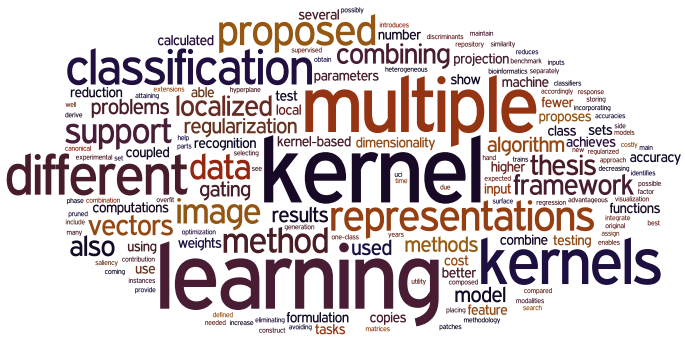
Localized Multiple Kernel Algorithms for Machine Learning

Mehmet Gönen
gonen@boun.edu.tr

Ph.D. Thesis Presentation
Supervisor: Prof. Ethem Alpaydın

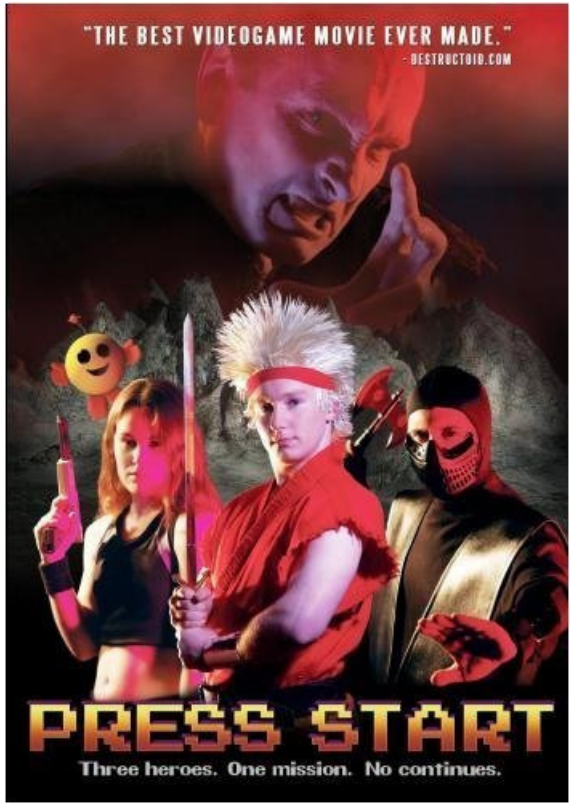
Department of Computer Engineering
Boğaziçi University, İstanbul, Turkey

May 25, 2010



Outline

1. Introduction
2. Multiple Kernel Learning
3. Localized Multiple Kernel Learning
4. Local Projection Kernels
5. Experiments
6. Conclusions



Basics

- Observed data with outputs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$
- Estimate output for an unseen test instance $\boldsymbol{x}$

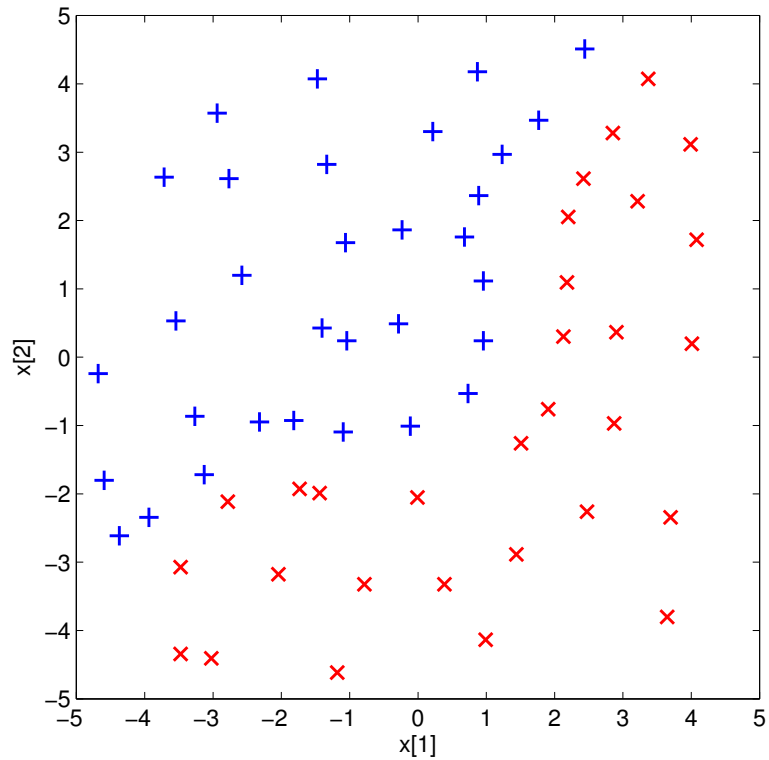
Basics

- Observed data with outputs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$
- Estimate output for an unseen test instance $\boldsymbol{x}$
- Sample problems

Problem	Input	Features	Output
Digit recognition		Pixel values	{0,1,. . . ,9}
Disease diagnosis	`ATCGGT...TTA`	Nucleotide bases	{Healthy, Not healthy}
Face recognition		Pixel values	{Person1, Person2, . . . }

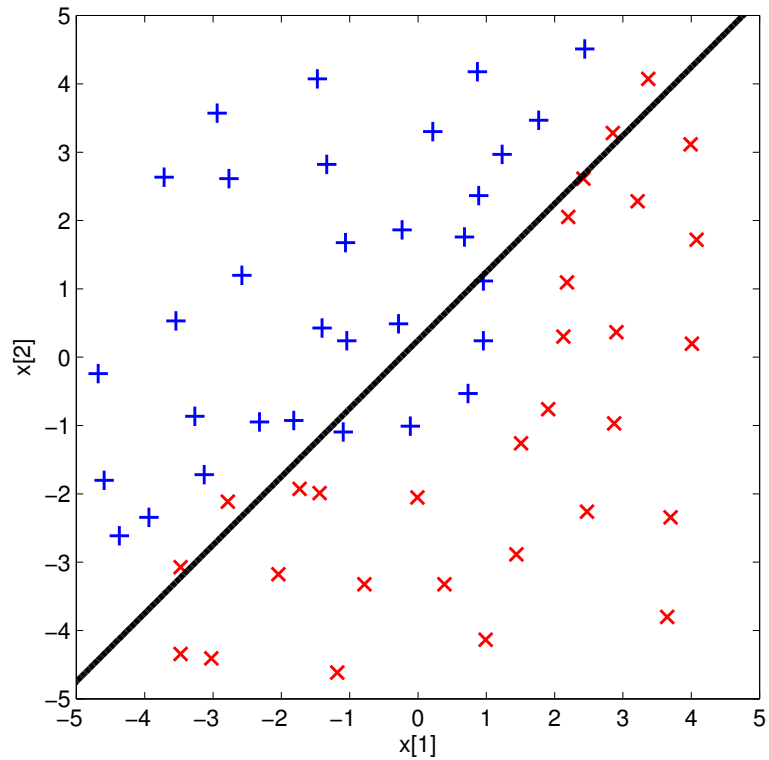
Support Vector Machines

- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \{-1, +1\}$



Support Vector Machines

- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \{-1, +1\}$
- $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$



Support Vector Machines

- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \{-1, +1\}$
- $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$
- Primal optimization problem (Vapnik, 1998)

$$\begin{aligned} \text{min.} \quad & \underbrace{\frac{1}{2}\|\boldsymbol{w}\|_2^2}_{\text{regularization}} + \underbrace{C \sum_{i=1}^N \xi_i}_{\text{soft error}} \\ \text{w.r.t.} \quad & \boldsymbol{w} \in \mathbb{R}^D, \boldsymbol{\xi} \in \mathbb{R}_+^N, b \in \mathbb{R} \\ \text{s.t.} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i \qquad \forall i \end{aligned}$$

Support Vector Machines

- Dual optimization problem

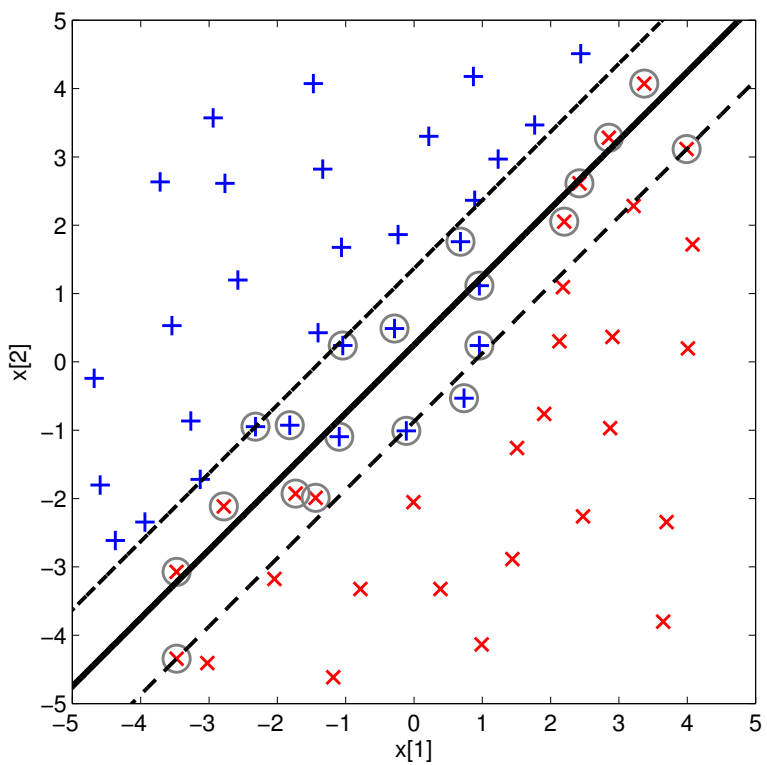
$$
\begin{aligned}
\text{max.} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_i y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\
\text{w.r.t.} \quad & \boldsymbol{\alpha} \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& C \geq \alpha_i \geq 0 \qquad \forall i
\end{aligned}
$$

Support Vector Machines

- Dual optimization problem

$$\begin{aligned}
\text{max.} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_i y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\
\text{w.r.t.} \quad & \boldsymbol{\alpha} \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& C \geq \alpha_i \geq 0 \qquad \forall i
\end{aligned}$$

- $\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i$
- $f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b$

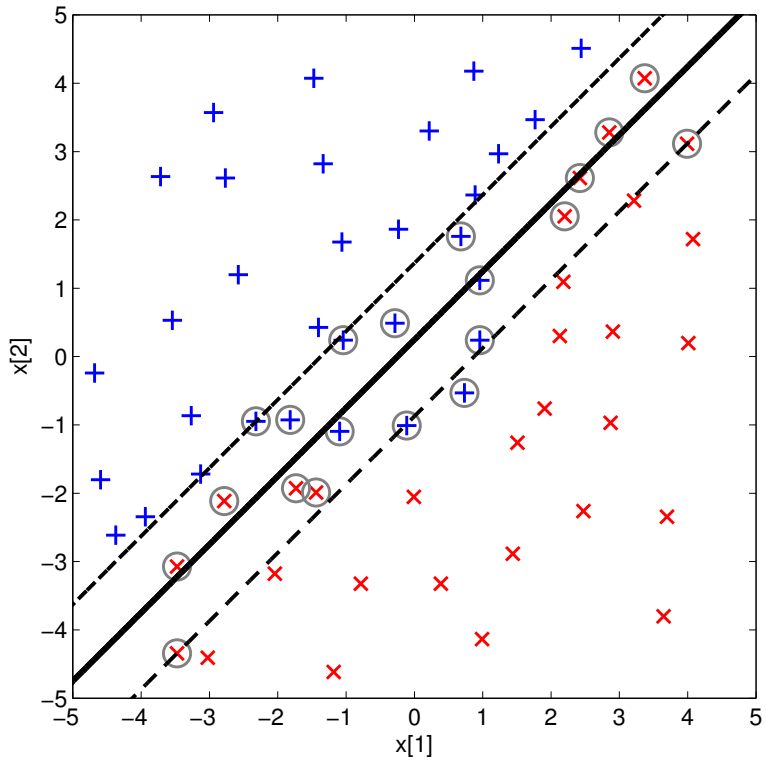


Kernel Functions

- Define a mapping function $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^S$
- Replace $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ with $\underbrace{\langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle}_{k(\boldsymbol{x}_i, \boldsymbol{x}_j)}$
- Choosing kernel function $k(\cdot, \cdot)$ and regularization parameter C

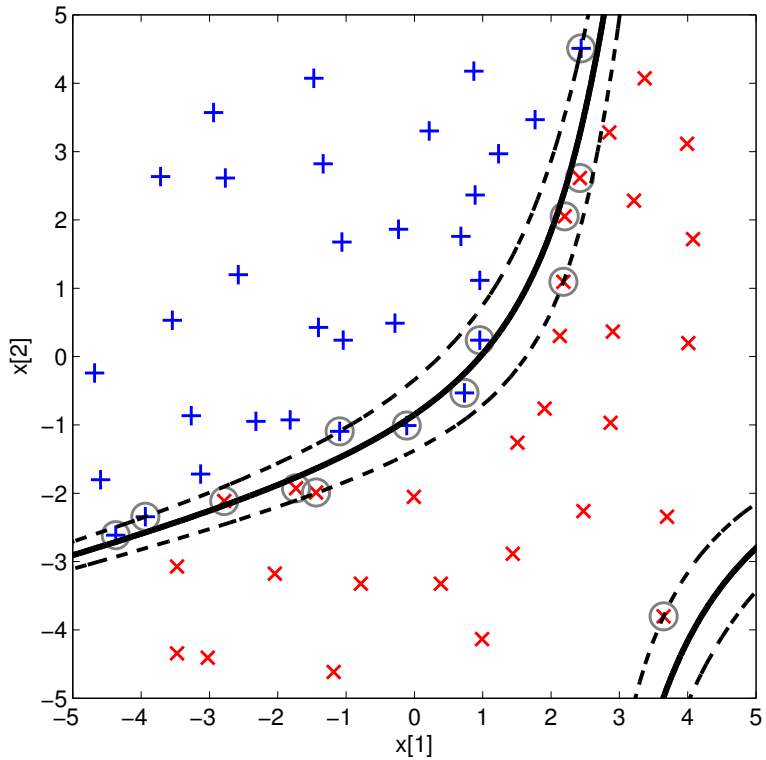
Kernel Functions

- Define a mapping function $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^S$
- Replace $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ with $\underbrace{\langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle}_{k(\boldsymbol{x}_i, \boldsymbol{x}_j)}$
- Choosing kernel function $k(\cdot, \cdot)$ and regularization parameter C
- Common kernel functions
  - Linear kernel
    $k_L(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$



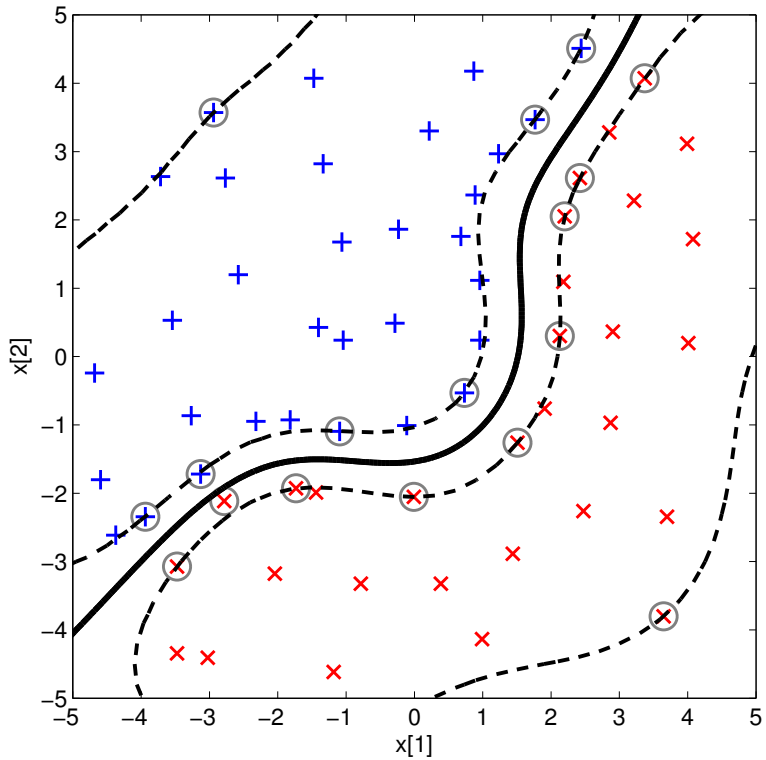
Kernel Functions

- Define a mapping function $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^S$
- Replace $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ with $\underbrace{\langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle}_{k(\boldsymbol{x}_i, \boldsymbol{x}_j)}$
- Choosing kernel function $k(\cdot, \cdot)$ and regularization parameter C
- Common kernel functions
  - Linear kernel
    $k_L(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$
  - Polynomial kernel
    $k_P(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + 1)^q$



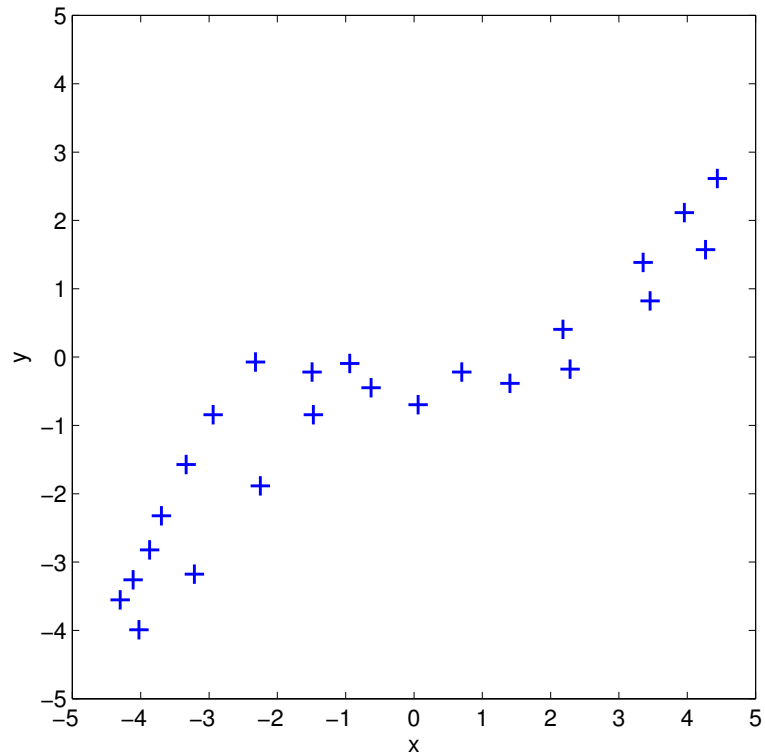
Kernel Functions

- Define a mapping function $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^S$
- Replace $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ with $\underbrace{\langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle}_{k(\boldsymbol{x}_i, \boldsymbol{x}_j)}$
- Choosing kernel function $k(\cdot, \cdot)$ and regularization parameter C
- Common kernel functions
  - Linear kernel
    $k_L(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$
  - Polynomial kernel
    $k_P(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + 1)^q$
  - Gaussian kernel
    $k_G(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 / s^2\right)$



