*ICIC 2006 Presentation*

# Fast Kernel Classifier Construction Using Orthogonal Forward Selection to Minimise Leave-One-Out Misclassification Rate

**X. Hong**[‡], **S. Chen**[†] **and C.J. Harris**[†]

[‡] **Department of Cybernetics,**
**University of Reading, RG6 6AY, UK**

[†] **School of Electronics and Computer Science,**
**University of Southampton, SO17 1BJ, UK**

# Outline

- Existing RBF and kernel classifier construction methods, and motivations for the present work

- The proposed fast sparse kernel classifier construction method

- Experimental investigation of the proposed method and comparison with some existing techniques

# Overview of Existing Methods

- Nonlinear optimisation approach: Optimise all parameters (kernel centre vectors, variances or covariance matrices, and weights)
  - ✰ Very "sparse" (small size)
  - ✰ All problems associated with nonlinear optimisation
- Linear optimisation approach: Fix centres to training input data, and seek a "linear" subset model
  - Orthogonal least squares forward selection
    - ✰ Sparse, good performance, and efficient construction
    - ✰ Need to specify kernel variance (via cross validation)
  - Kernel modelling methods
    - ✰ Sparse (though not as sparse as OLS), good performance
    - ✰ Need to specify kernel variance and other kernel hyperparameters (via costly cross validation)

# Motivations

❑ How good a kernel classifier method:

- ✫ Generalisation performance
- ✫ Sparsity level or classifier's size
- ✫ Efficiency of classifier construction process

❑ Adopt OLS forward selection approach with improvements:

- ✫ Select kernels by directly optimising generalisation capability, i.e. use leave-one-out misclassification rate to select kernels
- ✫ This further enhances sparsity of resulting kernel classifier
- ✫ and yet keeps efficiency of OLS construction process

# Two-Class Kernel Classifier

- Consider constructing two-class kernel classifier $f(\mathbf{x}) : \Re^n \rightarrow \{1, -1\}$

$$\hat{y}(i) = \mathrm{sgn}(f(i)) \quad \text{with} \quad f(i) = \sum_{j=1}^{L} \theta_j p_j(\mathbf{x}(i))$$

  with training set $D_N = \{\mathbf{x}(i), y(i)\}_{i=1}^{N}$, where $y(i) \in \{1, -1\}$ is class label for $\mathbf{x}(i)$, $\hat{y}(i)$ estimated class label, and $p_j(\bullet)$ classifier's kernel

- Use each data $\mathbf{x}(i)$ as kernel centre, i.e. $L = N$, and define residual $\xi(i) = y(i) - f(i)$. Then kernel model over $D_N$ can be expressed as

$$\mathbf{y} = \mathbf{P}\boldsymbol{\theta} + \boldsymbol{\Xi}$$

  where $\boldsymbol{\Xi} = [\xi(1)\ \xi(2) \cdots \xi(N)]^T$, and $\boldsymbol{\theta} = [\theta_1\ \theta_2 \cdots \theta_L]^T$

- $\mathbf{P} = [\mathbf{p}_1\ \ \mathbf{p}_2 \cdots \mathbf{p}_L]$ is regression matrix with column $\mathbf{p}_j = [p_j(\mathbf{x}(1))\ p_j(\mathbf{x}(2)) \cdots p_j(\mathbf{x}(N))]^T$ and row $\mathbf{p}^T(i) = [p_1(i)\ p_2(i) \cdots p_L(i)]$

# Orthogonal Decomposition

- Let orthogonal decomposition $\mathbf{P} = \mathbf{W}\mathbf{A}$, where $\mathbf{A} = \{a_{ij}\}$ is $L \times L$ unit upper triangular matrix and $\mathbf{W}$ is $N \times L$ orthogonal matrix

$$\mathbf{W}^T\mathbf{W} = \text{diag}\{\kappa_1, \kappa_2, \cdots, \kappa_L\} \quad \text{with} \quad \kappa_j = \mathbf{w}_j^T\mathbf{w}_j$$

