Kernel Logistic Regression and the Import Vector Machine

Ji Zhu and Trevor Hastie
Journal of Computational and Graphical Statistics, 2005

Presented by Mingtao Ding
Duke University

December 8, 2011
Summary

The authors propose a new approach for classification, called the import vector machine (IVM).

Provides estimates of the class probabilities. Often these are more useful than the classifications.

Generalizes naturally to M-class classification through kernel logistic regression (KLR).
Problem and Objective

- **Supervised Learning Problem**: a set of training data \{\( (x_i, y_i) \)\}, where \( x_i \in \mathcal{R}^p \) is an input vector, and \( y_i \) (dependent on \( x_i \)) is a univariate continuous output for the regress problem or binary output for the classification problem.

- **Objective**: learn a predictive function \( f(x) \) from the training data

\[
\min_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \frac{\lambda}{2} \Phi(\|f\|_{\mathcal{F}}) \right\}.
\]

In this presentation, \( \mathcal{F} \) is assumed as an reproducing kernel Hilbert space (RKHS) \( \mathcal{H}_K \).
The standard SVM can be fitted via Loss + Regularization

$$\min_{f \in \mathcal{H}_K} \left\{ \frac{1}{n} \sum_{i=1}^{n} [1 - y_if(x_i)]_+ + \frac{\lambda}{2} \|f\|_{\mathcal{H}_K}^2 \right\}.$$ 

Under very general conditions, the solution has the form

$$f(x) = \sum_{i=1}^{n} a_i K(x, x_i).$$

Example Conditions:
Arbitrary $L(((x_1, y_1, f(x_1)), (x_2, y_2, f(x_2)), \cdots, (x_n, y_n, f(x_n)))$ and strictly monotonic increasing function $\Phi$.

The points with $y_if(x_i) > 1$ have no influence in loss function. As a consequence, it often happens that a sizeable fraction of the $n$ values of $a_i$ can be zero. The points corresponding nonzero $a_i$ are called supporting points.
The loss function $(1 - yf)_+$ is plotted in Fig.1, along with the negative log-likelihood (NLL) of the binomial distribution (of $y$ over $\{1, -1\}$).

$$
NLL = \ln \left( 1 + e^{-yf} \right) = \begin{cases} 
- \ln p, & \text{if } y = 1 \\
- \ln (1 - p), & \text{if } y = -1
\end{cases},
$$

where $p \equiv P(Y = 1|X = x) = \frac{1}{1 + e^{-f}}$. 

Figure 1: Two loss functions, $y \in \{-1, 1\}$.
The SVM only estimates \( \text{sign}[p(x) - 1/2] \) (by calculating the distances between \( x \) and the hyperplanes), without defining the class probability \( p(x) \).

The NLL of \( y \) has a similar shape to that of the SVM.

If we let \( y \in \{0, 1\} \), then

\[
\text{NLL} = - (y \ln p + (1 - y) \ln p) = - (yf - \ln (1 + e^f))
\]

This is the loss function of the classical KLR.
SVM and KLR

If we replace \((1 - yf)_+\) with \(\ln (1 + e^{-yf})\), the SVM becomes a KLR problem with the objective junction

\[
\min_{f \in \mathcal{H}_K} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ln (1 + e^{-yf}) + \frac{\lambda}{2} \|f\|_{\mathcal{H}_K}^2 \right\}.
\]

- **Advantages:**
  Offer a natural estimate of the class probability \(p(x)\).
  Can naturally be generalized to the M-Class case through kernel multi-logit regress.

- **Disadvantages:** For the KLR solution \(f (x) = \sum_{i=1}^{n} a_i K (x, x_i)\), all the \(a_i\)'s are nonzero
Figure 2: The solid black lines are classification boundaries; the dashed purple lines are Bayes optimal boundaries. For the SVM, the dotted black lines are the edges of the margins and the black points are the points exactly on the edges of the margin. For the IVM, the dotted black lines are the $p_1(x) = 0.25$ and 0.75 lines and the black points are the import points. Since the classification boundaries of KLR and the IVM are almost identical, we omit the picture of KLR here.
Suppose the basis functions of the transformed features space $h(x)$ is rich enough, so that the superplane $f(x) = h(x)^T \beta + \beta_0 = 0$ can separate the training data.