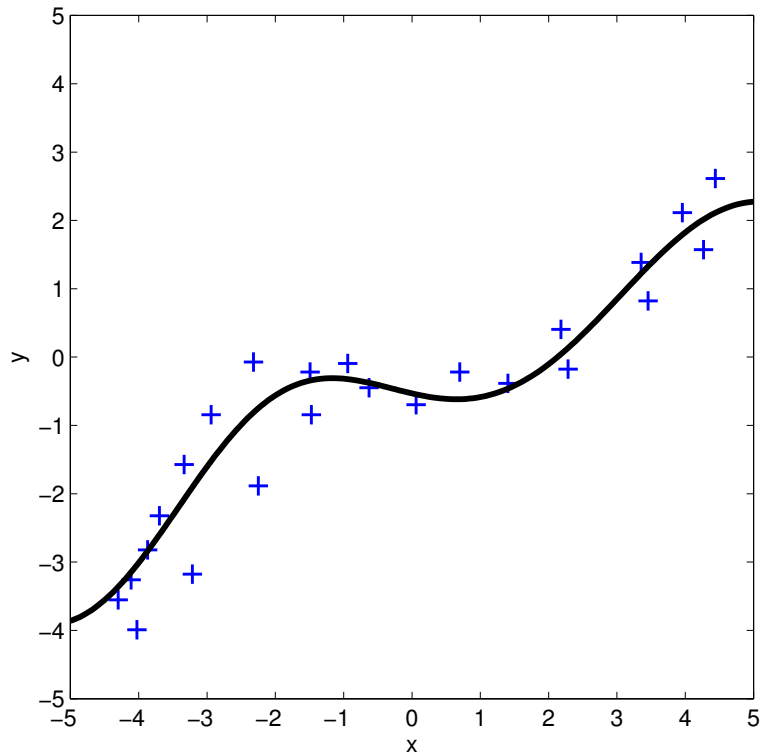
Regression Support Vector Machines

- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$



Regression Support Vector Machines

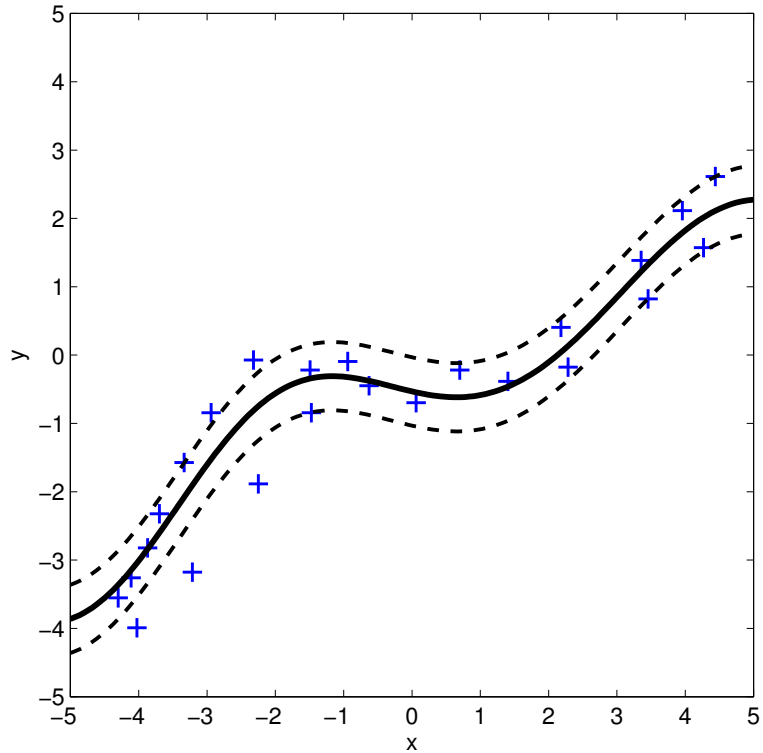
- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$
- $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle + b$ where $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^S$



Regression Support Vector Machines

- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$
- $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle + b$ where $\Phi \colon \mathbb{R}^D \rightarrow \mathbb{R}^S$
- ϵ-insensitive error function (Vapnik, 1998)

$$e(y, f(\boldsymbol{x})) = \begin{cases} 0 & \text{if } |y - f(\boldsymbol{x})| \leq \epsilon \\ |y - f(\boldsymbol{x})| & \text{otherwise} \end{cases}$$



Regression Support Vector Machines

- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$
- $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle + b$ where $\Phi \colon \mathbb{R}^D \rightarrow \mathbb{R}^S$
- ϵ-insensitive error function (Vapnik, 1998)

$$e(y, f(\boldsymbol{x})) = \begin{cases} 0 & \text{if } |y - f(\boldsymbol{x})| \leq \epsilon \\ |y - f(\boldsymbol{x})| & \text{otherwise} \end{cases}$$

- Primal optimization problem

$$\begin{aligned} \text{min.} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^N (\xi_i^+ + \xi_i^-) \\ \text{w.r.t.} \quad & \boldsymbol{w} \in \mathbb{R}^S, \boldsymbol{\xi}^+ \in \mathbb{R}_+^N, \boldsymbol{\xi}^- \in \mathbb{R}_+^N, b \in \mathbb{R} \\ \text{s.t.} \quad & \epsilon + \xi_i^+ \geq y_i - \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i) \rangle - b \quad \forall i \\ & \epsilon + \xi_i^- \geq \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i) \rangle + b - y_i \quad \forall i \end{aligned}$$

Regression Support Vector Machines

- Dual optimization problem

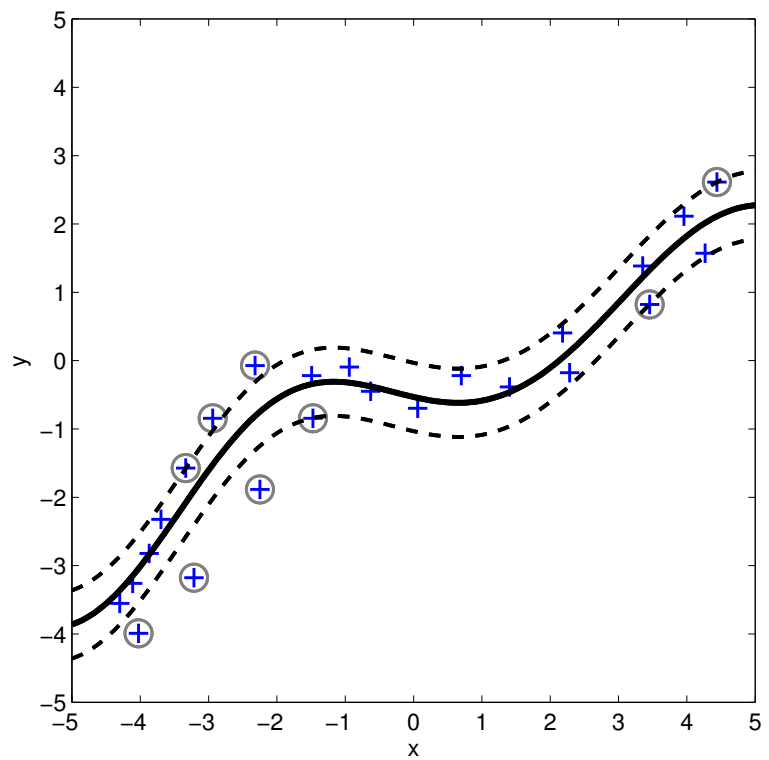
$$\begin{aligned}
\text{max.} \quad & \sum_{i=1}^{N} y_i(\alpha_i^+ - \alpha_i^-) - \epsilon \sum_{i=1}^{N}(\alpha_i^+ + \alpha_i^-) \\
& - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)k(\boldsymbol{x}_i, \boldsymbol{x}_j) \\
\text{w.r.t.} \quad & \boldsymbol{\alpha}^+ \in \mathbb{R}_+^N, \boldsymbol{\alpha}^- \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-) = 0 \\
& C \geq \alpha_i^+, \alpha_i^- \geq 0 \qquad \forall i
\end{aligned}$$

Regression Support Vector Machines

- Dual optimization problem

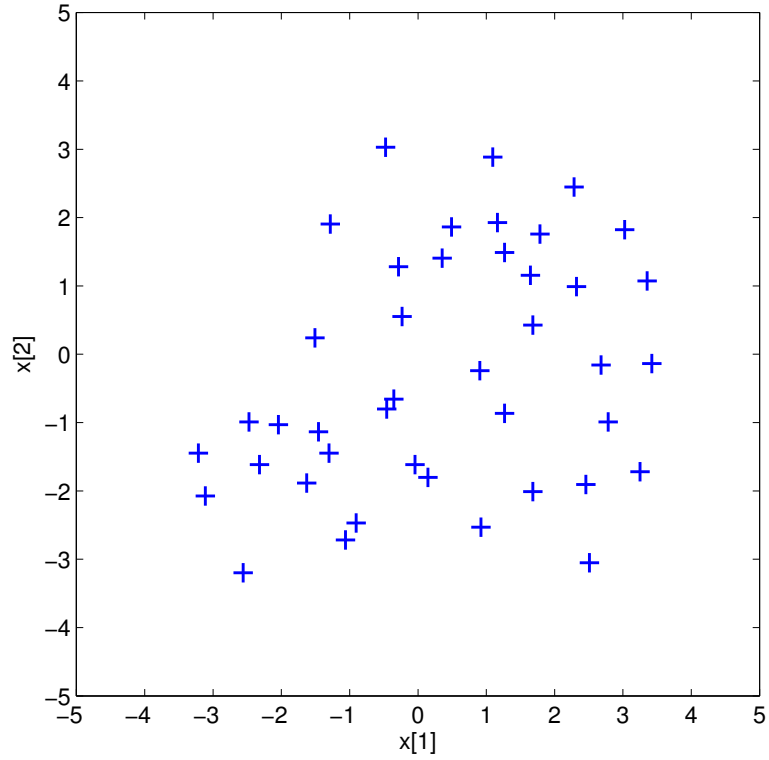
$$\text{max.} \quad \sum_{i=1}^{N} y_i(\alpha_i^+ - \alpha_i^-) - \epsilon \sum_{i=1}^{N}(\alpha_i^+ + \alpha_i^-)$$
$$- \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$
$$\text{w.r.t.} \quad \boldsymbol{\alpha}^+ \in \mathbb{R}_+^N, \boldsymbol{\alpha}^- \in \mathbb{R}_+^N$$
$$\text{s.t.} \quad \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-) = 0$$
$$C \geq \alpha_i^+, \alpha_i^- \geq 0 \qquad \forall i$$

- $\boldsymbol{w} = \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-)\Phi(\boldsymbol{x}_i)$
- $f(\boldsymbol{x}) = \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-)k(\boldsymbol{x}_i, \boldsymbol{x}) + b$



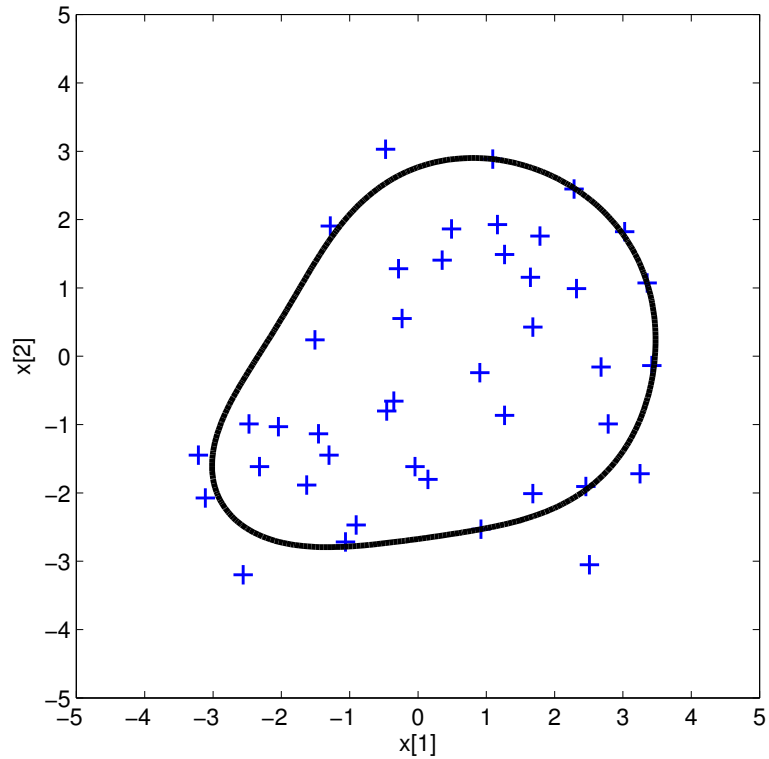
One-Class Classification Support Vector Machines

- $\{\boldsymbol{x}_i\}_{i=1}^N$ where $\boldsymbol{x}_i \in \mathbb{R}^D$



One-Class Classification Support Vector Machines

- $\{\boldsymbol{x}_i\}_{i=1}^N$ where $\boldsymbol{x}_i \in \mathbb{R}^D$
- $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle + b$ where $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^S$



One-Class Classification Support Vector Machines

- $\{\boldsymbol{x}_i\}_{i=1}^N$ where $\boldsymbol{x}_i \in \mathbb{R}^D$
- $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle + b$ where $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^S$
- Primal optimization problem (Schölkopf and Smola, 2002)

$$
\begin{aligned}
\text{min.} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^N \xi_i + b \\
\text{w.r.t.} \quad & \boldsymbol{w} \in \mathbb{R}^S, \boldsymbol{\xi} \in \mathbb{R}_+^N, b \in \mathbb{R} \\
\text{s.t.} \quad & \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i) \rangle + b + \xi_i \geq 0 \qquad \forall i
\end{aligned}
$$

One-Class Classification Support Vector Machines

- Dual optimization problem

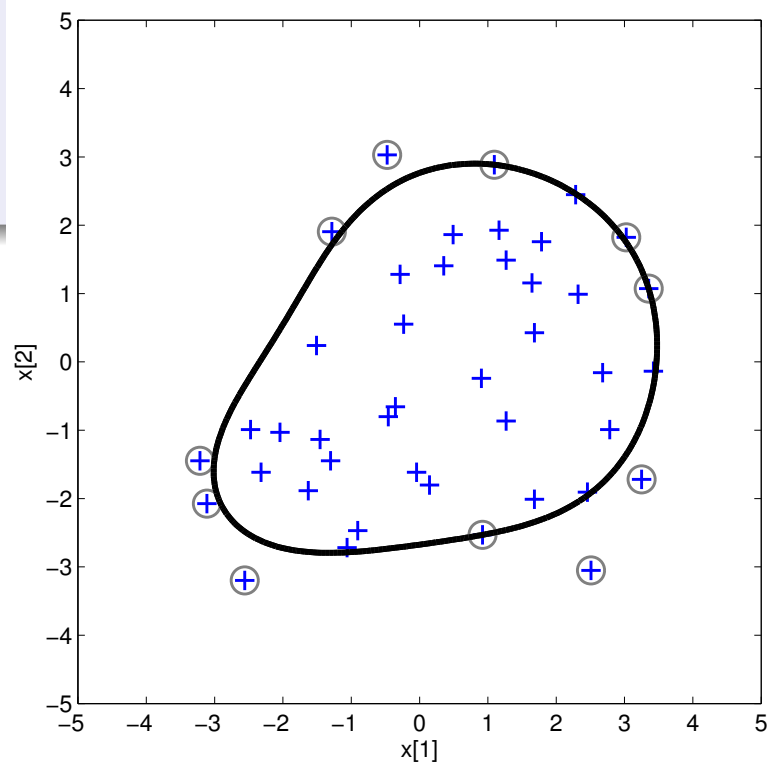
$$
\begin{aligned}
\text{max.} \quad & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_i k(\boldsymbol{x}_i,\boldsymbol{x}_j) \\
\text{w.r.t.} \quad & \boldsymbol{\alpha}\in\mathbb{R}_+^N \\
\text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i = 1 \\
& C \geq \alpha_i \geq 0 \qquad \forall i
\end{aligned}
$$

One-Class Classification Support Vector Machines

- Dual optimization problem

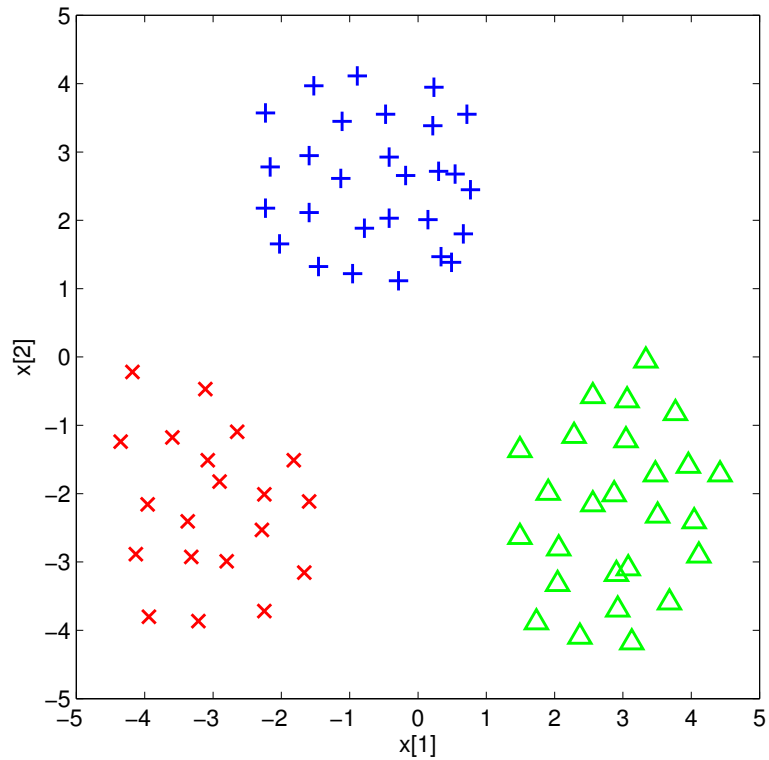
$$\begin{aligned} \text{max.} \quad & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_i k(\boldsymbol{x}_i,\boldsymbol{x}_j) \\ \text{w.r.t.} \quad & \boldsymbol{\alpha}\in\mathbb{R}_+^N \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i = 1 \\ & C \geq \alpha_i \geq 0 \qquad \forall i \end{aligned}$$

- $\boldsymbol{w} = \sum_{i=1}^{N}\alpha_i\Phi(\boldsymbol{x}_i)$
- $f(\boldsymbol{x}) = \sum_{i=1}^{N}\alpha_i k(\boldsymbol{x}_i,\boldsymbol{x}) + b$



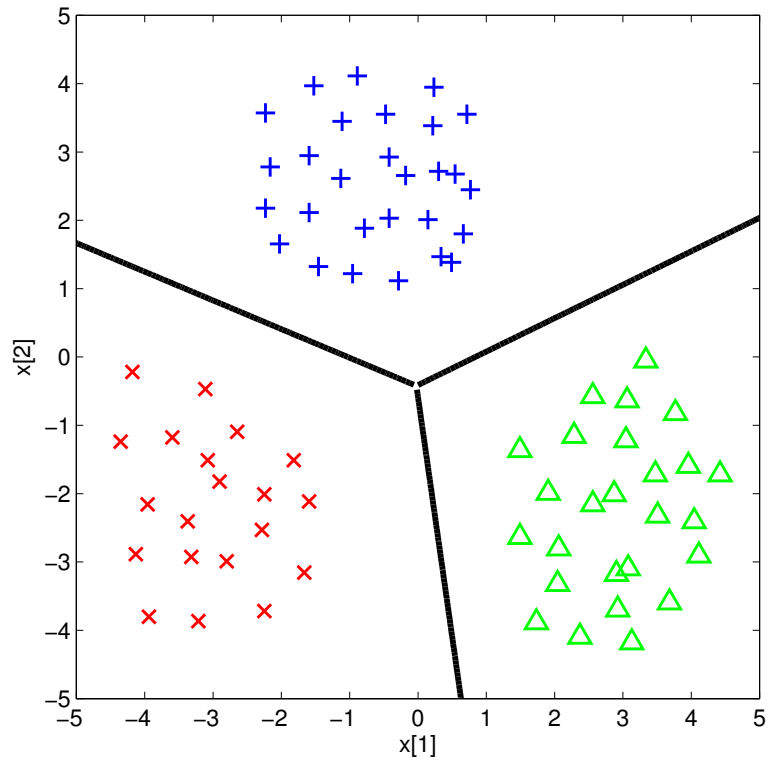
Multiclass Classification Support Vector Machines

- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \{1, \ldots, K\}$



Multiclass Classification Support Vector Machines

- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \{1, \ldots, K\}$
- $f^l(\boldsymbol{x}) = \langle \boldsymbol{w}^l, \Phi(\boldsymbol{x}) \rangle + b^l \quad \forall l$ where $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^S$



