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 $\{(\mathbf{x}_i, y_i)\}$, where $\mathbf{x}_i \in \mathcal{R}^p$ is an input vector, and y_i (dependent on $\mathbf{x}_i$) 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

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

In this presentation, $\mathcal{F}$ is assumed as an reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$.

SVM and KLR

- The standard SVM can be fitted via Loss + Regularization

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

- Under very general conditions, the solution has the form

$$f\left(\mathbf{x}\right)=\sum_{i=1}^{n}a_iK\left(\mathbf{x},\mathbf{x}_i\right).$$

  Example Conditions:
  Arbitrary $L\left(\left(\mathbf{x}_1,y_1,f(\mathbf{x}_1),\right)\left(\mathbf{x}_2,y_2,f(\mathbf{x}_2),\cdots,\right)\left(\mathbf{x}_n,y_n,f(\mathbf{x}_n)\right)\right)$ and strictly monotonic increasing function Φ.

- The points with $y_if(\mathbf{x}_i)>1$ 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

SVM and KLR

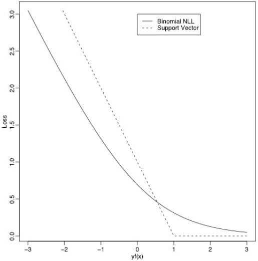


Figure 1: *Two loss functions,* $y \in \{-1, 1\}$.

The loss function $(1 - yf)_+$ is plotted in Fig.1, along with the negative log-likelihood (NLL) of the binomial distribution (of y over $\{1, -1\}$).

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

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

SVM and KLR

- The SVM only estimates $\mathrm{sign}\left[p(\mathbf{x}) - 1/2\right]$ (by calculating the distances between $\mathbf{x}$ and the hyperplanes), without defining the class probability $p(\mathbf{x})$.
- The NLL of y has a similar shape to that of the SVM.
- If we let $y \in \{0, 1\}$, then

  $$\mathrm{NLL} = -(y \ln p + (1 - y) \ln p) = -\left(yf - \ln\left(1 + e^{f}\right)\right).$$

  This is the loss function of the classical KLR.

SVM and KLR

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

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

- Advantages:
  Offer a natural estimate of the class probability $p(\mathbf{x})$.

  Can naturally be generalized to the M-Class case through kernel multi-logit regress.
- Disadvantages: For the KLR solution $f(\mathbf{x}) = \sum_{i=1}^{n} a_i K(\mathbf{x}, \mathbf{x}_i)$, all the a_i's are nonzero

SVM and KLR

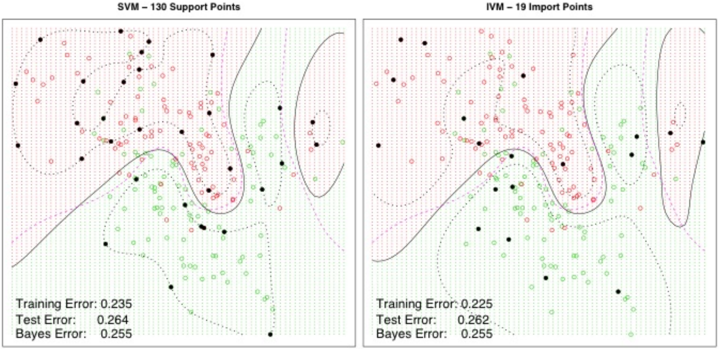


Figure 2: *The solid black lines are classification boundaries; the dashed purple lines are Bayes optimal boundaries. For the SVM, the dotted black lines are the edges of the margins and the black points are the points exactly on the edges of the margin. For the IVM, the dotted black lines are the* $p_1(\mathbf{x}) = 0.25$ *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.*

KLR as a Margin Maximizer

Suppose the basis functions of the transformed features space $\boldsymbol{h}(x)$ is rich enough, so that the superplane $f(x) = \boldsymbol{h}(x)^T\beta + \beta_0 = 0$ can separate the training data.

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

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

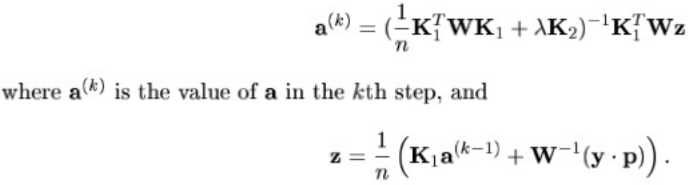
then $\lim_{\lambda \to 0} \hat{\beta}(\lambda) = \beta^*$, where β^* is the margin-maximizing SVM solution.

