact that 178 points cover 13 dimensions leads to the weak neighborhood
information. And the tuning approach cannot bring much help to refine affinity expressions. So SVCMMR doesn’t reach state-of-the-art accuracy. SVCMMR is competitive or close to the searching version, and it is friendlier than $\sigma$-SVCMMR due to the ease in parameterization.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SVM11</th>
<th>SVM1r</th>
<th>SVMqp</th>
<th>SVMlp</th>
<th>$\nu$-K-SVCR</th>
<th>SVCMMR</th>
<th>$\sigma$-SVCMMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>98.7</td>
<td>98.7</td>
<td>98</td>
<td>98</td>
<td>98.7</td>
<td>98.78</td>
<td>98.83</td>
</tr>
<tr>
<td>Wine</td>
<td>94.3</td>
<td>94.4</td>
<td>96.4</td>
<td>89.2</td>
<td>96.7</td>
<td>96.57</td>
<td>97.2</td>
</tr>
<tr>
<td>Glass</td>
<td>64.8</td>
<td>63.6</td>
<td>64.4</td>
<td>62.8</td>
<td>[67.6, 63.8]</td>
<td>67.98</td>
<td>68.55</td>
</tr>
<tr>
<td>Diabetes</td>
<td>76.2</td>
<td>76.16</td>
<td>75.8</td>
<td>76.2</td>
<td>81.2</td>
<td>81.38</td>
<td>82.2</td>
</tr>
<tr>
<td>Heart</td>
<td>85.9</td>
<td>86.2</td>
<td>84</td>
<td>83.2</td>
<td>87.3</td>
<td>87.7</td>
<td>88.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SVM11</th>
<th>SVM1r</th>
<th>DAGSVM</th>
<th>$\nu$-K-SVCR</th>
<th>SVCMMR</th>
<th>$\sigma$-SVCMMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>{NG2, NG3, NG4} (200)</td>
<td>69.0</td>
<td>69.2</td>
<td>69.5</td>
<td>69.7</td>
<td>70.14</td>
<td></td>
</tr>
<tr>
<td>{NG2(100), NG3(50), NG4(200)}</td>
<td>67.3</td>
<td>66.8</td>
<td>67.8</td>
<td>68.1</td>
<td>69.83</td>
<td></td>
</tr>
<tr>
<td>{NG6(200), NG7(50),NG8(150)}</td>
<td>82.3</td>
<td>83.2</td>
<td>84.1</td>
<td>84.82</td>
<td>84.85</td>
<td></td>
</tr>
<tr>
<td>{NG1(50), NG2(200),NG7(150), NG8(50)}</td>
<td>86.1</td>
<td>86.9</td>
<td>87.0</td>
<td>87.8</td>
<td>88.49</td>
<td></td>
</tr>
<tr>
<td>{0, 1, 2} (200)</td>
<td>97.8</td>
<td>97.3</td>
<td>97.3</td>
<td>98.1</td>
<td>98.2</td>
<td></td>
</tr>
<tr>
<td>{3(100), 5(60), 8(200)}</td>
<td>96.85</td>
<td>96.67</td>
<td>97.3</td>
<td>97.3</td>
<td>98.11</td>
<td></td>
</tr>
<tr>
<td>{2(100), 3(50), 5(150), 7(200)}</td>
<td>96.32</td>
<td>96.36</td>
<td>96.5</td>
<td>96.72</td>
<td>97.92</td>
<td></td>
</tr>
<tr>
<td>{3(100), 6(30), 5(150), 8(350)}</td>
<td>95.95</td>
<td>96.23</td>
<td>95.4</td>
<td>97.01</td>
<td>97.91</td>
<td></td>
</tr>
</tbody>
</table>

Finally, SVCMMR is performed on US Postal Codes [12] and News Groups [13]. US Postal Codes has 9298 samples covering digit ‘0’ to ‘9’. News Groups contains about 20,000 articles evenly divided among 20 newsgroups labelled as NG1…NG20. Classification accuracies are listed in Table 2, where DAGSVM refers to the 1-vs-1 SVMs combined with the DAGSVM idea proposed in [4]. Kernel scales used by other methods are set with cross-validation. Clearly, SVCMMR outperforms SVM11, SVM1r, and DAGSVM. It is competitive with, or in most cases better than $\nu$-K-SVCR. Especially in the datasets with uneven class sizes, SVCMMR keeps fine classification accuracy, while other classifiers are influenced by this change.

**Summary**

A tree-shaped decision schema SVCMMR is proposed in this paper for multi-classification. $M/2$ basic classifiers are constructed in a reasonable order, and each basic classifier three-sided partitions data. The test procedure corresponds to a path in this tree-shaped frame. The rejected case is addressed by the weighted-voting kNN coupled with an informative metric. The hyper parameters concerned with support-vector procedure are tuned context-dependently. SVCMMR gives good clustering results on real datasets, which shows the quality of designed schema.

**Acknowledgement**

This work was supported by the key Natural Science Foundation of Jiangsu Normal University under Grant No. 15XLA07.
References