Multiclass Classification Support Vector Machines

- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $y_i \in \{1, \ldots, K\}$
- $f^l(\boldsymbol{x}) = \langle \boldsymbol{w}^l, \Phi(\boldsymbol{x}) \rangle + b^l \quad \forall l$ where $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^S$
- Primal optimization problem (Vapnik, 1998; Weston and Watkins, 1998; Bredensteiner and Bennett, 1999)

$$
\begin{aligned}
\text{min.} \quad & \frac{1}{2} \sum_{l=1}^{K} \|\boldsymbol{w}^l\|_2^2 + C \sum_{i=1}^{N} \sum_{l=1}^{K} \xi_i^l \\
\text{w.r.t.} \quad & \boldsymbol{w}^l \in \mathbb{R}^S, \boldsymbol{\xi}^l \in \mathbb{R}_+^N, b^l \in \mathbb{R} \\
\text{s.t.} \quad & f^{y_i}(\boldsymbol{x}_i) - f^l(\boldsymbol{x}_i) \geq 2 - \xi_i^l \quad \forall (i, l \neq y_i) \\
& \xi_i^{y_i} = 0 \quad \forall i
\end{aligned}
$$

Multiclass Classification Support Vector Machines

- Dual optimization problem

$$\text{max.} \quad 2\sum_{i=1}^{N}\sum_{l=1}^{K}\alpha_i^l - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\delta_{y_i}^{y_j}A_iA_j - \sum_{l=1}^{K}\alpha_i^l(2\alpha_j^{y_i} - \alpha_j^l)\right)k(\boldsymbol{x}_i,\boldsymbol{x}_j)$$

$$\text{w.r.t.} \quad \boldsymbol{\alpha}^l \in \mathbb{R}_+^N$$

$$\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i^l - \sum_{i=1}^{N}\delta_{y_i}^l A_i = 0 \qquad \forall l \qquad \text{where } A_i = \sum_{l=1}^{K}\alpha_i^l$$

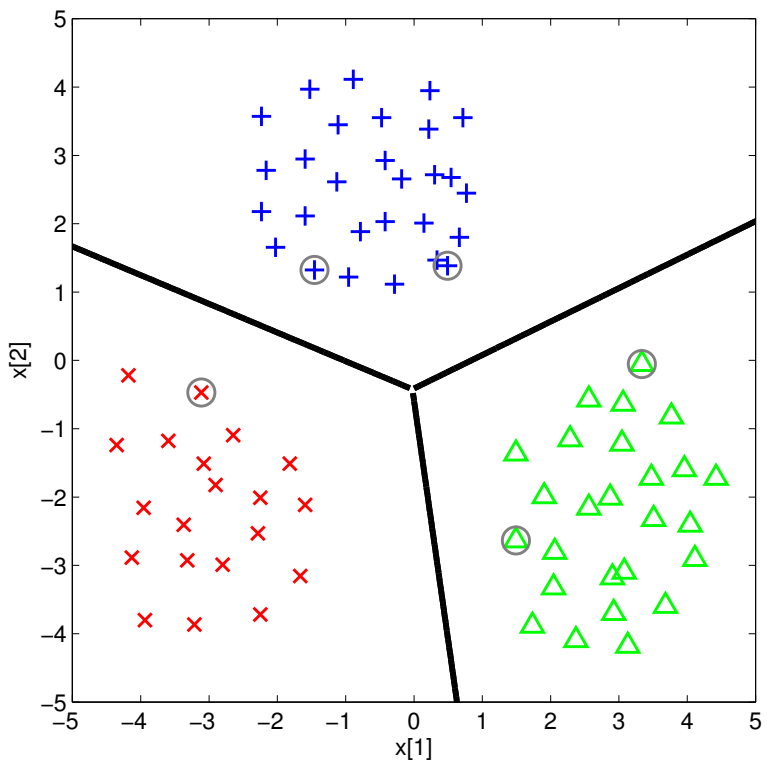
$$(1-\delta_{y_i}^l)C \geq \alpha_i^l \geq 0 \qquad \forall(i,l)$$

Multiclass Classification Support Vector Machines

- Dual optimization problem

$$\begin{aligned}
\text{max.} \quad & 2\sum_{i=1}^{N}\sum_{l=1}^{K}\alpha_i^l - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\delta_{y_i}^{y_j}A_iA_j - \sum_{l=1}^{K}\alpha_i^l(2\alpha_j^{y_i} - \alpha_j^l)\right)k(\boldsymbol{x}_i,\boldsymbol{x}_j) \\
\text{w.r.t.} \quad & \boldsymbol{\alpha}^l \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i^l - \sum_{i=1}^{N}\delta_{y_i}^l A_i = 0 \qquad \forall l \\
& (1-\delta_{y_i}^l)C \geq \alpha_i^l \geq 0 \qquad \forall (i,l)
\end{aligned}$$

- $\boldsymbol{w}^l = \sum_{i=1}^{N}(\delta_{y_i}^l A_i - \alpha_i^l)\Phi(\boldsymbol{x}_i)$
- $f^l(\boldsymbol{x}) = \sum_{i=1}^{N}(\delta_{y_i}^l A_i - \alpha_i^l)k(\boldsymbol{x}_i,\boldsymbol{x}) + b^l$



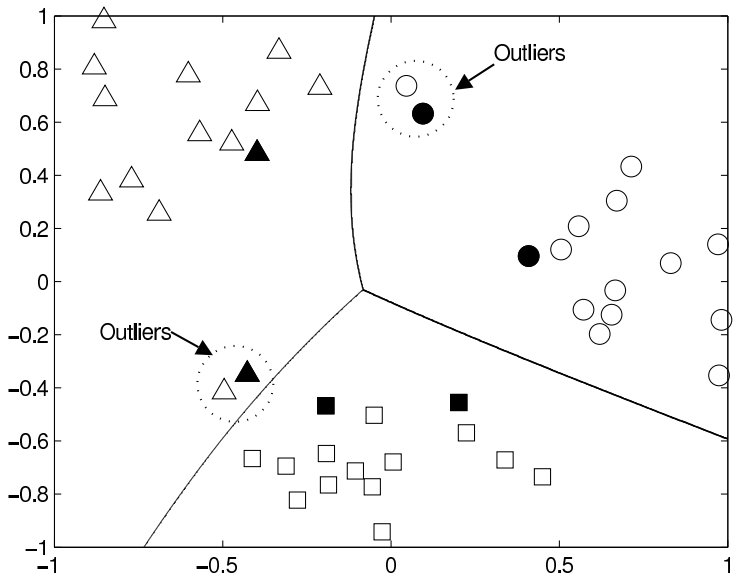
Posterior Probability Support Vector Machines

- Soft labels instead of hard labels (Tao et al., 2005)
  $\hat{y}_i = 2\Pr(+1|\boldsymbol{x}_i) - 1 \qquad \forall i$

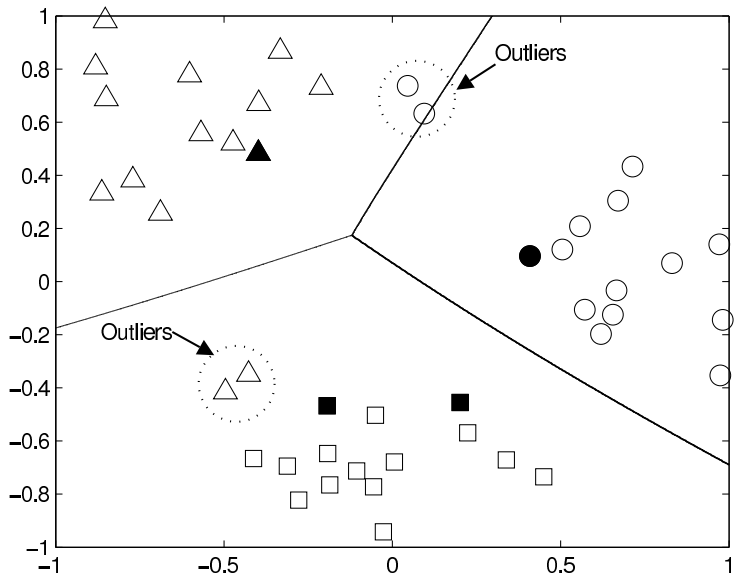
- $\hat{y}_i(\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i) \rangle + b) \geq \hat{y}_i^2 - \hat{y}_i^2 \xi_i \qquad \forall i$

Posterior Probability Support Vector Machines

- Soft labels instead of hard labels (Tao et al., 2005)
  $\hat{y}_i = 2\Pr(+1|\boldsymbol{x}_i) - 1 \qquad \forall i$
- $\hat{y}_i(\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle + b) \geq \hat{y}_i^2 - \hat{y}_i^2\xi_i \qquad \forall i$
- Extended to multiclass classification (Gönen et al., 2008)
  $f^{y_i}(\boldsymbol{x}_i) - f^{l}(\boldsymbol{x}_i) \geq 2(\Pr(y_i|\boldsymbol{x}_i) - \Pr(l|\boldsymbol{x}_i)) - \xi_i^l \qquad \forall (i, l \neq y_i)$

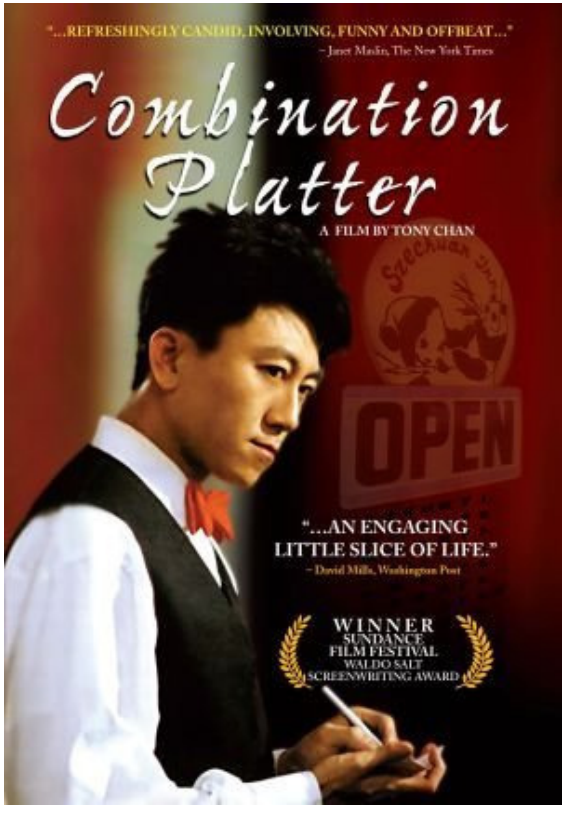


MCSVM

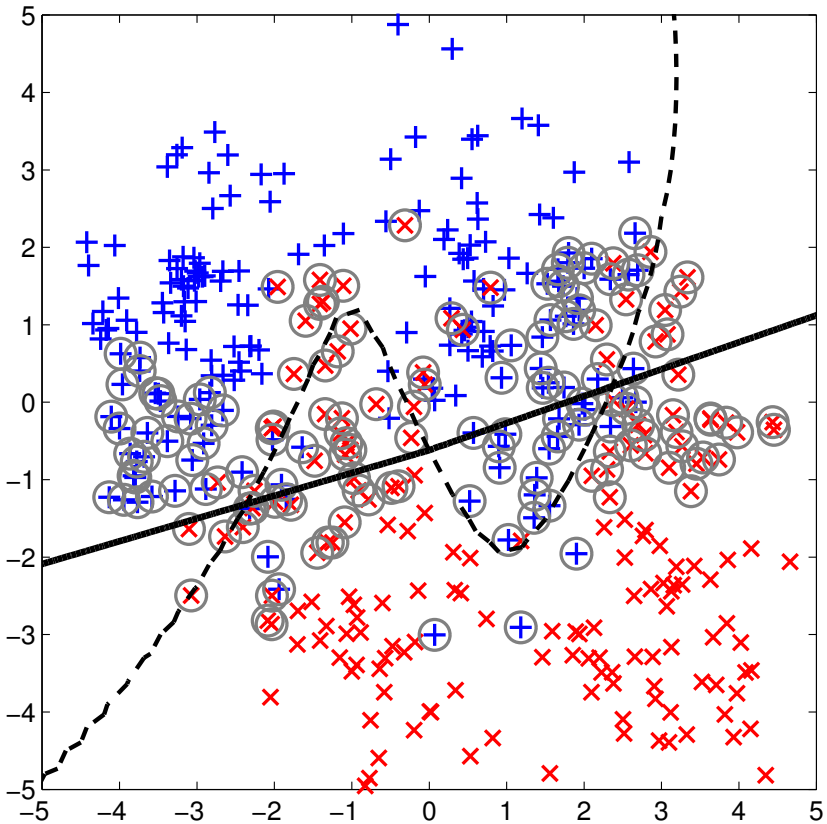


MCPPSVM

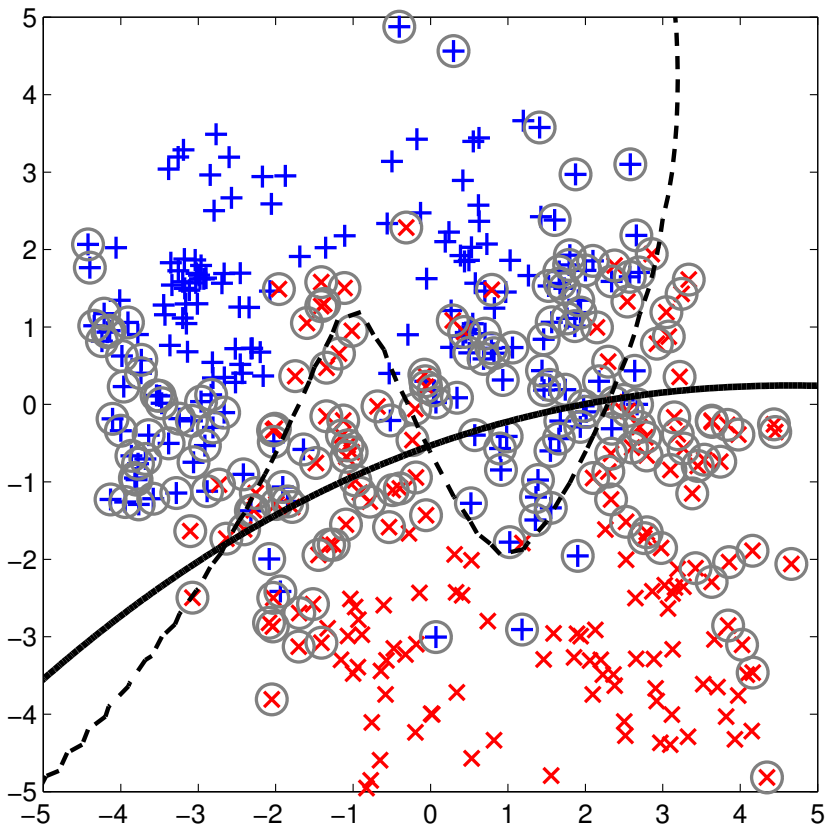
Outline



Multiple Kernel Learning

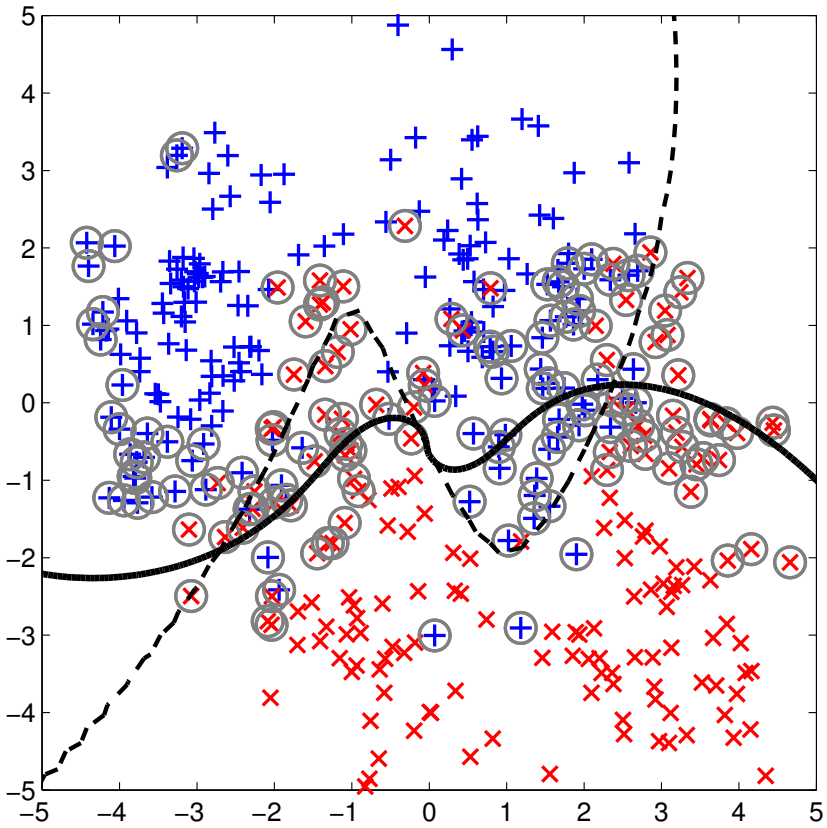


SVM (k_L)



SVM (k_P)

Multiple Kernel Learning



MKL (k_L-k_P)

- Bayes optimal

  Accuracy: 91.25

- SVM (k_L)

  Accuracy: 90.05±0.35

  Support Vector: 39.78±2.63

- SVM (k_P)

  Accuracy: 90.10±0.46

  Support Vector: 53.65±1.74

- MKL (k_L-k_P)

  Accuracy: 90.95±0.61

  Support Vector: 38.23±2.34

  $(\eta_L\text{-}\eta_P) = (0.31\text{-}0.69)$

Multiple Kernel Learning

- P feature representations (not necessarily different)
  $\boldsymbol{x}_i = \{\boldsymbol{x}_i^m\}_{m=1}^P$ where $\boldsymbol{x}_i^m \in \mathbb{R}^{D_m}$

- Multiple Kernel Learning (MKL)
  $k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) = f_\eta(\{k_m(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m)\}_{m=1}^P | \boldsymbol{\Theta})$ where $f_\eta : \mathbb{R}^P \rightarrow \mathbb{R}$

Multiple Kernel Learning

- P feature representations (not necessarily different)
  $\boldsymbol{x}_i = \{\boldsymbol{x}_i^m\}_{m=1}^P$ where $\boldsymbol{x}_i^m \in \mathbb{R}^{D_m}$

- Multiple Kernel Learning (MKL)
  $k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) = f_\eta(\{k_m(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m)\}_{m=1}^P | \boldsymbol{\Theta})$ where $f_\eta : \mathbb{R}^P \rightarrow \mathbb{R}$

- Why MKL?
  - Different notions of similarity
    $\boldsymbol{x}_i^1 = \boldsymbol{x}_i^2 = \cdots = \boldsymbol{x}_i^P$
    $k_1 \neq k_2 \neq \cdots \neq k_P$
  - Different representations from different sources or modalities
    $\boldsymbol{x}_i^1 \neq \boldsymbol{x}_i^2 \neq \cdots \neq \boldsymbol{x}_i^P$
    $k_1 = k_2 = \cdots = k_P$

Multiple Kernel Learning

- Unweighted sum

$$k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{P} k_m(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m)$$

(Pavlidis et al., 2001)

Multiple Kernel Learning

- Unweighted sum

$$k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{P} k_m(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m)$$

(Pavlidis et al., 2001)

- Weighted sum

$$k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j | \boldsymbol{\Theta} = \boldsymbol{\eta}) = \sum_{m=1}^{P} \eta_m k_m(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m)$$

(Lanckriet et al., 2004; Bach et al., 2004)

Multiple Kernel Learning

- Unweighted sum

$$k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{P} k_m(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m)$$

(Pavlidis et al., 2001)

- Weighted sum

$$k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j | \boldsymbol{\Theta} = \boldsymbol{\eta}) = \sum_{m=1}^{P} \eta_m k_m(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m)$$

(Lanckriet et al., 2004; Bach et al., 2004)

- Locally weighted sum

$$k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j | \boldsymbol{\Theta} = \mathbf{V}) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}_i | \mathbf{V}) k_m(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m) \eta_m(\boldsymbol{x}_j | \mathbf{V})$$

(Gönen and Alpaydın, 2008)

Multiple Kernel Learning

- Weighted MKL formulation of Bach et al. (2004)

- $f(\boldsymbol{x}) = \sum_{m=1}^{P} \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}_i^m) \rangle + b$