- Let $\mathbf{w}_j$ be $j$th column and $\mathbf{w}^T(i) = [w_1(i)\ w_2(i) \cdots w_L(i)]$ $i$th row of $\mathbf{W}$. Then kernel model can alternatively be expressed as

$$\mathbf{y} = \mathbf{W}\boldsymbol{\gamma} + \boldsymbol{\Xi}$$

in which $\boldsymbol{\gamma} = [\gamma_1\ \gamma_2 \cdots \gamma_L]^T = \mathbf{A}\boldsymbol{\theta}$

- Regularised OLS parameter estimate is

$$\gamma_j = \frac{\mathbf{w}_j^T\mathbf{y}}{\kappa_j + \lambda_j}$$

where $\lambda_j$ is small positive regularisation parameter

# Leave-One-Out Misclassification Rate

- Define signed decision variable $g(i) = y(i)f(i)$. Then misclassification rate over $D_N$ is computed by

$$\frac{1}{N}\sum_{i=1}^{N} \mathrm{Id}[g(i)] \quad \text{where} \quad \mathrm{Id}[v] = \begin{cases} 1, & v \le 0 \\ 0, & v > 0 \end{cases}$$

- Let $f_k^{(-i)}(\bullet)$ be k-term kernel classifier identified using $D_N$ but with its $i$th data point being removed. Test output of this $k$-term classifier at $i$th data point not used in training is $f_k^{(-i)}(i)$

- Leave-one-out signed decision variable is defined $g_k^{(-i)}(i) = y(i)f_k^{(-i)}(i)$, and leave-one-out misclassification rate is computed by

$$J_k = \frac{1}{N}\sum_{i=1}^{N} \mathrm{Id}[g_k^{(-i)}(i)]$$

# Efficient Computation

- Leave-one-out misclassification rate $J_k$ is a measure of the classifier's generalisation capability
- $J_k$ can be computed efficiently, as leave-one-out signed decision variable

$$g_k^{(-i)}(i) = y(i) f_k^{(-i)}(i) = \frac{\alpha_k(i)}{\beta_k(i)}$$

  can be computed recursively

$$\alpha_k(i) = \alpha_{k-1}(i) + \gamma_k w_k(i) y(i) - \frac{w_k^2(i)}{\kappa_k + \lambda_j}$$

$$\beta_k(i) = \beta_{k-1}(i) - \frac{w_k^2(i)}{\kappa_k + \lambda_j}$$

  where $w_k(i)$ is $i$th element of $\mathbf{w}_k$
- Proposed algorithm selects kernels one by one by minimising $J_k$

# Proposed Algorithm

1. Initialise $\alpha_0(i) = 0$ and $\beta_0(i) = 1$ for $1 \le i \le N$
2. At $k$th step where $k \ge 1$, for $1 \le l \le L$, $l \ne l_1, \cdots, l \ne l_{k-1}$, compute

$$a_{jk}^{(l)} = \begin{cases} \dfrac{\mathbf{w}_j^T \mathbf{p}_l}{\mathbf{w}_j^T \mathbf{w}_j}, & 1 \le j < k, \\ 1, & j = k, \end{cases} \qquad \mathbf{w}_k^{(l)} = \begin{cases} \mathbf{p}_l, & k = 1, \\ \mathbf{p}_l - \sum_{j=1}^{k-1} a_{jk}^{(l)} \mathbf{w}_j, & k \ge 2, \end{cases} \qquad \gamma_k^{(l)} = \frac{(\mathbf{w}_k^{(l)})^T \mathbf{y}}{\kappa_k^{(l)} + \lambda},$$

$$\alpha_k^{(l)}(i) = \alpha_{k-1}(i) + \gamma_k^{(l)} w_k^{(l)}(i) y(i) - \frac{[w_k^{(l)}(i)]^2}{\kappa_k^{(l)} + \lambda}, \quad \beta_k^{(l)}(i) = \beta_{k-1}(i) - \frac{[w_k^{(l)}(i)]^2}{\kappa_k^{(l)} + \lambda}, \quad g_k^{(-i,l)}(i) = \frac{\alpha_k^{(l)}(i)}{\beta_k^{(l)}(i)},$$

for $1 \le i \le N$, and

$$J_k^{(l)} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{Id}[g_k^{(-i,l)}(i)].$$