Import Vector Machine

The objective function of KLR can be written as

$$H = \frac{1}{n}\mathbf{1}^T \ln\left(1 + e^{\mathbf{y}\cdot(\mathbf{K}_1\mathbf{a})}\right) + \frac{\lambda}{2}\mathbf{a}^T\mathbf{K}_2\mathbf{a}.$$

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



$$\mathbf{a}^{(k)} = (\frac{1}{n}\mathbf{K}_1^T\mathbf{W}\mathbf{K}_1 + \lambda\mathbf{K}_2)^{-1}\mathbf{K}_1^T\mathbf{W}\mathbf{z}$$

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

$$\mathbf{z} = \frac{1}{n}\left(\mathbf{K}_1\mathbf{a}^{(k-1)} + \mathbf{W}^{-1}(\mathbf{y}\cdot\mathbf{p})\right).$$

Import Vector Machine

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.

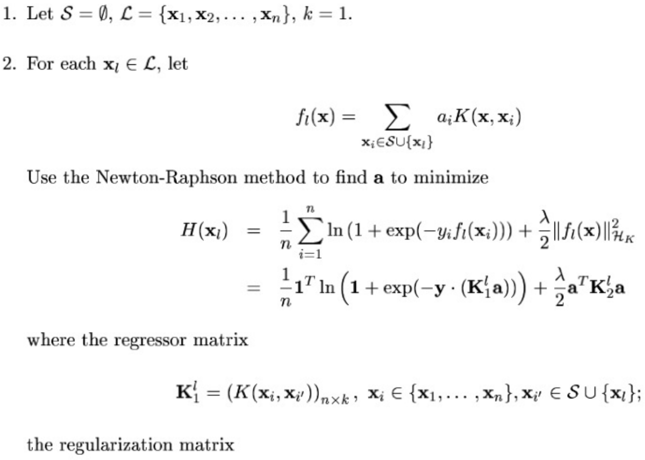
The sub-model has the form

$$f(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathcal{S}} a_i K(\mathbf{x}, \mathbf{x}_i)$$

where $\mathcal{S}$ is a subset of the training data, and the data in $\mathcal{S}$ are called import points.

Import Vector Machine

Algorithm 1:



1. Let $\mathcal{S} = \emptyset$, $\mathcal{L} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, $k = 1$.

2. For each $\mathbf{x}_l \in \mathcal{L}$, let

$$f_l(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathcal{S} \cup \{\mathbf{x}_l\}} a_i K(\mathbf{x}, \mathbf{x}_i)$$

Use the Newton-Raphson method to find $\mathbf{a}$ to minimize

$$\begin{aligned} H(\mathbf{x}_l) &= \frac{1}{n}\sum_{i=1}^{n} \ln\left(1 + \exp(-y_i f_l(\mathbf{x}_i))\right) + \frac{\lambda}{2}\|f_l(\mathbf{x})\|^2_{\mathcal{H}_K} \\ &= \frac{1}{n}\mathbf{1}^T \ln\left(\mathbf{1} + \exp(-\mathbf{y} \cdot (\mathbf{K}^l_1 \mathbf{a}))\right) + \frac{\lambda}{2}\mathbf{a}^T \mathbf{K}^l_2 \mathbf{a} \end{aligned}$$

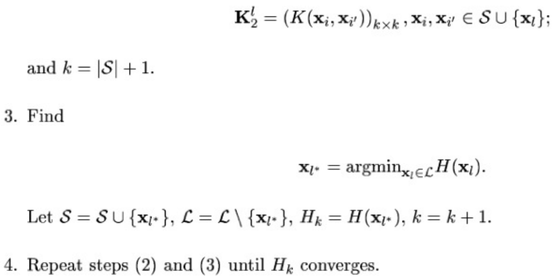
where the regressor matrix

$$\mathbf{K}^l_1 = (K(\mathbf{x}_i, \mathbf{x}_{i'}))_{n \times k}, \ \mathbf{x}_i \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \mathbf{x}_{i'} \in \mathcal{S} \cup \{\mathbf{x}_l\};$$

the regularization matrix

Import Vector Machine

Algorithm 1:



$$\mathbf{K}_2^l = (K(\mathbf{x}_i, \mathbf{x}_{i'}))_{k \times k}, \mathbf{x}_i, \mathbf{x}_{i'} \in \mathcal{S} \cup \{\mathbf{x}_l\};$$

and $k = |\mathcal{S}| + 1$.