Multiple Kernel Learning

- Weighted MKL formulation of Bach et al. (2004)
- $f(\boldsymbol{x}) = \sum_{m=1}^{P} \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}_i^m) \rangle + b$
- Primal optimization problem

$$\begin{aligned}
\text{min.} \quad & \frac{1}{2} \left(\sum_{m=1}^{P} d_m \|\boldsymbol{w}_m\|_2 \right)^2 + C \sum_{i=1}^{N} \xi_i \\
\text{w.r.t.} \quad & \boldsymbol{w}_m \in \mathbb{R}^{S_m}, \boldsymbol{\xi} \in \mathbb{R}_+^N, b \in \mathbb{R} \\
\text{s.t.} \quad & y_i \left(\sum_{m=1}^{P} \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}_i^m) \rangle + b \right) \geq 1 - \xi_i \qquad \forall i
\end{aligned}$$

- weighted l_1-norm on feature spaces
- l_2-norm within each feature space

Multiple Kernel Learning

- Dual optimization problem

$$
\begin{aligned}
\text{min.} \quad & \frac{1}{2}\gamma^2 - \sum_{i=1}^{N} \alpha_i \\
\text{w.r.t.} \quad & \gamma \in \mathbb{R}, \boldsymbol{\alpha} \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \gamma^2 d_m^2 \geq \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k_m(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m) \qquad \forall m \\
& \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& C \geq \alpha_i \geq 0 \qquad \forall i
\end{aligned}
$$

Multiple Kernel Learning

- Dual optimization problem

$$
\begin{aligned}
\text{min.} \quad & \frac{1}{2}\gamma^2 - \sum_{i=1}^{N} \alpha_i \\
\text{w.r.t.} \quad & \gamma \in \mathbb{R}, \boldsymbol{\alpha} \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \gamma^2 d_m^2 \geq \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k_m(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m) \qquad \forall m \qquad \eta_m \\
& \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& C \geq \alpha_i \geq 0 \qquad \forall i
\end{aligned}
$$

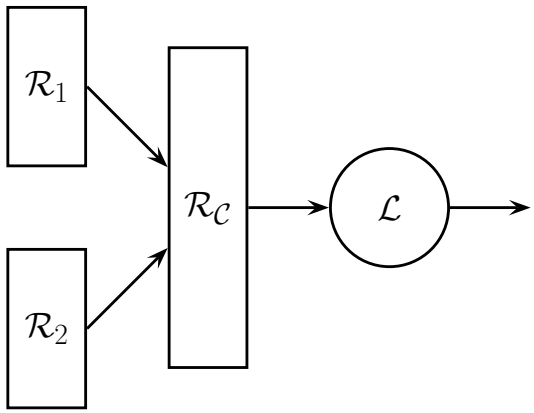
- $\boldsymbol{w}_m = \sum_{i=1}^{N} \alpha_i y_i \eta_m \Phi_m(\boldsymbol{x}_i^m)$
- $f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \sum_{m=1}^{P} \eta_m k_m(\boldsymbol{x}_i^m, \boldsymbol{x}^m) + b \qquad \text{where} \sum_{m=1}^{P} d_m^2 \eta_m = 1$

Multiple Kernel Learning as Intermediate Integration

- Three methods for combining multiple representations (Noble, 2004)

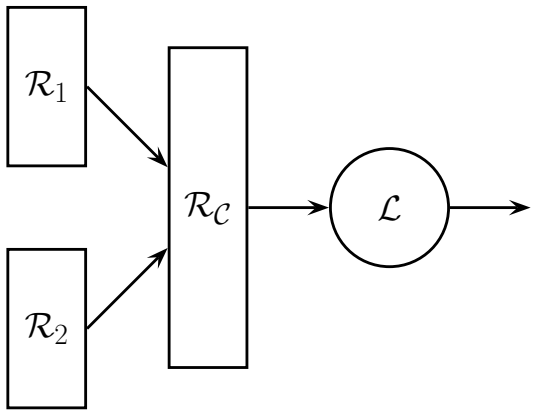
Multiple Kernel Learning as Intermediate Integration

- Three methods for combining multiple representations (Noble, 2004)
  - Early integration

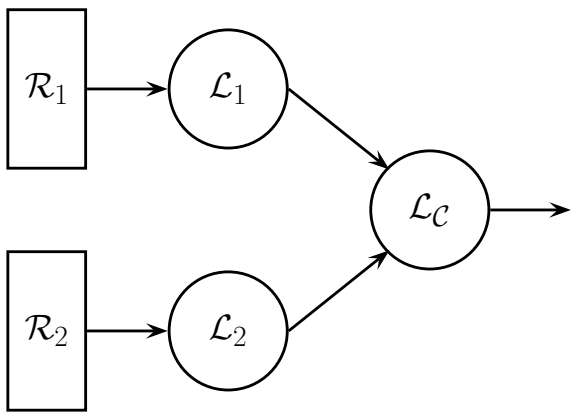


Multiple Kernel Learning as Intermediate Integration

- Three methods for combining multiple representations (Noble, 2004)
  - Early integration

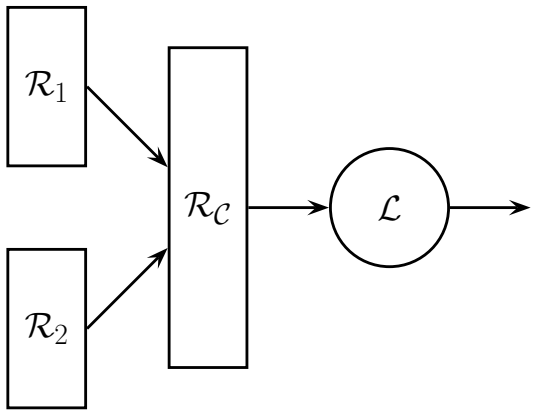


  - Late integration

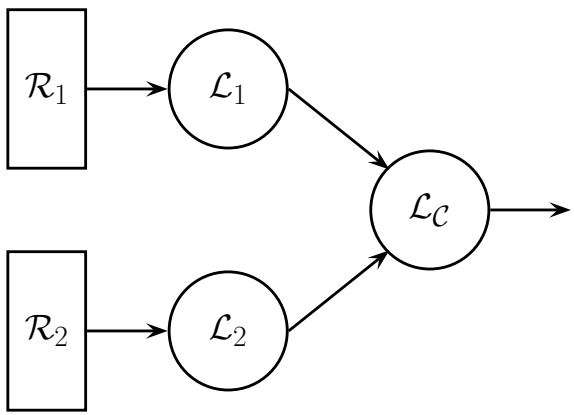


Multiple Kernel Learning as Intermediate Integration

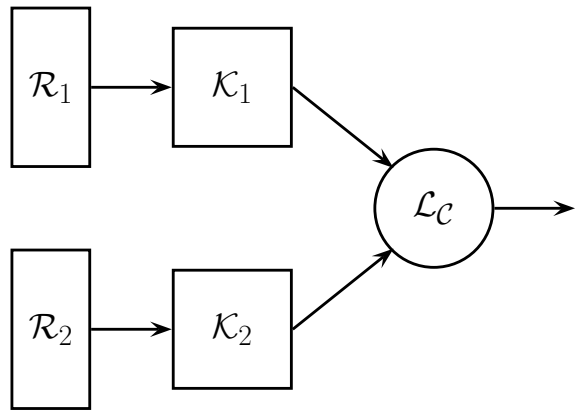
- Three methods for combining multiple representations (Noble, 2004)
  - Early integration



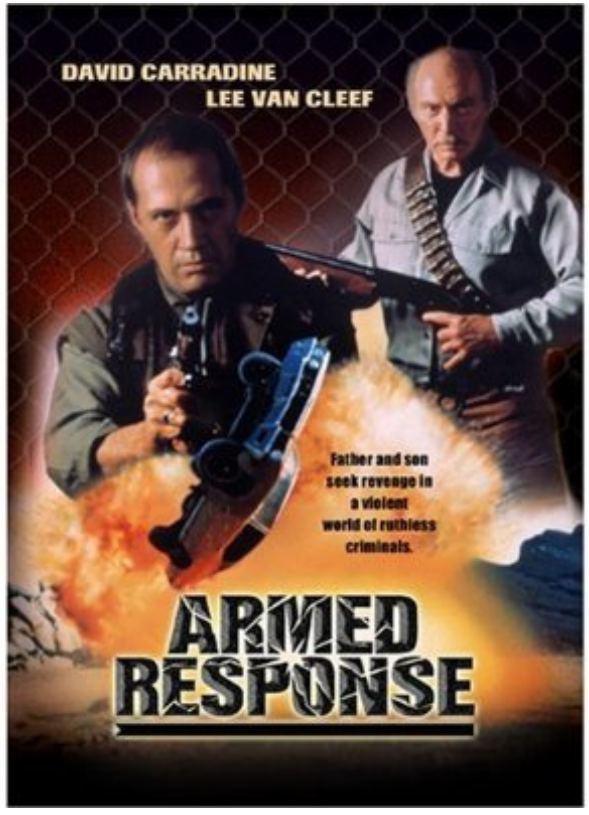
  - Late integration



  - Intermediate integration

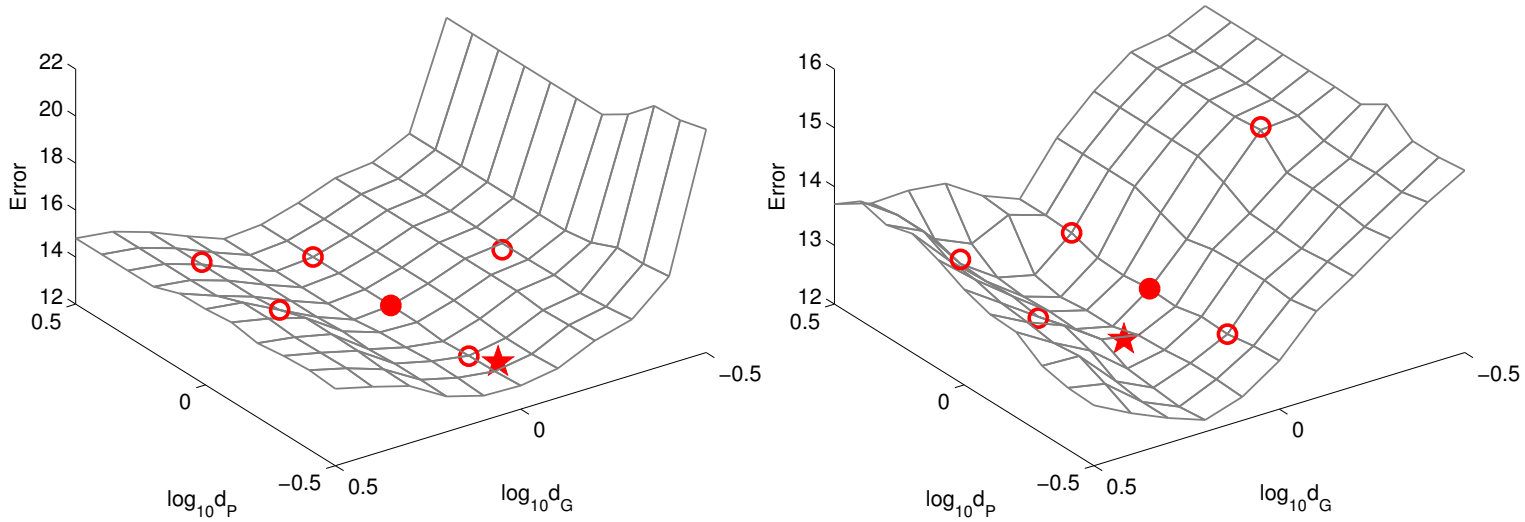


Regularizing Multiple Kernel Learning Using Response Surface Methodology



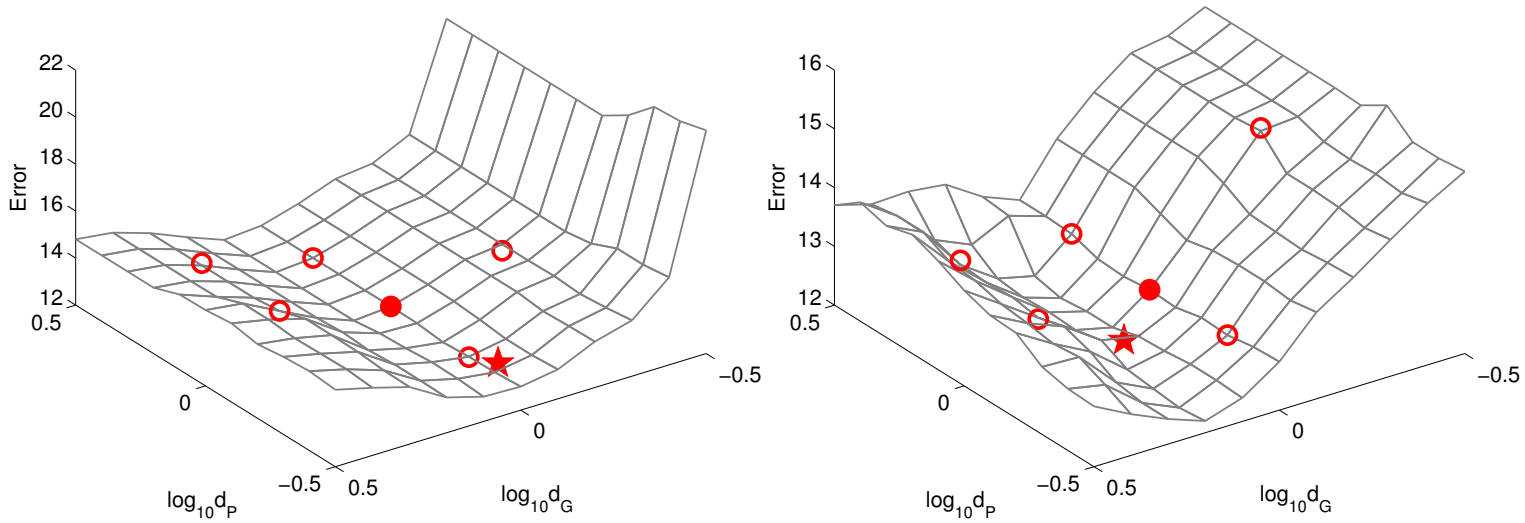
Regularized Multiple Kernel Learning

- $\{d_m\}_{m=1}^{P}$ parameters in the formulation of Bach et al. (2004)



Regularized Multiple Kernel Learning

- $\{d_m\}_{m=1}^{P}$ parameters in the formulation of Bach et al. (2004)

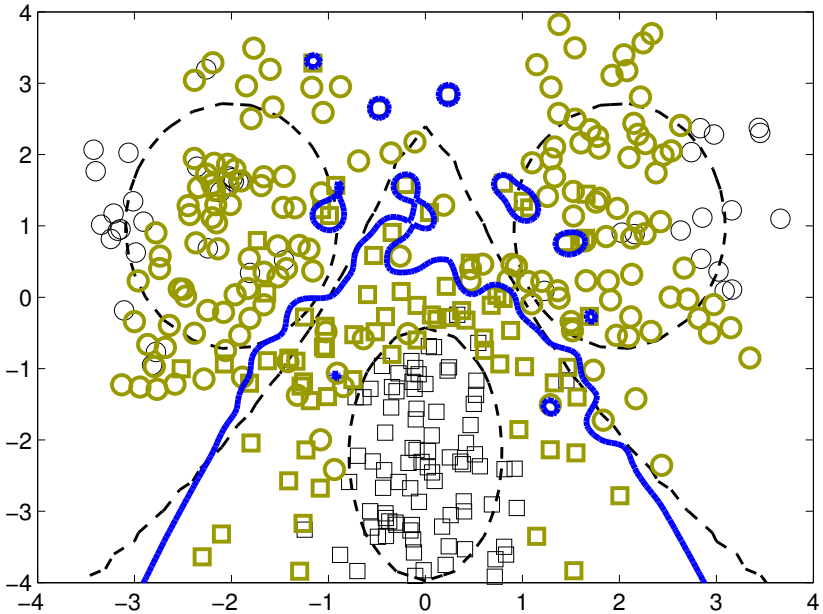


- Second-order model in the log scale

$$\text{error} = \beta_0 + \sum_{m=2}^{P} \beta_m \log_{10} d_m + \sum_{m=2}^{P} \sum_{h=m}^{P} \beta_{mh} \log_{10} d_m \log_{10} d_h$$

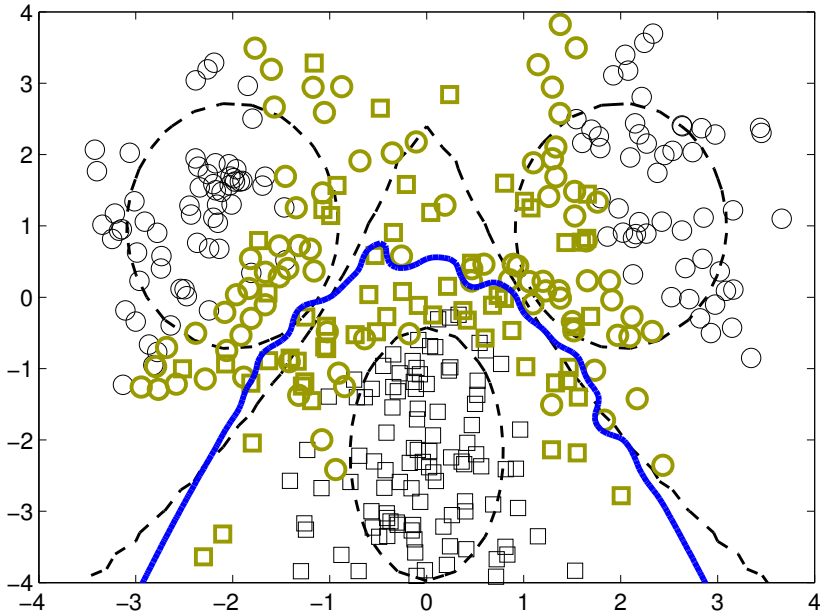
- Grid search versus RSM
  - $L^{(P-1)}$ points versus $\mathcal{O}(P^2)$ points
  - predefined factor levels versus arbitrary factor values

Regularized Multiple Kernel Learning



MKL

$(\eta_L\text{-}\eta_P\text{-}\eta_G) = (0.00\text{-}0.25\text{-}0.75)$

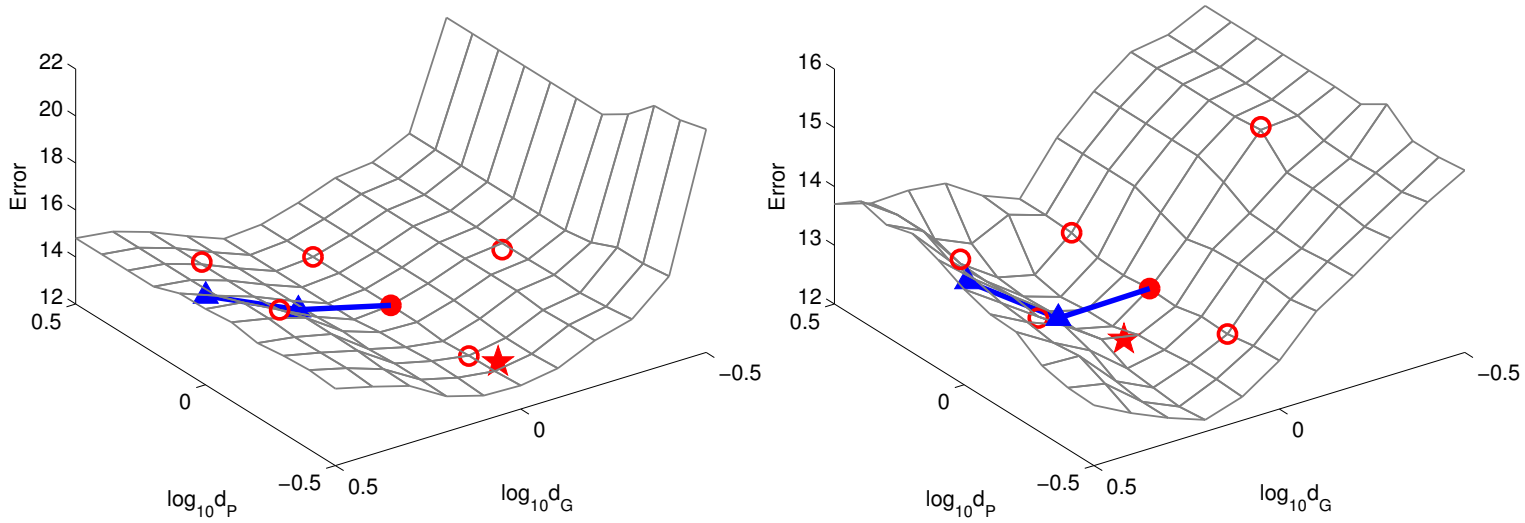


RMKL

$(\eta_L\text{-}\eta_P\text{-}\eta_G) = (0.00\text{-}0.48\text{-}0.31)$

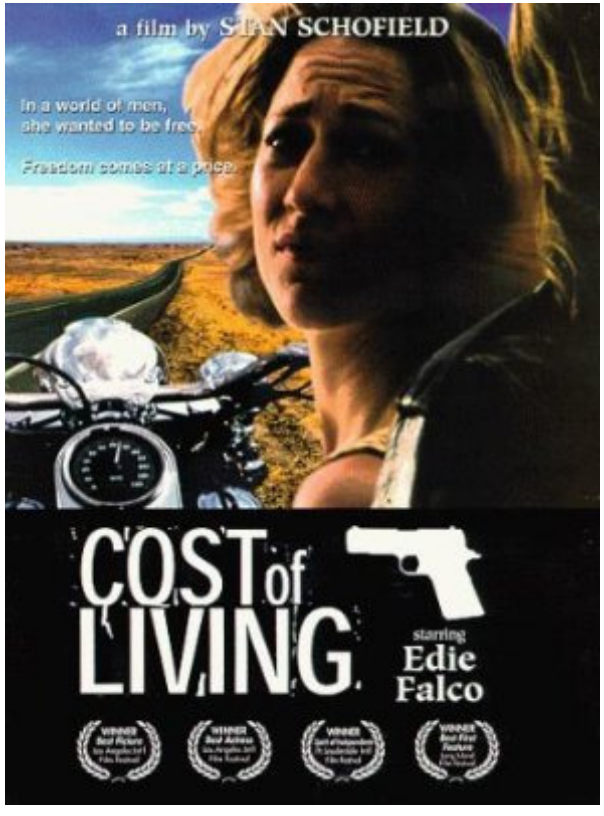
Regularized Multiple Kernel Learning

- Bach et al. (2005) propose to select $\{d_m\}_{m=1}^P$ by looking at eigenvalues of kernel matrices (EMKL)