Find

$$l_k = \arg[\min\{J_k^{(l)}, 1 \le l \le L, l \ne l_1, \cdots, l \ne l_{k-1}\}]$$

and select $J_k = J_k^{(l_k)}$, $a_{jk} = a_{jk}^{(l_k)}$ for $1 \le j \le k$, $\alpha_k(i) = \alpha_k^{(l_k)}(i)$ and $\beta_k(i) = \beta_k^{(l_k)}(i)$ for $1 \le i \le N$, and

$$\mathbf{w}_k = \mathbf{w}_k^{(l_k)} = \begin{cases} \mathbf{p}_{l_k}, & k = 1, \\ \mathbf{p}_{l_k} - \sum_{j=1}^{k-1} a_{jk} \mathbf{w}_j, & k \ge 2. \end{cases}$$

3. Procedure is terminated when $J_k \ge J_{k-1}$. Otherwise, set $k = k + 1$, and go to step 2

# Breast Cancer Data Set

Average classification test error rate in % over 100 realizations

| | Misclassification rate | Model Size |
|---|---|---|
| RBF | $27.6 \pm 4.7$ | 5 |
| Adaboost with RBF | $30.4 \pm 4.7$ | 5 |
| AdaBoost$_{Reg}$ | $26.5 \pm 4.5$ | 5 |
| LP$_{Reg}$-AdaBoost | $26.8 \pm 6.1$ | 5 |
| QP$_{Reg}$-AdaBoost | $25.9 \pm 4.6$ | 5 |
| SVM with RBF kernel | $26.0 \pm 4.7$ | not available |
| **Proposed** | $25.74 \pm 5$ | $6 \pm 2$ |

Data and first 6 results from:

`http://ida.first.fhg.de/projects/bench/benchmarks.htm`

# Diabetis Data Set

Average classification test error rate in % over 100 realizations

| | Misclassification rate | Model Size |
|---|---|---|
| RBF | $24.3 \pm 1.9$ | 15 |
| Adaboost with RBF | $26.5 \pm 2.3$ | 15 |
| $\text{AdaBoost}_{Reg}$ | $23.8 \pm 1.8$ | 15 |
| $\text{LP}_{Reg}$-AdaBoost | $24.1 \pm 1.9$ | 15 |
| $\text{QP}_{Reg}$-AdaBoost | $25.4 \pm 2.2$ | 15 |
| SVM with RBF kernel | $23.5 \pm 1.7$ | not available |
| **Proposed** | $23.0 \pm 1.7$ | $6 \pm 1$ |

Data and first 6 results from:

`http://ida.first.fhg.de/projects/bench/benchmarks.htm`

# Heart Data Set

Average classification test error rate in % over 100 realizations

| | Misclassification rate | Model Size |
|---|---|---|
| RBF | $17.6 \pm 3.3$ | 4 |
| Adaboost with RBF | $20.3 \pm 3.4$ | 4 |
| AdaBoost$_{Reg}$ | $16.5 \pm 3.5$ | 4 |
| LP$_{Reg}$-AdaBoost | $17.5 \pm 3.5$ | 4 |
| QP$_{Reg}$-AdaBoost | $17.2 \pm 3.4$ | 4 |
| SVM with RBF kernel | $16.0 \pm 3.3$ | not available |
| **Proposed** | $15.8 \pm 3.7$ | $10 \pm 3$ |

Data and first 6 results from:

`http://ida.first.fhg.de/projects/bench/benchmarks.htm`

# Conclusions

- A novel construction algorithm has been proposed for kernel classifiers
  - ✰ Kernels are selected in a computationally efficient orthogonal forward selection procedure
  - ✰ Kernels are selected by minimising leave-one-out misclassification rate, a measure of generalisation capability
- Several examples have shown that proposed method compares favourably with existing state-of-the-art

# THANK YOU.

S. Chen wish to thank the support of the United Kingdom Royal Academy of Engineering