**Theorem 1** Denote by $\hat{\beta}(\lambda)$ the solution to KLR

$$
\min_{f \in \mathcal{H}_K} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ln (1 + e^{-y_i f}) + \frac{\lambda}{2} \|f\|_{\mathcal{H}_K}^2 \right\},
$$

then $\lim_{\lambda \to 0} \hat{\beta}(\lambda) = \beta^*$, where $\beta^*$ is the margin-maximizing SVM solution.
The objective function of KLR can be written as

\[ H = \frac{1}{n} \mathbf{1}^T \ln \left( 1 + e^{y \cdot (K_1 \mathbf{a})} \right) + \frac{\lambda}{2} \mathbf{a}^T K_2 \mathbf{a}. \]

To find \( \mathbf{a} \), we set the derivative of \( H \) with respect to \( \mathbf{a} \) equal to 0, and use the Newton method iteratively solve the score equation. The Newton update can be written as

\[ \mathbf{a}^{(k)} = \left( \frac{1}{n} K_1^T \mathbf{W} K_1 + \lambda K_2 \right)^{-1} K_1^T \mathbf{W} z \]

where \( \mathbf{a}^{(k)} \) is the value of \( \mathbf{a} \) in the \( k \)th step, and

\[ z = \frac{1}{n} \left( K_1 \mathbf{a}^{(k-1)} + \mathbf{W}^{-1} (y \cdot \mathbf{p}) \right). \]
The computational cost of the KLR is $O(n^3)$. To save the cost, the IVM algorithm will find a sub-model to approximate the full model given by KLR.

The sub-model has the form

$$f(x) = \sum_{x_i \in S} a_i K(x, x_i)$$

where $S$ is a subset of the training data, and the data in $S$ are called import points.
Algorithm 1:

1. Let $S = \emptyset$, $L = \{x_1, x_2, \ldots, x_n\}$, $k = 1$.

2. For each $x_l \in L$, let

$$f_l(x) = \sum_{x_i \in S \cup \{x_l\}} a_i K(x, x_i)$$

Use the Newton-Raphson method to find $a$ to minimize

$$H(x_l) = \frac{1}{n} \sum_{i=1}^{n} \ln \left(1 + \exp(-y_i f_l(x_i))\right) + \frac{\lambda}{2} \|f_l(x)\|_{\mathcal{H}_K}^2$$

$$= \frac{1}{n} \mathbf{1}^T \ln \left(1 + \exp(-y \cdot (K_l^T x))\right) + \frac{\lambda}{2} a^T K_2^l a$$

where the regressor matrix

$$K_l^l = (K(x_i, x_i'))_{n \times k}, \; x_i \in \{x_1, \ldots, x_n\}, \; x_i' \in S \cup \{x_l\};$$

the regularization matrix

$$K_2^l = \text{diag}(\lambda, \lambda, \ldots, \lambda).$$
Import Vector Machine

Algorithm 1:

\[ K_2^l = (K(x_i, x_i'))_{k \times k}, x_i, x_i' \in S \cup \{x_l\}; \]

and \( k = |S| + 1. \)

3. Find

\[ x_{l^*} = \arg\min_{x_l \in \mathcal{L}} H(x_l). \]

Let \( S = S \cup \{x_{l^*}\}, \mathcal{L} = \mathcal{L} \setminus \{x_{l^*}\}, H_k = H(x_{l^*}), k = k + 1. \)

4. Repeat steps (2) and (3) until \( H_k \) converges.
The algorithm can be accelerated by revising Step (2) as

\[(2^*)\text{ For each } x_t \in \mathcal{L}, \text{ correspondingly augment } K_1 \text{ with a column, and } K_2 \text{ with a column and a row. Use the current sub-model from iteration } (k - 1) \text{ to compute } z \text{ in (21) and use the updating formula (20) to find } a. \text{ Compute (23).}\]

- Stopping rule for adding point to \( S \): run the algorithm until

\[
\frac{|H_k - H_{k - \Delta k}|}{|H_k|} < \epsilon.
\]

- Choosing the regularization parameter \( \lambda \): we can split all the data into a training set and a tuning set, and use the misclassification error on the tuning set as a criterion for choosing \( \lambda \) (Algorithm 3).
Algorithm 3:

1. Start with a large regularization parameter $\lambda$.

2. Let $\mathcal{S} = \emptyset$, $\mathcal{L} = \{x_1, x_2, \ldots, x_n\}$, $k = 1$. Let $a^{(0)} = 0$, hence $z = 2y/n$.