- Perform RSM on $\{\eta_m\}_{m=1}^P$ directly (RWKL)

Cost-Conscious Multiple Kernel Learning



Cost-Conscious Multiple Kernel Learning

- $\{d_m\}_{m=1}^{P}$ parameters in the formulation of Bach et al. (2004)

- Can be interpreted as the costs of kernels (Gönen and Alpaydın, 2010a)

Cost-Conscious Multiple Kernel Learning

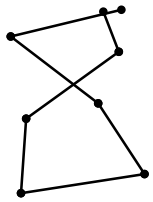
- $\{d_m\}_{m=1}^{P}$ parameters in the formulation of Bach et al. (2004)
- Can be interpreted as the costs of kernels (Gönen and Alpaydın, 2010a)
- The costs of evaluating kernels
  - preprocessing data
  - computation time

Cost-Conscious Multiple Kernel Learning

- $\{d_m\}_{m=1}^{P}$ parameters in the formulation of Bach et al. (2004)
- Can be interpreted as the costs of kernels (Gönen and Alpaydın, 2010a)
- The costs of evaluating kernels
  - preprocessing data
  - computation time
- The costs of extracting/sensing the corresponding representations/signals
  - bioinformatics: protein sequence, protein structure, etc.
  - biometrics: face, fingerprint, iris, signature, etc.
  - speech recognition: acoustic input, visual input, etc.

Cost-Conscious Multiple Kernel Learning

- Representation selection



DYN

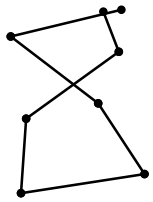
STA4

STA8

STA16

Cost-Conscious Multiple Kernel Learning

- Representation selection



DYN

STA4

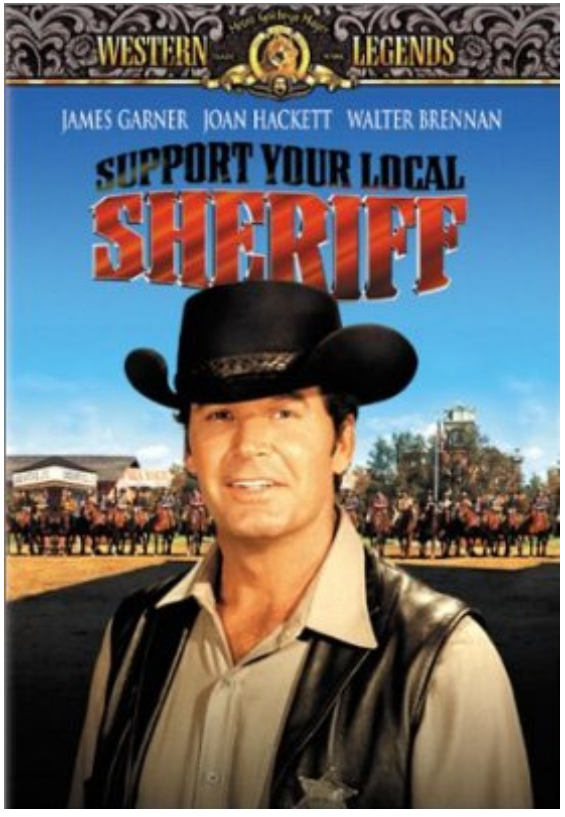
STA8

STA16

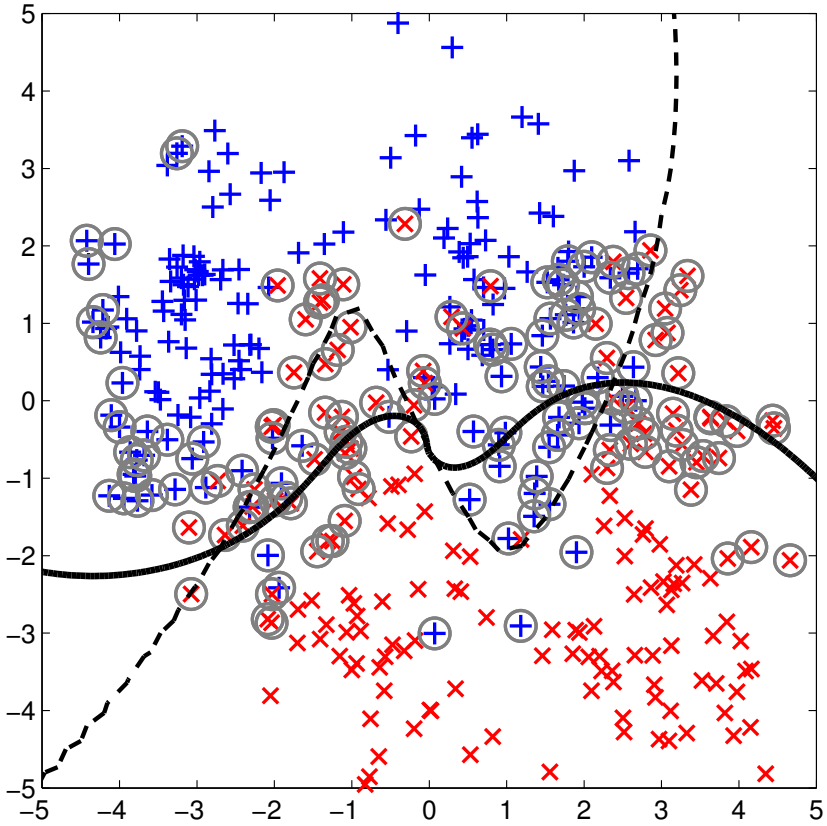
- Kernel/representation selection

Kernel	Explanation	Data Source
k_{SW}	Smith-Waterman kernel	Protein sequences
k_B	BLAST kernel	Protein sequences
k_{Pfam}	Pfam kernel	Protein sequences
k_{FFT}	FFT kernel	Hydropathy profiles
k_{LI}	Linear kernel	Protein interactions
k_D	Diffusion kernel	Protein interactions
k_E	Gaussian kernel	Gene expression profiles

Outline



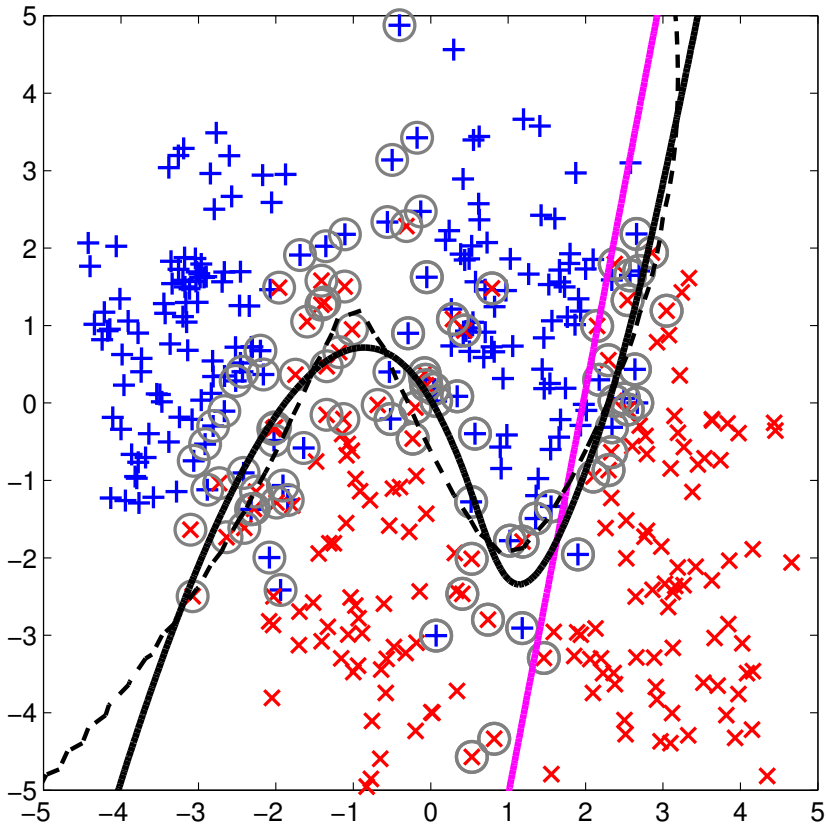
Localized Multiple Kernel Learning



MKL (k_L-k_P)

Accuracy: 90.95±0.61

Support Vector: 38.23±2.34



LMKL (k_L-k_P)

Accuracy: 91.83±0.24

Support Vector: 25.12±0.91

Localized Multiple Kernel Learning

- $f(\boldsymbol{x}) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}^m) \rangle + b$ where $\eta_m : \mathbb{R}^{D_{\mathcal{G}}} \rightarrow \mathbb{R}$
- Similar to (but also different from) "mixture of experts" (Jacobs et al., 1991)

Localized Multiple Kernel Learning

- $f(\boldsymbol{x}) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}^m) \rangle + b$ where $\eta_m : \mathbb{R}^{D_{\mathcal{G}}} \rightarrow \mathbb{R}$
- Similar to (but also different from) "mixture of experts" (Jacobs et al., 1991)
- Softmax gating

$$\eta_m(\boldsymbol{x}|\mathbf{V}) = \frac{\exp(\langle \boldsymbol{v}_m, \boldsymbol{x}^{\mathcal{G}} \rangle + v_{m0})}{\sum_{h=1}^{P} \exp(\langle \boldsymbol{v}_h, \boldsymbol{x}^{\mathcal{G}} \rangle + v_{h0})} \quad \forall m$$

Localized Multiple Kernel Learning

- $f(\boldsymbol{x}) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}^m) \rangle + b$ where $\eta_m : \mathbb{R}^{D_{\mathcal{G}}} \rightarrow \mathbb{R}$
- Similar to (but also different from) "mixture of experts" (Jacobs et al., 1991)
- Softmax gating

$$\eta_m(\boldsymbol{x}|\mathbf{V}) = \frac{\exp(\langle \boldsymbol{v}_m, \boldsymbol{x}^{\mathcal{G}} \rangle + v_{m0})}{\sum_{h=1}^{P} \exp(\langle \boldsymbol{v}_h, \boldsymbol{x}^{\mathcal{G}} \rangle + v_{h0})} \quad \forall m$$

- Sigmoid gating

$$\eta_m(\boldsymbol{x}|\mathbf{V}) = \frac{1}{1 + \exp(-\langle \boldsymbol{v}_m, \boldsymbol{x}^{\mathcal{G}} \rangle - v_{m0})} \quad \forall m$$

Localized Multiple Kernel Learning

- Primal optimization problem (Gönen and Alpaydın, 2008)

$$\begin{aligned}
\text{min.} \quad & \frac{1}{2}\sum_{m=1}^{P}\|\boldsymbol{w}_m\|_2^2 + C\sum_{i=1}^{N}\xi_i \\
\text{w.r.t.} \quad & \boldsymbol{w}_m \in \mathbb{R}^{S_m}, \boldsymbol{\xi} \in \mathbb{R}_+^N, \mathbf{V} \in \mathbb{R}^{P\times(D_{\mathcal{G}}+1)}, b \in \mathbb{R} \\
\text{s.t.} \quad & y_i\left(\sum_{m=1}^{P}\eta_m(\boldsymbol{x}_i|\mathbf{V})\langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}_i^m)\rangle + b\right) \geq 1-\xi_i \qquad \forall i
\end{aligned}$$

Localized Multiple Kernel Learning

- Primal optimization problem (Gönen and Alpaydın, 2008)

$$\begin{aligned}
\text{min.} \quad & \frac{1}{2}\sum_{m=1}^{P}\|\boldsymbol{w}_m\|_2^2 + C\sum_{i=1}^{N}\xi_i \\
\text{w.r.t.} \quad & \boldsymbol{w}_m \in \mathbb{R}^{S_m}, \boldsymbol{\xi} \in \mathbb{R}_+^N, \mathbf{V} \in \mathbb{R}^{P\times(D_{\mathcal{G}}+1)}, b \in \mathbb{R} \\
\text{s.t.} \quad & y_i\left(\sum_{m=1}^{P}\eta_m(\boldsymbol{x}_i|\mathbf{V})\langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}_i^m)\rangle + b\right) \geq 1-\xi_i \qquad \forall i
\end{aligned}$$

- Nonconvexity due to gating model
- For a given $\mathbf{V}$, problem becomes convex

Localized Multiple Kernel Learning

- Dual optimization problem for a given $\mathbf{V}$

$$\begin{aligned}
\text{max.} \quad & J(\mathbf{V}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_i y_i y_j k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) \\
\text{w.r.t.} \quad & \boldsymbol{\alpha} \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& C \geq \alpha_i \geq 0 \qquad \forall i
\end{aligned}$$

Localized Multiple Kernel Learning

- Dual optimization problem for a given $\mathbf{V}$

$$\begin{aligned}
\text{max.} \quad & J(\mathbf{V}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_i y_i y_j k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) \\
\text{w.r.t.} \quad & \boldsymbol{\alpha} \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& C \geq \alpha_i \geq 0 \qquad \forall i
\end{aligned}$$

- *locally combined kernel function*

$$k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}_i|\mathbf{V}) \underbrace{\langle \Phi_m(\boldsymbol{x}_i^m), \Phi_m(\boldsymbol{x}_j^m) \rangle}_{k_m(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m)} \eta_m(\boldsymbol{x}_j|\mathbf{V})$$

Localized Multiple Kernel Learning

1: Initialize $\mathbf{V}^{(0)}$ randomly
2: **repeat**
3: Calculate $\mathbf{K}_\eta^{(t)} = \left\{k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j)\right\}_{i,j=1}^{N}$ using $\mathbf{V}^{(t)}$
4: Solve kernel machine with $\mathbf{K}_\eta^{(t)}$
5: Determine step size, $\Delta^{(t)}$, using Armijo's rule
6: $\mathbf{V}^{(t+1)} \Leftarrow \mathbf{V}^{(t)} - \Delta^{(t)} \dfrac{\partial J(\mathbf{V})}{\partial \mathbf{V}}$
7: **until** convergence

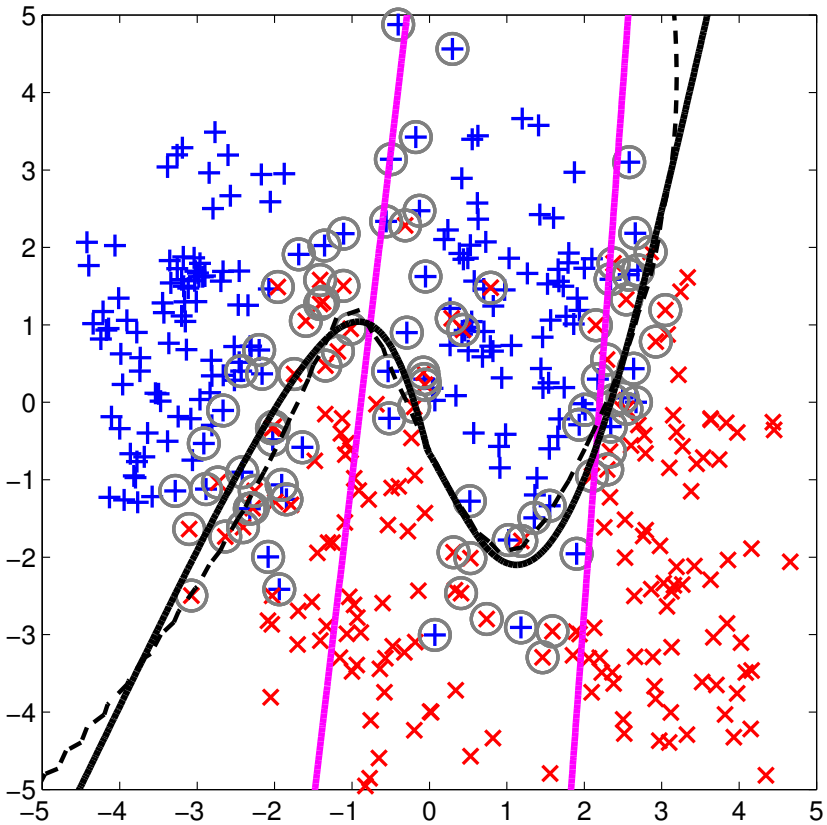
- $$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \underbrace{\sum_{m=1}^{P} \eta_m(\boldsymbol{x}_i|\mathbf{V}) k_m(\boldsymbol{x}_i^m, \boldsymbol{x}^m) \eta_m(\boldsymbol{x}|\mathbf{V})}_{k_\eta(\boldsymbol{x}_i, \boldsymbol{x})} + b$$

Localized Multiple Kernel Learning

1: Initialize $\mathbf{V}^{(0)}$ randomly
2: **repeat**
3: Calculate $\mathbf{K}_\eta^{(t)} = \left\{k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j)\right\}_{i,j=1}^{N}$ using $\mathbf{V}^{(t)}$
4: Solve kernel machine with $\mathbf{K}_\eta^{(t)}$
5: Determine step size, $\Delta^{(t)}$, using Armijo's rule
6: $\mathbf{V}^{(t+1)} \Leftarrow \mathbf{V}^{(t)} - \Delta^{(t)} \dfrac{\partial J(\mathbf{V})}{\partial \mathbf{V}}$
7: **until** convergence

- $$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \underbrace{\sum_{m=1}^{P} \eta_m(\boldsymbol{x}_i|\mathbf{V}) k_m(\boldsymbol{x}_i^m, \boldsymbol{x}^m) \eta_m(\boldsymbol{x}|\mathbf{V})}_{k_\eta(\boldsymbol{x}_i, \boldsymbol{x})} + b$$
- Can be generalized to other kernel-based methods