3. Find

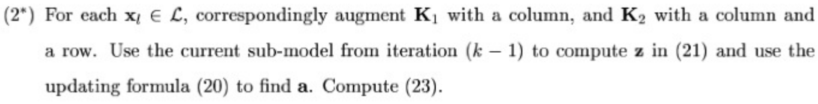
$$\mathbf{x}_{l^*} = \mathrm{argmin}_{\mathbf{x}_l \in \mathcal{L}} H(\mathbf{x}_l).$$

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

4. Repeat steps (2) and (3) until H_k converges.

Import Vector Machine

- The algorithm can be accelerated by revising Step (2) as



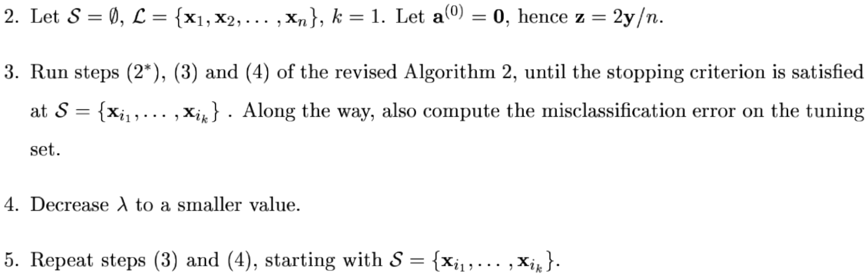
(2*) For each $\mathbf{x}_l \in \mathcal{L}$, correspondingly augment $\mathbf{K}_1$ with a column, and $\mathbf{K}_2$ with a column and a row. Use the current sub-model from iteration $(k-1)$ to compute $\mathbf{z}$ in (21) and use the updating formula (20) to find $\mathbf{a}$. Compute (23).

- Stopping rule for adding point to $\mathcal{S}$: run the algorithm until
$$\frac{|H_k - H_{k-\Delta k}|}{|H_k|} < \epsilon.$$
- Choosing the regularization parameter λ: 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 λ (Algorithm 3).

Import Vector Machine

Algorithm 3:

1. Start with a large regularization parameter λ.



2. Let $\mathcal{S} = \emptyset$, $\mathcal{L} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, $k = 1$. Let $\mathbf{a}^{(0)} = \mathbf{0}$, hence $\mathbf{z} = 2\mathbf{y}/n$.

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

4. Decrease λ to a smaller value.

5. Repeat steps (3) and (4), starting with $\mathcal{S} = \{\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_k}\}$.

Simulation Results

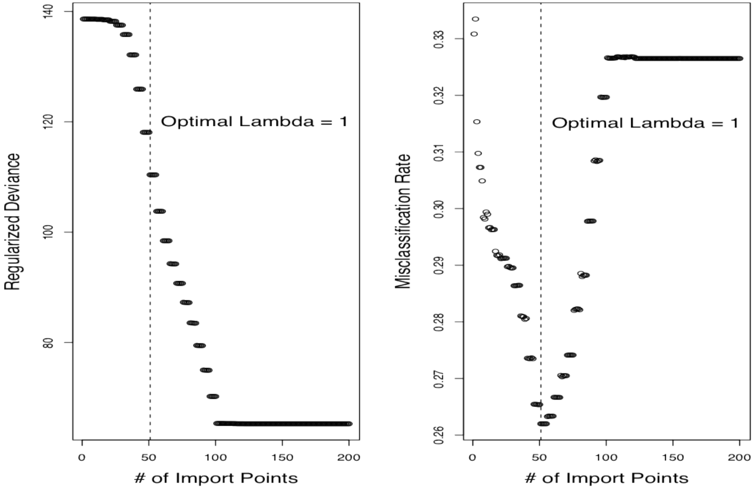


Figure 3: *Radial kernel is used.* $n = 200$, $\sigma^2 = 0.7$, $\Delta k = 3$, $\epsilon = 0.001$, λ *decreases from* e^{10} *to* e^{-10}. *The minimum misclassification rate* 0.262 *is found to correspond to* $\lambda = 1$.