3. Run steps (2*), (3) and (4) of the revised Algorithm 2, until the stopping criterion is satisfied at $\mathcal{S} = \{x_{i_1}, \ldots, x_{i_k}\}$. Along the way, also compute the misclassification error on the tuning set.

4. Decrease $\lambda$ to a smaller value.

5. Repeat steps (3) and (4), starting with $\mathcal{S} = \{x_{i_1}, \ldots, x_{i_k}\}$.
Figure 3: Radial kernel is used. $n = 200$, $\sigma^2 = 0.7$, $\Delta k = 3$, $\epsilon = 0.001$, $\lambda$ decreases from $e^{10}$ to $e^{-10}$. The minimum misclassification rate 0.262 is found to correspond to $\lambda = 1$. 

Optimal Lambda = 1
Figure 4: Radial kernel is used. $n = 200$, $\sigma^2 = 0.7$, $\Delta k = 3$, $\epsilon = 0.001$, $\lambda = 1$. The stopping criterion is satisfied when $|S| = 19$. 
Table 1: Summary of the ten benchmark datasets. $n$ is the size of the training data, $p$ is the dimension of the original input, $\sigma^2$ is the parameter of the radial kernel, $\lambda$ is the tuning parameter, and $N$ is the size of the test data.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$n$</th>
<th>$p$</th>
<th>$\sigma^2$</th>
<th>$\lambda$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>400</td>
<td>2</td>
<td>1</td>
<td>$3.16 \times 10^{-3}$</td>
<td>4900</td>
</tr>
<tr>
<td>Breast-cancer</td>
<td>200</td>
<td>9</td>
<td>50</td>
<td>$6.58 \times 10^{-2}$</td>
<td>77</td>
</tr>
<tr>
<td>Flare-solar</td>
<td>666</td>
<td>9</td>
<td>30</td>
<td>0.978</td>
<td>400</td>
</tr>
<tr>
<td>German</td>
<td>700</td>
<td>20</td>
<td>55</td>
<td>0.316</td>
<td>300</td>
</tr>
<tr>
<td>Heart</td>
<td>170</td>
<td>13</td>
<td>120</td>
<td>0.316</td>
<td>100</td>
</tr>
<tr>
<td>Image</td>
<td>1300</td>
<td>18</td>
<td>3</td>
<td>0.002</td>
<td>1010</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>400</td>
<td>20</td>
<td>10</td>
<td>$10^{-9}$</td>
<td>7000</td>
</tr>
<tr>
<td>Thyroid</td>
<td>140</td>
<td>5</td>
<td>3</td>
<td>0.1</td>
<td>75</td>
</tr>
<tr>
<td>Titanic</td>
<td>150</td>
<td>3</td>
<td>2</td>
<td>$10^{-5}$</td>
<td>2051</td>
</tr>
<tr>
<td>Twonorm</td>
<td>400</td>
<td>20</td>
<td>40</td>
<td>0.316</td>
<td>7000</td>
</tr>
<tr>
<td>Waveform</td>
<td>400</td>
<td>21</td>
<td>20</td>
<td>1</td>
<td>4600</td>
</tr>
</tbody>
</table>
Table 2: Comparison of classification performance of SVM and IVM on ten benchmark datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SVM Error (%)</th>
<th>IVM Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>10.78(±0.68)</td>
<td>10.34(±0.46)</td>
</tr>
<tr>
<td>Breast-cancer</td>
<td>25.58(±4.50)</td>
<td>25.92(±4.79)</td>
</tr>
<tr>
<td>Flare-solar</td>
<td>32.65(±1.42)</td>
<td>33.66(±1.64)</td>
</tr>
<tr>
<td>German</td>
<td>22.88(±2.28)</td>
<td>23.53(±2.48)</td>
</tr>
<tr>
<td>Heart</td>
<td>15.95(±3.14)</td>
<td>15.80(±3.49)</td>
</tr>
<tr>
<td>Image</td>
<td>3.34(0.70)</td>
<td>3.31(±0.80)</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>2.03(±0.19)</td>
<td>1.97(±0.29)</td>
</tr>
<tr>
<td>Thyroid</td>
<td>4.80(±2.98)</td>
<td>5.00(±3.02)</td>
</tr>
<tr>
<td>Titanic</td>
<td>22.16(±0.60)</td>
<td>22.39(±1.03)</td>
</tr>
<tr>
<td>Twonorm</td>
<td>2.90(±0.25)</td>
<td>2.45(±0.15)</td>
</tr>
<tr>
<td>Waveform</td>
<td>9.98(±0.43)</td>
<td>10.13(±0.47)</td>
</tr>
</tbody>
</table>
Table 3: Comparison of number of kernel basis used by SVM and IVM on ten benchmark datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of SV</th>
<th># of IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>90(±10)</td>
<td>21(±7)</td>
</tr>
<tr>
<td>Breast-cancer</td>
<td>115(±5)</td>
<td>14(±3)</td>
</tr>
<tr>
<td>Flare-solar</td>
<td>597(±8)</td>
<td>9(±1)</td>
</tr>
<tr>
<td>German</td>
<td>407(±10)</td>
<td>17(±2)</td>
</tr>
<tr>
<td>Heart</td>
<td>90(±4)</td>
<td>12(±2)</td>
</tr>
<tr>
<td>Image</td>
<td>221(±11)</td>
<td>72(±18)</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>89(±5)</td>
<td>72(±30)</td>
</tr>
<tr>
<td>Thyroid</td>
<td>21(±2)</td>
<td>22(±3)</td>
</tr>
<tr>
<td>Titanic</td>
<td>69(±9)</td>
<td>8(±2)</td>
</tr>
<tr>
<td>Twonorm</td>
<td>70(±5)</td>
<td>24(±4)</td>
</tr>
<tr>
<td>Waveform</td>
<td>151(±9)</td>
<td>26(±3)</td>
</tr>
</tbody>
</table>
Generalization to M-Class Case