Localized Multiple Kernel Learning

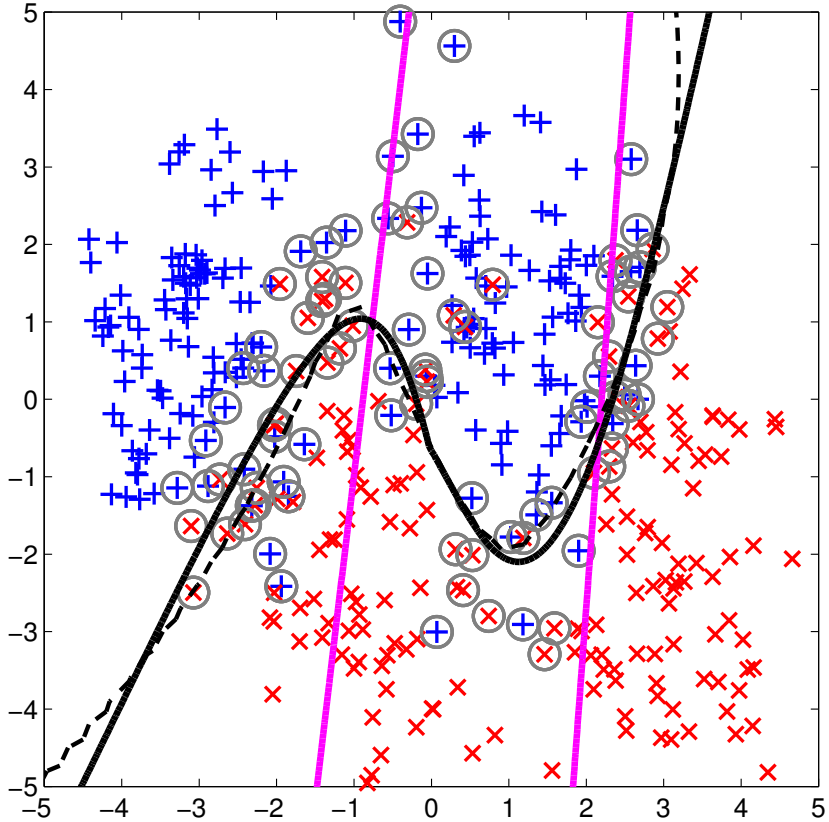


LMKL (k_L-k_L-k_L)

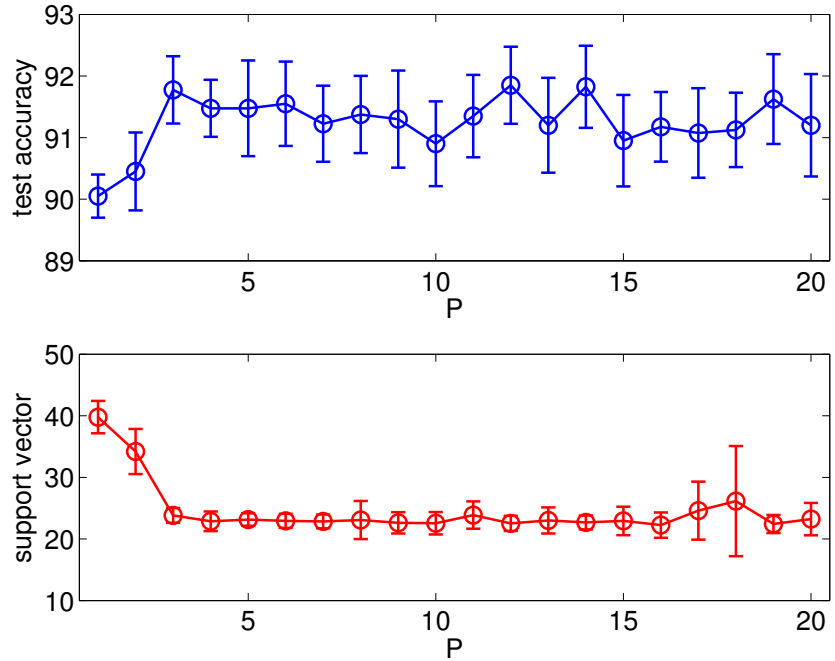
Accuracy: 91.78±0.55

Support Vector: 23.82±1.20

Localized Multiple Kernel Learning



LMKL (k_L-k_L-k_L)

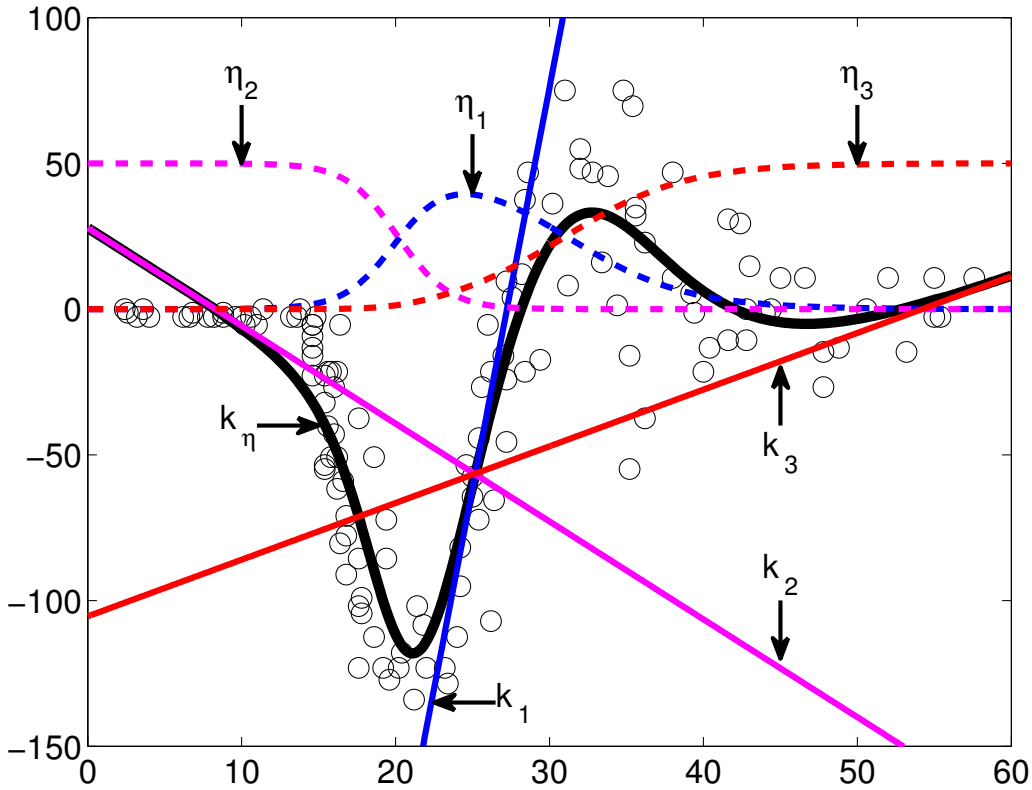


Accuracy: 91.78±0.55

Support Vector: 23.82±1.20

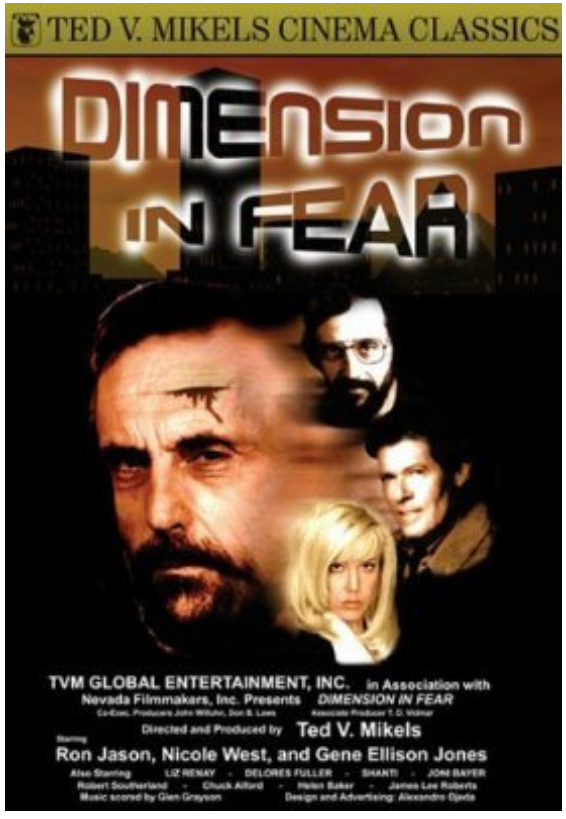
Localized Multiple Kernel Regression

- LMKL on regression problems (Gönen and Alpaydın, 2010b)
- MOTORCYCLE data set



LMKL (k_L-k_L-k_L)

Outline



Global Projection Kernels

- $\boldsymbol{z} = \mathbf{W}^\top \boldsymbol{x}$ where $\mathbf{W} \in \mathbb{R}^{D \times R}$
- $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\mathbf{W}^\top \boldsymbol{x}) \rangle + b$ where $\Phi : \mathbb{R}^R \rightarrow \mathbb{R}^S$

Global Projection Kernels

- $\boldsymbol{z} = \mathbf{W}^\top \boldsymbol{x}$ where $\mathbf{W} \in \mathbb{R}^{D \times R}$
- $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\mathbf{W}^\top \boldsymbol{x}) \rangle + b$ where $\Phi \colon \mathbb{R}^R \rightarrow \mathbb{R}^S$
- Primal optimization problem (Chapelle et al., 2002)

$$
\begin{aligned}
\text{min.} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^{N} \xi_i \\
\text{w.r.t.} \quad & \boldsymbol{w} \in \mathbb{R}^S, \boldsymbol{\xi} \in \mathbb{R}_+^N, \mathbf{W} \in \mathbb{R}^{D \times R}, b \in \mathbb{R} \\
\text{s.t.} \quad & y_i(\langle \boldsymbol{w}, \Phi(\mathbf{W}^\top \boldsymbol{x}_i) \rangle + b) \geq 1 - \xi_i \qquad \forall i
\end{aligned}
$$

Global Projection Kernels

- $\boldsymbol{z} = \mathbf{W}^\top \boldsymbol{x}$ where $\mathbf{W} \in \mathbb{R}^{D \times R}$
- $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\mathbf{W}^\top \boldsymbol{x}) \rangle + b$ where $\Phi : \mathbb{R}^R \rightarrow \mathbb{R}^S$
- Primal optimization problem (Chapelle et al., 2002)

$$
\begin{aligned}
\text{min.} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^{N} \xi_i \\
\text{w.r.t.} \quad & \boldsymbol{w} \in \mathbb{R}^S, \boldsymbol{\xi} \in \mathbb{R}_+^N, \mathbf{W} \in \mathbb{R}^{D \times R}, b \in \mathbb{R} \\
\text{s.t.} \quad & y_i(\langle \boldsymbol{w}, \Phi(\mathbf{W}^\top \boldsymbol{x}_i) \rangle + b) \geq 1 - \xi_i \qquad \forall i
\end{aligned}
$$

- For a given $\mathbf{W}$, problem becomes convex

Global Projection Kernels

- Dual optimization problem for a given $\mathbf{W}$

$$\begin{aligned}
\text{max.} \quad & J(\mathbf{W}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_i y_i y_j k(\boldsymbol{z}_i, \boldsymbol{z}_j) \\
\text{w.r.t.} \quad & \boldsymbol{\alpha} \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& C \geq \alpha_i \geq 0 \qquad \forall i
\end{aligned}$$

Global Projection Kernels

- Dual optimization problem for a given $\mathbf{W}$

$$\begin{aligned}
\text{max.} \quad & J(\mathbf{W}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_i y_i y_j k(\boldsymbol{z}_i, \boldsymbol{z}_j) \\
\text{w.r.t.} \quad & \boldsymbol{\alpha} \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\
& C \geq \alpha_i \geq 0 \qquad \forall i
\end{aligned}$$

- *global projection kernel function*

$$k(\boldsymbol{z}_i, \boldsymbol{z}_j) = \underbrace{\langle \Phi(\mathbf{W}^\top \boldsymbol{x}_i), \Phi(\mathbf{W}^\top \boldsymbol{x}_j) \rangle}_{k(\mathbf{W}^\top \boldsymbol{x}_i, \mathbf{W}^\top \boldsymbol{x}_j)}$$

Global Projection Kernels

1: Initialize $\mathbf{W}^{(0)}$ randomly
2: **repeat**
3: Calculate $\mathbf{K}^{(t)} = \{k(\boldsymbol{z}_i, \boldsymbol{z}_j)\}_{i,j=1}^{N}$ using $\mathbf{W}^{(t)}$
4: Solve kernel machine with $\mathbf{K}^{(t)}$
5: Determine step size, $\mu^{(t)}$, using Armijo's rule
6: $\mathbf{W}^{(t+1)} \Leftarrow \mathbf{W}^{(t)} - \mu^{(t)} \frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$
7: **until** convergence

- $f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i k(\mathbf{W}^\top \boldsymbol{x}_i, \mathbf{W}^\top \boldsymbol{x}) + b$

Global Projection Kernels

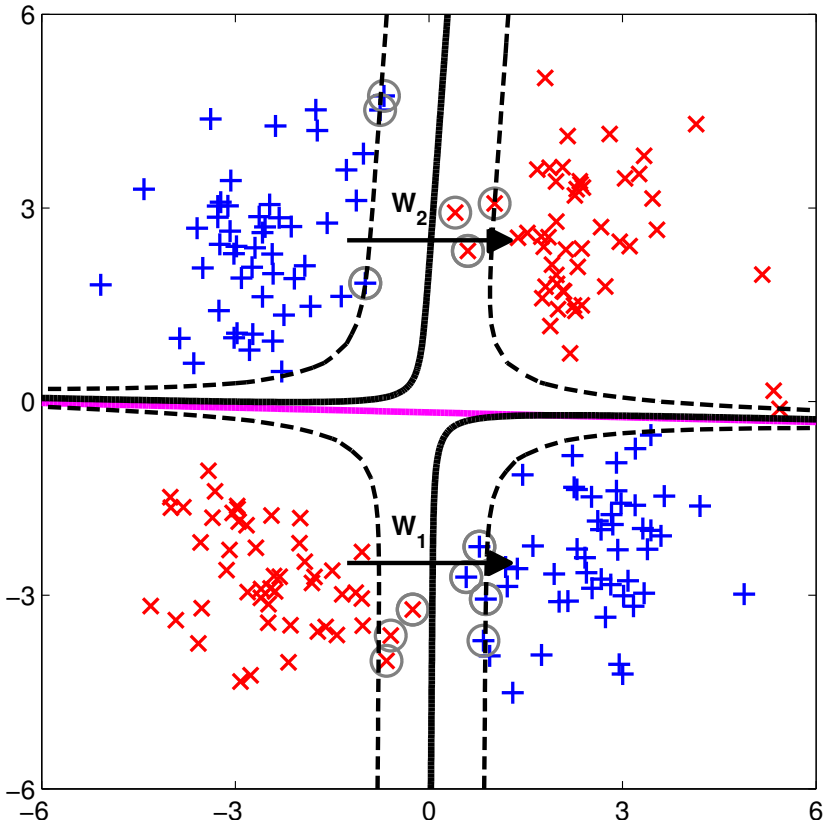
1: Initialize $\mathbf{W}^{(0)}$ randomly
2: **repeat**
3: Calculate $\mathbf{K}^{(t)} = \{k(\boldsymbol{z}_i, \boldsymbol{z}_j)\}_{i,j=1}^{N}$ using $\mathbf{W}^{(t)}$
4: Solve kernel machine with $\mathbf{K}^{(t)}$
5: Determine step size, $\mu^{(t)}$, using Armijo's rule
6: $\mathbf{W}^{(t+1)} \Leftarrow \mathbf{W}^{(t)} - \mu^{(t)} \dfrac{\partial J(\mathbf{W})}{\partial \mathbf{W}}$
7: **until** convergence

- $f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i k(\mathbf{W}^\top \boldsymbol{x}_i, \mathbf{W}^\top \boldsymbol{x}) + b$

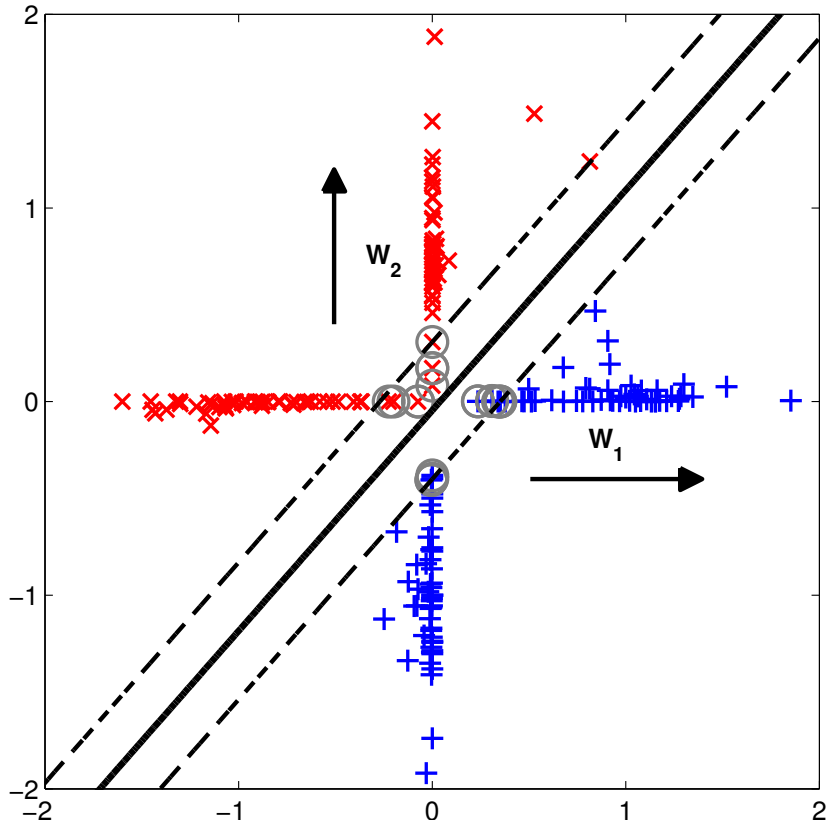
- Common kernel functions

$$k_L(\boldsymbol{z}_i, \boldsymbol{z}_j) = \langle \boldsymbol{z}_i, \boldsymbol{z}_j \rangle = \boldsymbol{x}_i^\top \mathbf{W}\mathbf{W}^\top \boldsymbol{x}_j$$
$$k_P(\boldsymbol{z}_i, \boldsymbol{z}_j) = (\langle \boldsymbol{z}_i, \boldsymbol{z}_j \rangle + 1)^q = (\boldsymbol{x}_i^\top \mathbf{W}\mathbf{W}^\top \boldsymbol{x}_j + 1)^q$$
$$k_G(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp(-\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2 / s^2)$$
$$= \exp(-(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \mathbf{W}\mathbf{W}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) / s^2)$$

Local Projection Kernels



Original Data



Projected Data

Local Projection Kernels

- $\boldsymbol{z}^m = \mathbf{W}_m^\top \boldsymbol{x}^m$ where $\mathbf{W}_m \in \mathbb{R}^{D_m \times R_m}$
- $f(\boldsymbol{x}) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\mathbf{W}_m^\top \boldsymbol{x}^m) \rangle + b$ where $\eta_m : \mathbb{R}^{D_\mathcal{G}} \rightarrow \mathbb{R}$

Local Projection Kernels

- $\boldsymbol{z}^m = \mathbf{W}_m^\top \boldsymbol{x}^m$ where $\mathbf{W}_m \in \mathbb{R}^{D_m \times R_m}$
- $f(\boldsymbol{x}) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\mathbf{W}_m^\top \boldsymbol{x}^m) \rangle + b$ where $\eta_m : \mathbb{R}^{D_\mathcal{G}} \rightarrow \mathbb{R}$
- Primal optimization problem (Gönen and Alpaydın, 2010c)

$$\begin{aligned}
\text{min.} \quad & \frac{1}{2} \sum_{m=1}^{P} \|\boldsymbol{w}_m\|_2^2 + C \sum_{i=1}^{N} \xi_i \\
\text{w.r.t.} \quad & \boldsymbol{w}_m \in \mathbb{R}^{S_m}, \boldsymbol{\xi} \in \mathbb{R}_+^N, \mathbf{V} \in \mathbb{R}^{P \times (D_\mathcal{G}+1)}, \mathbf{W}_m \in \mathbb{R}^{D_m \times R_m}, b \in \mathbb{R} \\
\text{s.t.} \quad & y_i \left(\sum_{m=1}^{P} \eta_m(\boldsymbol{x}_i|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\mathbf{W}_m^\top \boldsymbol{x}_i^m) \rangle + b \right) \geq 1 - \xi_i \quad \forall i
\end{aligned}$$