Simulation Results

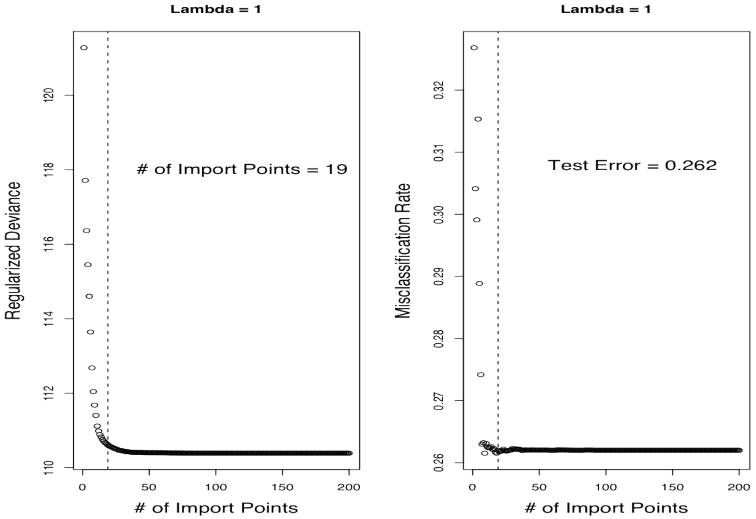


Figure 4: *Radial kernel is used.* $n = 200$, $\sigma^2 = 0.7$, $\Delta k = 3$, $\epsilon = 0.001$, $\lambda = 1$. *The stopping criterion is satisfied when* $|\mathcal{S}| = 19$.

Real Data Results

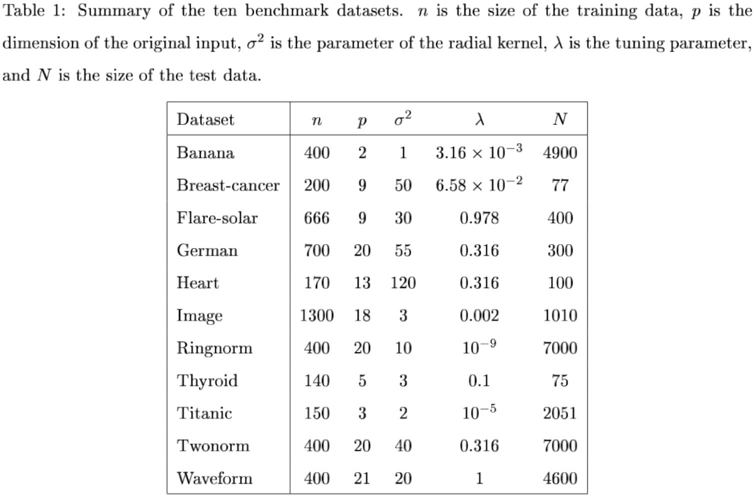


Table 1: Summary of the ten benchmark datasets. n is the size of the training data, p is the dimension of the original input, σ^2 is the parameter of the radial kernel, λ is the tuning parameter, and N is the size of the test data.

Dataset	n	p	σ^2	λ	N
Banana	400	2	1	3.16×10^{-3}	4900
Breast-cancer	200	9	50	6.58×10^{-2}	77
Flare-solar	666	9	30	0.978	400
German	700	20	55	0.316	300
Heart	170	13	120	0.316	100
Image	1300	18	3	0.002	1010
Ringnorm	400	20	10	10^{-9}	7000
Thyroid	140	5	3	0.1	75
Titanic	150	3	2	10^{-5}	2051
Twonorm	400	20	40	0.316	7000
Waveform	400	21	20	1	4600

Real Data Results

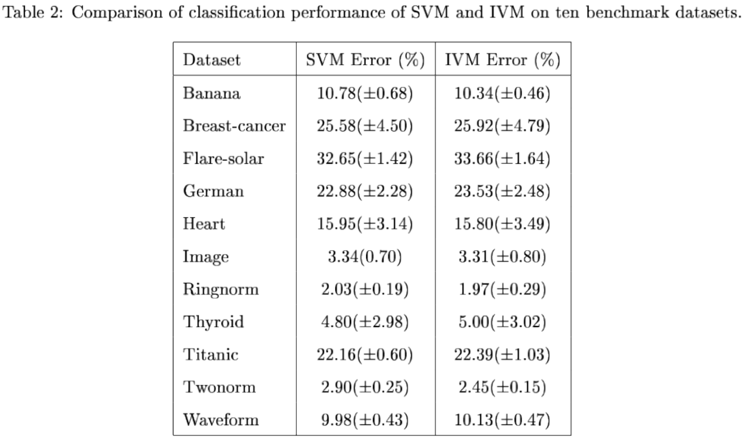


Table 2: Comparison of classification performance of SVM and IVM on ten benchmark datasets.