Similar with the kernel multi-logit regression, we define the class probabilities as

\[
p_1(x) = \frac{e^{f_1(x)}}{\sum_{c=1}^{C} e^{f_c(x)}},
\]

\[
p_2(x) = \frac{e^{f_2(x)}}{\sum_{c=1}^{C} e^{f_c(x)}},
\]

\[\vdots\]

\[
p_C(x) = \frac{e^{f_C(x)}}{\sum_{c=1}^{C} e^{f_c(x)}},
\]

\[
\sum_{c=1}^{C} f_c(x) = 0.
\]
Generalization to M-Class Case

The M-class KLR fits a model to minimize the regularized NLL of multinomial distribution

\[ H = -\frac{1}{n} \sum_{i=1}^{n} \ln p_{y_i}(x_i) + \frac{\lambda}{2} \| f \|_{H_K}^2 \]

\[ = \frac{1}{n} \sum_{i=1}^{n} \left[ -y_i^T f(x_i) + \ln \left( e^{f_1(x_i)} + \cdots + e^{f_C(x_i)} \right) \right] + \frac{\lambda}{2} \| f \|_{H_K}^2 \]

The approximate solution can be obtained by a M-class IVM procedure, which is similar to the two-class case.
Generalization to M-Class Case

Multi-class IVM – 32 Import Points

![Diagram of multi-class IVM with 32 import points](image)

Training Error: 0.237
Test Error: 0.259
Bayes Error: 0.251

Figure 6: Radial kernel is used. $C = 3$, $n = 300$, $\lambda = 0.368$, $|S| = 32$. 
Conclusion

- IVM not only performs as well as SVM in two-class classification, but also can naturally be generalized to the M-class case.

- **Computational Cost:** KLR \((O(n^3))\), SVM \((O(n^2 n_s))\), IVM \((O(n^2 n_i^2), O(Cn^2 n_i^2))\).

- IVM has limiting optimal margin properties.