Local Projection Kernels

- $\boldsymbol{z}^m = \mathbf{W}_m^\top \boldsymbol{x}^m$ where $\mathbf{W}_m \in \mathbb{R}^{D_m \times R_m}$
- $f(\boldsymbol{x}) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\mathbf{W}_m^\top \boldsymbol{x}^m) \rangle + b$ where $\eta_m : \mathbb{R}^{D_\mathcal{G}} \to \mathbb{R}$
- Primal optimization problem (Gönen and Alpaydın, 2010c)

$$\begin{aligned}
\text{min.} \quad & \frac{1}{2} \sum_{m=1}^{P} \|\boldsymbol{w}_m\|_2^2 + C \sum_{i=1}^{N} \xi_i \\
\text{w.r.t.} \quad & \boldsymbol{w}_m \in \mathbb{R}^{S_m}, \boldsymbol{\xi} \in \mathbb{R}_+^N, \mathbf{V} \in \mathbb{R}^{P \times (D_\mathcal{G}+1)}, \mathbf{W}_m \in \mathbb{R}^{D_m \times R_m}, b \in \mathbb{R} \\
\text{s.t.} \quad & y_i \left(\sum_{m=1}^{P} \eta_m(\boldsymbol{x}_i|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\mathbf{W}_m^\top \boldsymbol{x}_i^m) \rangle + b \right) \geq 1 - \xi_i \quad \forall i
\end{aligned}$$

- For given $\mathbf{V}$ and $\{\mathbf{W}_m\}_{m=1}^P$, problem becomes convex

Local Projection Kernels

- Dual optimization problem for given $\mathbf{V}$ and $\{\mathbf{W}_m\}_{m=1}^P$

$$\begin{aligned}
\text{max.} \quad & J(\mathbf{V}, \{\mathbf{W}_m\}_{m=1}^P) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_i y_i y_j k\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) \\
\text{w.r.t.} \quad & \boldsymbol{\alpha} \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \\
& C \geq \alpha_i \geq 0 \qquad \forall i
\end{aligned}$$

Local Projection Kernels

- Dual optimization problem for given $\mathbf{V}$ and $\{\mathbf{W}_m\}_{m=1}^P$

$$\begin{aligned}
\text{max.} \quad & J(\mathbf{V}, \{\mathbf{W}_m\}_{m=1}^P) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_i y_i y_j k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) \\
\text{w.r.t.} \quad & \boldsymbol{\alpha} \in \mathbb{R}_+^N \\
\text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \\
& C \geq \alpha_i \geq 0 \qquad \forall i
\end{aligned}$$

- *local projection kernel function*

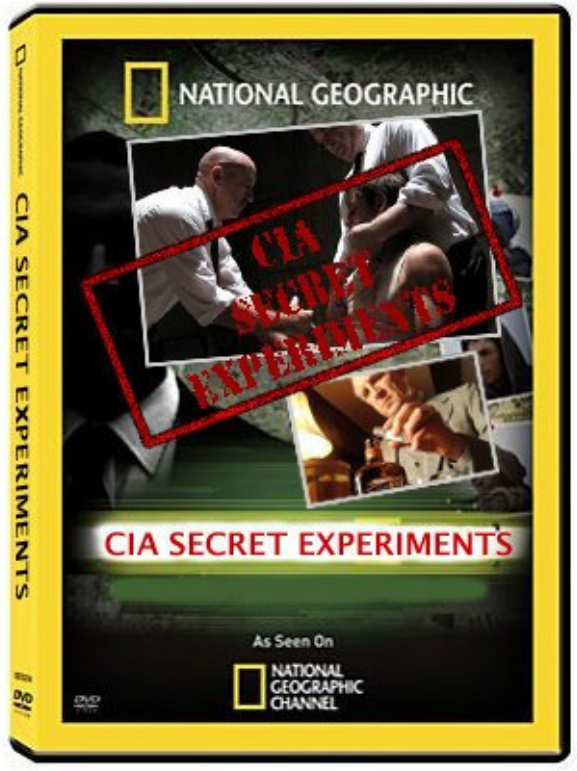
$$k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^P \eta_m(\boldsymbol{x}_i|\mathbf{V}) \underbrace{\langle \Phi_m(\mathbf{W}_m^\top \boldsymbol{x}_i^m), \Phi_m(\mathbf{W}_m^\top \boldsymbol{x}_j^m) \rangle}_{k_m(\mathbf{W}_m^\top \boldsymbol{x}_i^m, \mathbf{W}_m^\top \boldsymbol{x}_j^m)} \eta_m(\boldsymbol{x}_j|\mathbf{V})$$

Local Projection Kernels

1: Initialize $\mathbf{V}^{(0)}$ and $\{\mathbf{W}_m^{(0)}\}_{m=1}^P$ randomly
2: **repeat**
3: Calculate $\mathbf{K}_\eta^{(t)} = \left\{k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j)\right\}_{i,j=1}^N$ using $\mathbf{V}^{(t)}$ and $\{\mathbf{W}_m^{(t)}\}_{m=1}^P$
4: Solve kernel machine with $\mathbf{K}_\eta^{(t)}$
5: Determine step size, $\mu^{(t)}$, using Armijo's rule
6: $\mathbf{W}_m^{(t+1)} \Leftarrow \mathbf{W}_m^{(t)} - \mu^{(t)} \dfrac{\partial J(\mathbf{V}, \{\mathbf{W}_m\}_{m=1}^P)}{\partial \mathbf{W}_m} \quad \forall m$
7: Calculate $\mathbf{K}_\eta^{(t)} = \left\{k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j)\right\}_{i,j=1}^N$ using $\mathbf{V}^{(t)}$ and $\{\mathbf{W}_m^{(t+1)}\}_{m=1}^P$
8: Solve kernel machine with $\mathbf{K}_\eta^{(t)}$
9: Determine step size, $\Delta^{(t)}$, using Armijo's rule
10: $\mathbf{V}^{(t+1)} \Leftarrow \mathbf{V}^{(t)} - \Delta^{(t)} \dfrac{\partial J(\mathbf{V}, \{\mathbf{W}_m\}_{m=1}^P)}{\partial \mathbf{V}}$
11: **until** convergence

- $f(\boldsymbol{x}) = \sum_{i=1}^N \alpha_i y_i \underbrace{\sum_{m=1}^P \eta_m(\boldsymbol{x}_i|\mathbf{V}) k_m(\mathbf{W}_m^\top \boldsymbol{x}_i^m, \mathbf{W}_m^\top \boldsymbol{x}^m) \eta_m(\boldsymbol{x}|\mathbf{V})}_{k_\eta(\boldsymbol{x}_i, \boldsymbol{x})} + b$

Outline



Intermediate Integration Experiments

- Protein stability change due to amino acid substitutions (Özen et al., 2009)

- We extract a data set from the ProTherm database (Gromiha et al., 2000)

Repr.	Original Feat.	New Repr.	New Feat.
SO	± 3 Neighbors (± 3NE) Mutation (MUT) T/PH	SO*	PAM250 (PAM)
TO	Mutation (MUT) C_α Contacts (CA) SA/T/PH	TO*	PAM250 (PAM) C_α B-FACTORS (BFA) C_β B-FACTORS (BFB) C_α and C_β Contacts (CB)
ST	± 3 neighbors (± 3 NE) Mutation (MUT) C_α Contacts (CA) SA/T/PH	ST*	PAM250 (PAM) C_α B-FACTORS (BFA) C_β B-FACTORS (BFB) C_α and C_β Contacts (CB)

Intermediate Integration Experiments

- Classification results

SO	SO*	TO	TO*	ST	ST*
0.800	0.799	0.805	0.802	0.806	0.804

- Knowledge extraction results

SO	(0.19)1NE + (0.20)2NE + (0.23)3NE + (0.27)MUT + (0.09)T + (0.03)PH
SO*	(0.19)1NE + (0.20)2NE + (0.22)3NE + (0.27)MUT + (0.09)T + (0.03)PH + (0.00)PAM
TO	(0.19)MUT + (0.56)CA + (0.17)SA + (0.05)T + (0.02)PH
TO*	(0.21)MUT + (0.23)CA + (0.12)SA + (0.06)T + (0.02)PH + (0.00)PAM + (0.23)CB + (0.07)BFA + (0.06)BFB
ST	(0.04)1NE + (0.03)2NE + (0.04)3NE + (0.21)MUT + (0.45)CA + (0.15)SA + (0.06)T + (0.02)PH
ST*	(0.02)1NE + (0.02)2NE + (0.03)3NE + (0.21)MUT + (0.21)CA + (0.11)SA + (0.06)T + (0.02)PH + (0.00)PAM + (0.19)CB + (0.06)BFA + (0.06)BFB

Combining General Purpose Kernels

- Acceptors and Donors $\Rightarrow$ human splice site detection
- Arabidopsis and Vertebrates $\Rightarrow$ translation initiation site detection
- Polyadenylation $\Rightarrow$ polyadenylation signal prediction

Combining General Purpose Kernels

- Acceptors and Donors $\Rightarrow$ human splice site detection
- Arabidopsis and Vertebrates $\Rightarrow$ translation initiation site detection
- Polyadenylation $\Rightarrow$ polyadenylation signal prediction
- MKL versus RMKL with $(k_L\text{-}k_P\text{-}k_G)$

	MKL		RMKL		
	Test Acc.	SV	Test Acc.	SV	
	$\eta_L\text{-}\eta_P\text{-}\eta_G$		$\eta_L\text{-}\eta_P\text{-}\eta_G$		$d_L\text{-}d_P\text{-}d_G$
Acceptors	90.45±0.42	50.12±1.11	**91.19±0.38**	**44.90±0.82**	
	0.30-0.70-0.00		0.00-0.00-4.59		1.00-1.55-0.47
Donors	94.77±0.28	27.78±0.47	94.78±0.21	**26.25±0.45**	
	0.11-0.03-0.86		0.17-0.00-0.73		1.00-1.52-1.06
Arabidopsis	85.29±0.84	62.32±1.10	**85.92±0.85**	62.39±0.98	
	0.25-0.75-0.00		0.00-0.00-5.16		1.00-1.00-0.44
Vertebrates	86.22±0.22	53.84±0.72	86.15±0.30	**40.86±0.44**	
	0.20-0.80-0.00		0.70-0.15-0.00		1.00-1.42-1.16
Polyadenylation	68.25±1.24	72.14±1.02	68.70±1.34	72.81±1.06	
	0.01-0.16-0.84		0.00-0.00-1.09		1.00-1.54-0.96

Combining Domain Specific Kernels

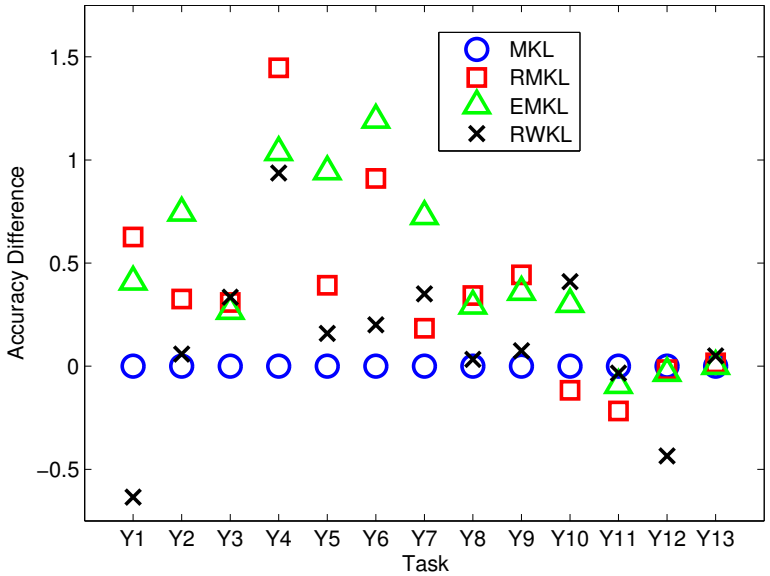
- Protein function prediction

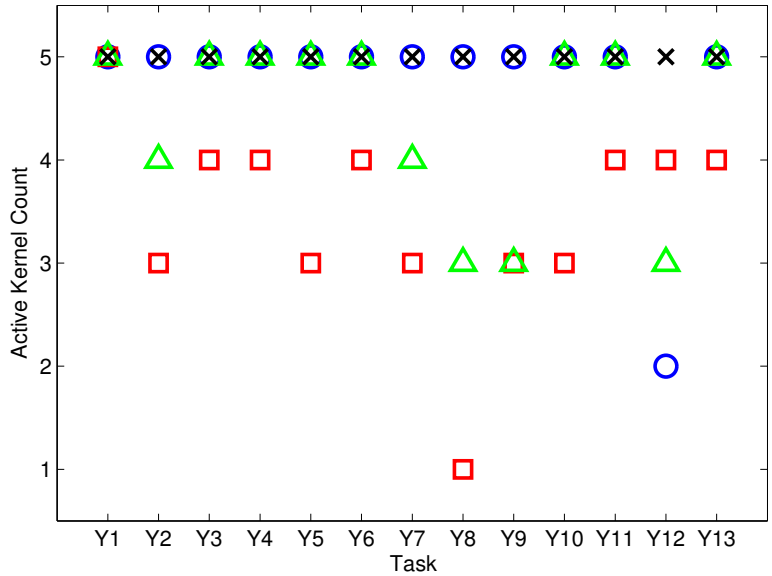
- Different kernel functions

Kernel	Explanation	Data Source
k_{Pfam}	Pfam kernel	Protein sequences
k_{PfamE}	Enriched Pfam kernel	Protein sequences
k_{TAP}	Diffusion kernel	Protein interactions
k_{Phys}	Diffusion kernel	Protein interactions
k_{Gen}	Diffusion kernel	Protein interactions
k_{Exp}	Correlation kernel	Gene expression profiles
k_{ExpG}	Gaussian kernel	Gene expression profiles
k_{SW}	Smith-Waterman kernel	Protein sequences

- We compare MKL, RMKL, EMKL, and RWKL

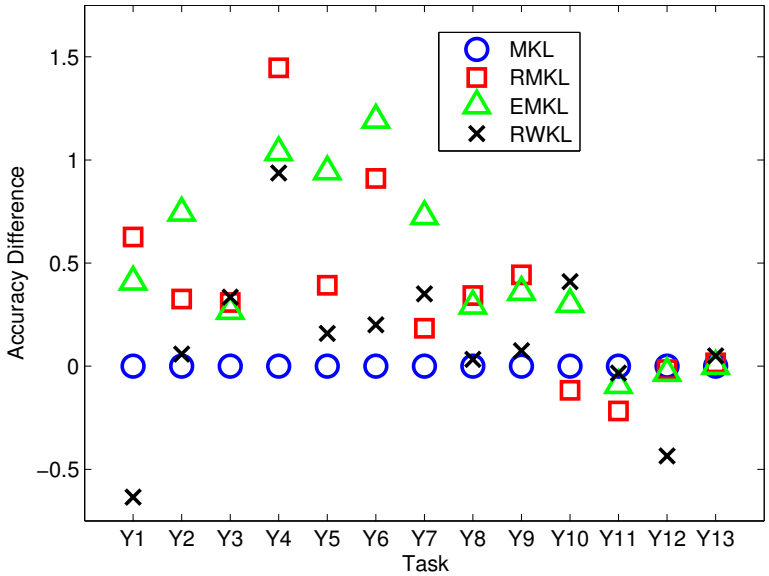
Combining Domain Specific Kernels

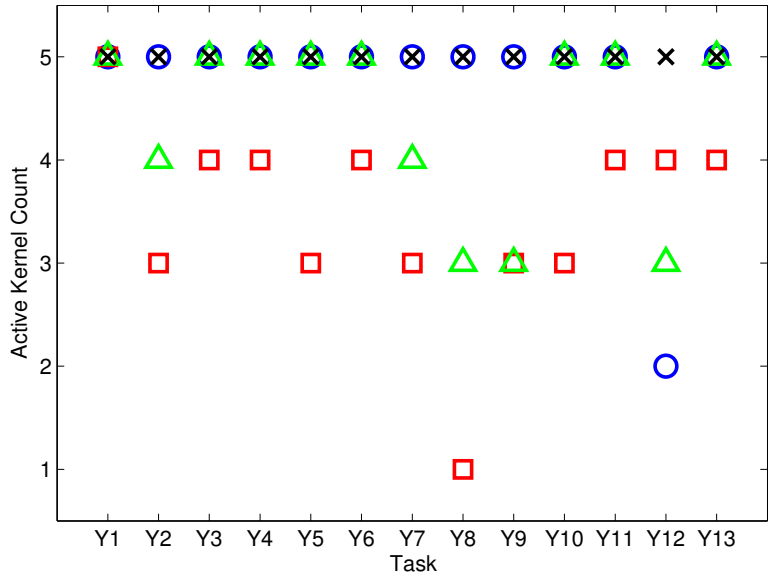




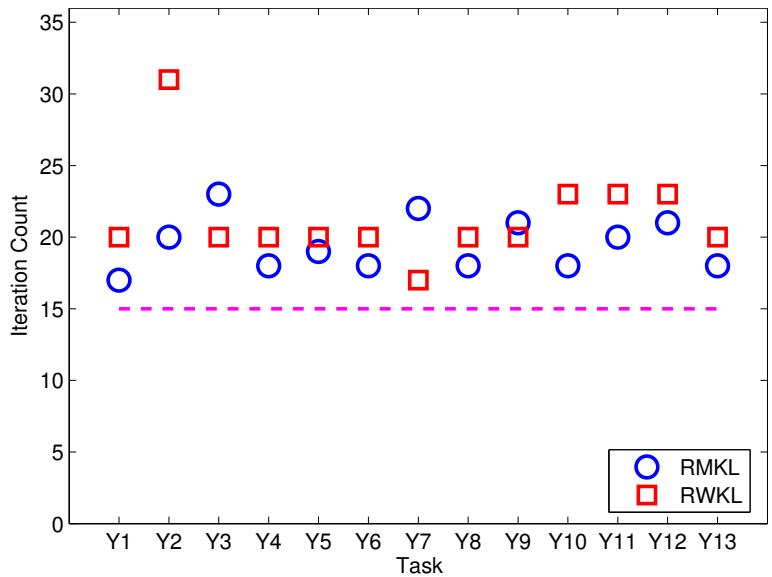
- RMKL produces smaller kernel ensembles without sacrificing accuracy

Combining Domain Specific Kernels





- RMKL produces smaller kernel ensembles without sacrificing accuracy
- Grid search requires 81 $\left(3^4\right)$ iterations

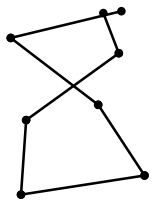


Representation Selection

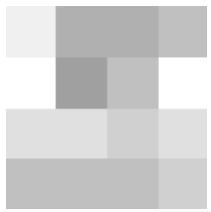
- $$\text{cost} = \underbrace{\sum_{i=1}^{N} \mathbf{1}(\alpha_i > 0)}_{\text{the number of support vectors}} \underbrace{\sum_{m=1}^{P} \mathbf{1}(\eta_m > 0) \frac{d_m}{\sum_{h=1}^{P} d_h}}_{\text{the total normalized cost for active kernels}}$$

Representation Selection

- $$\text{cost} = \underbrace{\sum_{i=1}^{N} \mathbf{1}(\alpha_i > 0)}_{\text{the number of support vectors}} \underbrace{\sum_{m=1}^{P} \mathbf{1}(\eta_m > 0) \frac{d_m}{\sum_{h=1}^{P} d_h}}_{\text{the total normalized cost for active kernels}}$$

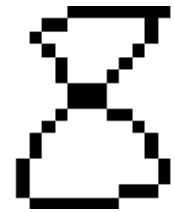


$d_{\text{DYN}} = 1.00$



$d_{\text{STA4}} = 1.00$

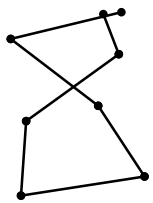
$d_{\text{STA8}} = 2.00$



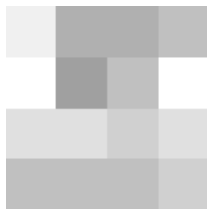
$d_{\text{STA16}} = 4.00$

Representation Selection

- $$\text{cost} = \underbrace{\sum_{i=1}^{N} \mathbf{1}(\alpha_i > 0)}_{\text{the number of support vectors}} \underbrace{\sum_{m=1}^{P} \mathbf{1}(\eta_m > 0) \frac{d_m}{\sum_{h=1}^{P} d_h}}_{\text{the total normalized cost for active kernels}}$$