Dataset	SVM Error (%)	IVM Error (%)
Banana	10.78(±0.68)	10.34(±0.46)
Breast-cancer	25.58(±4.50)	25.92(±4.79)
Flare-solar	32.65(±1.42)	33.66(±1.64)
German	22.88(±2.28)	23.53(±2.48)
Heart	15.95(±3.14)	15.80(±3.49)
Image	3.34(0.70)	3.31(±0.80)
Ringnorm	2.03(±0.19)	1.97(±0.29)
Thyroid	4.80(±2.98)	5.00(±3.02)
Titanic	22.16(±0.60)	22.39(±1.03)
Twonorm	2.90(±0.25)	2.45(±0.15)
Waveform	9.98(±0.43)	10.13(±0.47)

Real Data Results

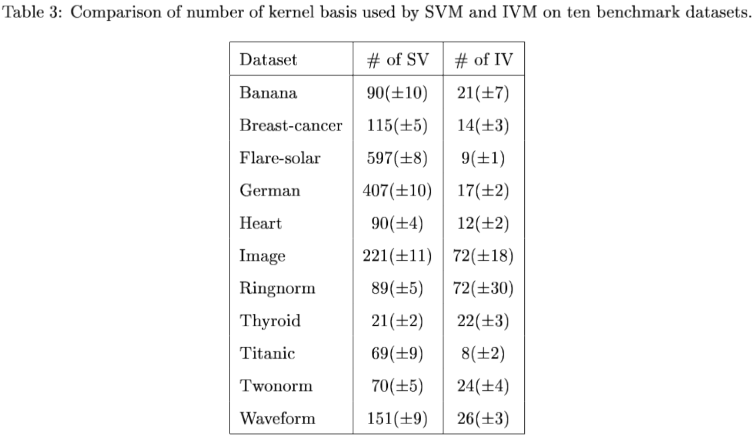


Table 3: Comparison of number of kernel basis used by SVM and IVM on ten benchmark datasets.

Dataset	# of SV	# of IV
Banana	90(±10)	21(±7)
Breast-cancer	115(±5)	14(±3)
Flare-solar	597(±8)	9(±1)
German	407(±10)	17(±2)
Heart	90(±4)	12(±2)
Image	221(±11)	72(±18)
Ringnorm	89(±5)	72(±30)
Thyroid	21(±2)	22(±3)
Titanic	69(±9)	8(±2)
Twonorm	70(±5)	24(±4)
Waveform	151(±9)	26(±3)

Generalization to M-Class Case

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

$$
\begin{aligned}
p_1(\mathbf{x}) &= \frac{e^{f_1(\mathbf{x})}}{\sum_{c=1}^{C} e^{f_c(\mathbf{x})}}, \\
p_2(\mathbf{x}) &= \frac{e^{f_2(\mathbf{x})}}{\sum_{c=1}^{C} e^{f_c(\mathbf{x})}}, \\
&\vdots \\
p_C(\mathbf{x}) &= \frac{e^{f_C(\mathbf{x})}}{\sum_{c=1}^{C} e^{f_c(\mathbf{x})}},
\end{aligned}
$$

$$\sum_{c=1}^{C} f_c(\mathbf{x}) = 0.$$

Generalization to M-Class Case

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

$$\begin{aligned} H &= -\frac{1}{n}\sum_{i=1}^{n} \ln p_{y_i}(\mathbf{x}_i) + \frac{\lambda}{2}\|\mathbf{f}\|^2_{\mathcal{H}_K} \\ &= \frac{1}{n}\sum_{i=1}^{n} \left[-\mathbf{y}_i^T \mathbf{f}(\mathbf{x}_i) + \ln\left(e^{f_1(\mathbf{x}_i)} + \cdots + e^{f_C(\mathbf{x}_i)} \right) \right] + \frac{\lambda}{2}\|\mathbf{f}\|^2_{\mathcal{H}_K} \end{aligned}$$

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

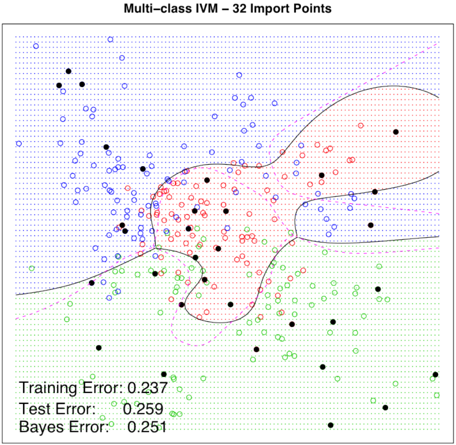


Figure 6: *Radial kernel is used.* $C = 3$, $n = 300$, $\lambda = 0.368$, $|\mathcal{S}| = 32$.

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.