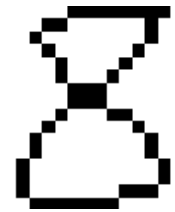


$d_{\text{DYN}} = 1.00$



$d_{\text{STA4}} = 1.00$

$d_{\text{STA8}} = 2.00$



$d_{\text{STA16}} = 4.00$

Data Set	Test Acc.	SV	Total Cost	η_{DYN}-η_{STA4}-η_{STA8}-η_{STA16}
PENDIGITS (0VS8)	99.50±0.11 99.30±0.32	10.70±0.11 **5.80±0.42**	10.70±0.33 **2.90±0.21**	0.24-0.16-0.33-0.26 0.48-0.38-0.14-0.00
PENDIGITS (1VS7)	99.90±0.11 99.70±0.32	9.70±0.53 **3.00±0.42**	9.70±1.33 **1.50±0.21**	0.32-0.31-0.31-0.06 0.45-0.51-0.04-0.00
PENDIGITS (5VS9)	99.30±0.11 99.50±0.11	12.90±0.95 **10.30±0.11**	12.90±0.04 **2.58±0.03**	0.35-0.12-0.10-0.44 0.64-0.36-0.00-0.00

Kernel/Representation Selection

- Protein location prediction with $(k_{SW}\text{-}k_B\text{-}k_{Pfam}\text{-}k_{FFT}\text{-}k_{LI}\text{-}k_D\text{-}k_E)$

Kernel	Explanation	Data Source
k_{SW}	Smith-Waterman kernel	Protein sequences
k_B	BLAST kernel	Protein sequences
k_{Pfam}	Pfam kernel	Protein sequences
k_{FFT}	FFT kernel	Hydropathy profiles
k_{LI}	Linear kernel	Protein interactions
k_D	Diffusion kernel	Protein interactions
k_E	Gaussian kernel	Gene expression profiles

- $(d_{SW}\text{-}d_B\text{-}d_{Pfam}\text{-}d_{FFT}\text{-}d_{LI}\text{-}d_D\text{-}d_E) = (1.41\text{-}1.41\text{-}1.41\text{-}1.41\text{-}1.00\text{-}2.00\text{-}1.00)$

Kernel/Representation Selection

- Protein location prediction with $(k_{SW}\text{-}k_B\text{-}k_{Pfam}\text{-}k_{FFT}\text{-}k_{LI}\text{-}k_D\text{-}k_E)$

Kernel	Explanation	Data Source
k_{SW}	Smith-Waterman kernel	Protein sequences
k_B	BLAST kernel	Protein sequences
k_{Pfam}	Pfam kernel	Protein sequences
k_{FFT}	FFT kernel	Hydropathy profiles
k_{LI}	Linear kernel	Protein interactions
k_D	Diffusion kernel	Protein interactions
k_E	Gaussian kernel	Gene expression profiles

- $(d_{SW}\text{-}d_B\text{-}d_{Pfam}\text{-}d_{FFT}\text{-}d_{LI}\text{-}d_D\text{-}d_E) = (1.41\text{-}1.41\text{-}1.41\text{-}1.41\text{-}1.00\text{-}2.00\text{-}1.00)$

Task	Test Acc.	SV	Total Cost	$\eta_{SW}\text{-}\eta_B\text{-}\eta_{Pfam}\text{-}\eta_{FFT}\text{-}\eta_{LI}\text{-}\eta_D\text{-}\eta_E$
MEMBRANE	86.30±0.61	84.42±1.03	75.34±5.09	0.28-0.31-0.11-0.05-0.00-0.15-0.10
	85.40±0.55	**71.33±1.67**	**37.81±5.09**	0.00-0.26-0.06-0.00-0.07-0.00-0.60
RIBOSOMAL	98.99±0.26	18.56±1.56	6.86±1.96	0.01-0.00-0.00-0.16-0.03-0.00-0.80
	99.07±0.18	18.61±1.46	6.08±1.96	0.00-0.00-0.00-0.01-0.03-0.00-0.96

Combining Multiple Feature Representations

- Multiple Features (MULTIFEAT) digit recognition data set
- Separate small ('0' - '4') digits from large ('5' - '9') digits

Name	Dim.	Data Source
FAC	216	Profile correlations
FOU	76	Fourier coefficients of the shapes
KAR	64	Karhunen-Loève coefficients
MOR	6	Morphological features
PIX	240	Pixel averages in 2×3 windows
ZER	47	Zernike moments

Combining Multiple Feature Representations

- Multiple Features (MULTIFEAT) digit recognition data set
- Separate small ('0' - '4') digits from large ('5' - '9') digits

Name	Dim.	Data Source
FAC	216	Profile correlations
FOU	76	Fourier coefficients of the shapes
KAR	64	Karhunen-Loève coefficients
MOR	6	Morphological features
PIX	240	Pixel averages in 2×3 windows
ZER	47	Zernike moments

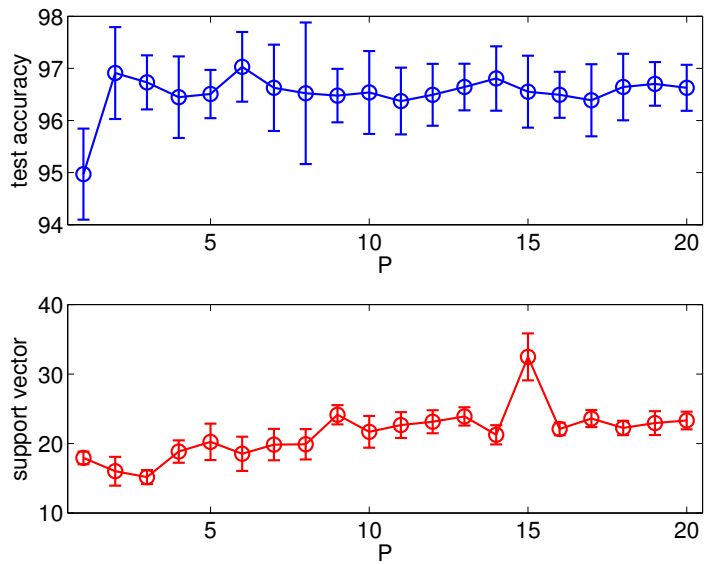
Method	Test Accuracy	Support Vector
SVM (FAC)	94.97±0.87	17.93±0.91
SVM (FOU)	90.54±1.11	28.90±1.69
SVM (KAR)	88.13±0.73	33.62±1.31
SVM (MOR)	69.61±0.14	61.90±0.49
SVM (PIX)	89.42±0.65	46.35±1.64
SVM (ZER)	89.12±0.63	26.27±1.67
SVM (ALL)	97.69±0.44	23.36±1.15
MKL	97.40±0.37	32.59±0.82
LMKL (softmax)	97.69±0.44	15.06±1.03
LMKL (sigmoid)	98.58±0.41	15.27±0.92

Combining Multiple Feature Representations

- Multiple Features (MULTIFEAT) digit recognition data set
- Separate small ('0' - '4') digits from large ('5' - '9') digits

Name	Dim.	Data Source
FAC	216	Profile correlations
FOU	76	Fourier coefficients of the shapes
KAR	64	Karhunen-Loève coefficients
MOR	6	Morphological features
PIX	240	Pixel averages in 2×3 windows
ZER	47	Zernike moments

Method	Test Accuracy	Support Vector
SVM (FAC)	94.97±0.87	17.93±0.91
SVM (FOU)	90.54±1.11	28.90±1.69
SVM (KAR)	88.13±0.73	33.62±1.31
SVM (MOR)	69.61±0.14	61.90±0.49
SVM (PIX)	89.42±0.65	46.35±1.64
SVM (ZER)	89.12±0.63	26.27±1.67
SVM (ALL)	97.69±0.44	23.36±1.15
MKL	97.40±0.37	32.59±0.82
LMKL (softmax)	97.69±0.44	15.06±1.03
LMKL (sigmoid)	98.58±0.41	15.27±0.92



Combining Multiple Feature Representations

- Internet Advertisements (ADVERT) data set
- Predict whether an image is an advertisement or not

Name	Dim.	Data Source
URL	457	Phrases in the URL
ORIGURL	495	Phrases in the URL of the image
ANCURL	472	Phrases in the anchor text
ALT	111	Phrases in the alternative text
CAPTION	19	Phrases in the caption terms

Combining Multiple Feature Representations

- Internet Advertisements (ADVERT) data set
- Predict whether an image is an advertisement or not

Name	Dim.	Data Source
URL	457	Phrases in the URL
ORIGURL	495	Phrases in the URL of the image
ANCURL	472	Phrases in the anchor text
ALT	111	Phrases in the alternative text
CAPTION	19	Phrases in the caption terms

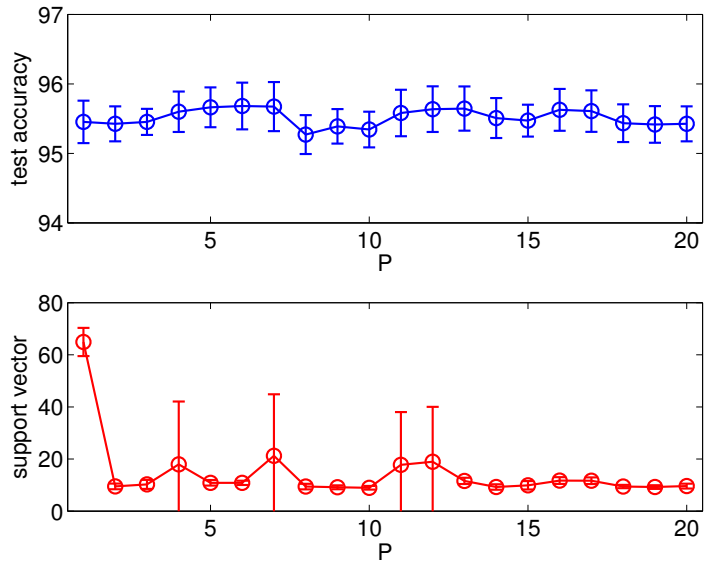
Method	Test Accuracy	Support Vector
SVM (URL)	94.67±0.24	83.32± 1.89
SVM (ORIGURL)	92.04±0.26	96.16± 0.51
SVM (ANCURL)	95.45±0.31	64.90± 5.41
SVM (ALT)	89.64±0.38	87.73± 1.17
SVM (CAPTION)	86.60±0.09	96.65± 0.42
SVM (ALL)	96.43±0.24	41.99± 1.76
MKL	96.32±0.50	35.82± 4.35
LMKL (softmax)	95.78±0.46	41.72±11.59
LMKL (sigmoid)	96.72±0.46	34.40± 1.51

Combining Multiple Feature Representations

- Internet Advertisements (ADVERT) data set
- Predict whether an image is an advertisement or not

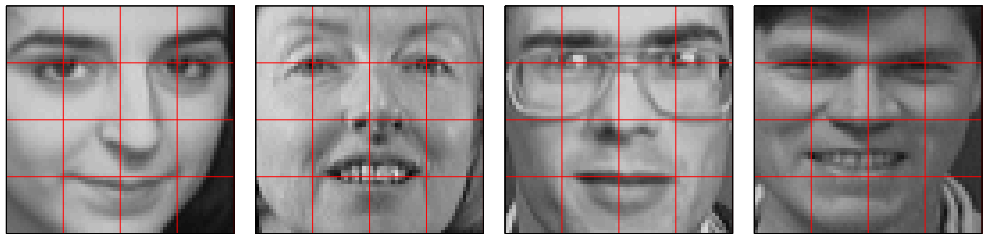
Name	Dim.	Data Source
URL	457	Phrases in the URL
ORIGURL	495	Phrases in the URL of the image
ANCURL	472	Phrases in the anchor text
ALT	111	Phrases in the alternative text
CAPTION	19	Phrases in the caption terms

Method	Test Accuracy	Support Vector
SVM (URL)	94.67±0.24	83.32± 1.89
SVM (ORIGURL)	92.04±0.26	96.16± 0.51
SVM (ANCURL)	95.45±0.31	64.90± 5.41
SVM (ALT)	89.64±0.38	87.73± 1.17
SVM (CAPTION)	86.60±0.09	96.65± 0.42
SVM (ALL)	96.43±0.24	41.99± 1.76
MKL	96.32±0.50	35.82± 4.35
LMKL (softmax)	95.78±0.46	41.72±11.59
LMKL (sigmoid)	96.72±0.46	34.40± 1.51



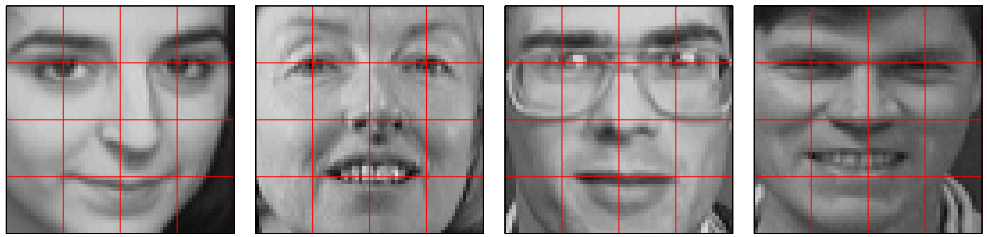
Combining Multiple Input Patches

- Gender recognition experiments (Gönen and Alpaydın, 2009)



Combining Multiple Input Patches

- Gender recognition experiments (Gönen and Alpaydın, 2009)



Method	Test Accuracy	Support Vector
SVM ($\boldsymbol{x} = 4 \times 4$)	93.28±0.65	21.70±0.93
SVM ($\boldsymbol{x} = 8 \times 8$)	97.50±1.16	20.13±1.04
SVM ($\boldsymbol{x} = 16 \times 16$)	97.03±0.93	19.91±1.01
SVM ($\boldsymbol{x} = 32 \times 32$)	97.97±1.48	23.71±1.39
SVM ($\boldsymbol{x} = 64 \times 64$)	97.66±1.41	25.94±1.01
MKL	99.06±0.88	22.19±1.00
LMKL (softmax and $\boldsymbol{x}^{\mathcal{G}} = 4 \times 4$)	97.19±2.81	16.65±3.34
LMKL (softmax and $\boldsymbol{x}^{\mathcal{G}} = 8 \times 8$)	97.19±2.48	22.54±4.56
LMKL (softmax and $\boldsymbol{x}^{\mathcal{G}} = 16 \times 16$)	99.22±1.33	16.38±1.50
LMKL (sigmoid and $\boldsymbol{x}^{\mathcal{G}} = 4 \times 4$)	99.22±0.93	22.72±1.83
LMKL (sigmoid and $\boldsymbol{x}^{\mathcal{G}} = 8 \times 8$)	99.84±0.44	26.88±2.24
LMKL (sigmoid and $\boldsymbol{x}^{\mathcal{G}} = 16 \times 16$)	99.38±1.34	21.65±1.44

Combining Multiple Input Patches

(a) (b) (c)

(a) $\Phi_m(\boldsymbol{x}^m)$: features fed into kernels

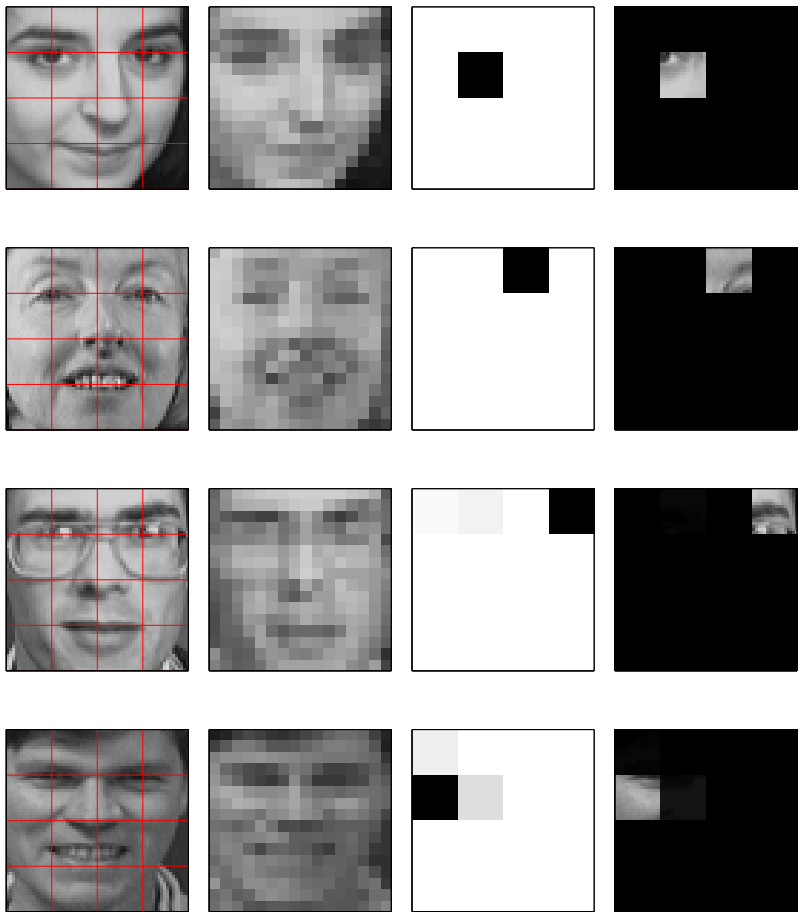
(b) η_m: MKL combination weights

(c) $\eta_m \Phi_m(\boldsymbol{x}^m)$: features weighted with MKL combination weights

- MKL uses the same weights over the whole input space

Combining Multiple Input Patches

Softmax Gating



(a) (b) (c) (d)

(a) $\Phi_m(\boldsymbol{x}^m)$: features fed into kernels

(b) $\boldsymbol{x}^{\mathcal{G}}$: features fed into softmax gating model

(c) $\eta_m(\boldsymbol{x}|\mathbf{V})$: softmax gating model outputs

(d) $\eta_m(\boldsymbol{x}|\mathbf{V})\Phi_m(\boldsymbol{x}^m)$: features weighted with softmax gating model outputs

- The softmax gating model generally activates a single patch
- Eyes or eyebrows depending on the subject

Combining Multiple Input Patches

Sigmoid Gating

(a) (b) (c) (d)

(a) $\Phi_m(\boldsymbol{x}^m)$: features fed into kernels

(b) $\boldsymbol{x}^{\mathcal{G}}$: features fed into sigmoid gating model

(c) $\eta_m(\boldsymbol{x}|\mathbf{V})$: sigmoid gating model outputs

(d) $\eta_m(\boldsymbol{x}|\mathbf{V})\Phi_m(\boldsymbol{x}^m)$: features weighted with sigmoid gating model outputs

- Multiple patches are given nonzero weights
- Drives a high-resolution "eye" to regions of high saliency

Combining Multiple Kernels

- Wine Quality (WHITEWINE) data set

Method	MSE	Support Vector
SVR (k_L)	0.59±0.00	66.83± 0.57
SVR (k_P and $q = 2$)	0.54±0.01	66.22± 0.67
SVR (k_P and $q = 3$)	0.54±0.00	66.14± 1.13
SVR (k_P and $q = 4$)	0.52±0.01	66.55± 1.03
SVR (k_P and $q = 5$)	0.52±0.01	66.27± 1.24
LMKL (softmax)	0.52±0.01	18.66±13.41
LMKL (sigmoid)	0.51±0.00	38.29± 2.34

Combining Multiple Kernels

- Wine Quality (WHITEWINE) data set
- Concrete Compressive Strength (CONCRETE) data set

Method	MSE	Support Vector
SVR (k_L)	0.59±0.00	66.83± 0.57
SVR (k_P and $q = 2$)	0.54±0.01	66.22± 0.67
SVR (k_P and $q = 3$)	0.54±0.00	66.14± 1.13
SVR (k_P and $q = 4$)	0.52±0.01	66.55± 1.03
SVR (k_P and $q = 5$)	0.52±0.01	66.27± 1.24
LMKL (softmax)	0.52±0.01	18.66±13.41
LMKL (sigmoid)	0.51±0.00	38.29± 2.34

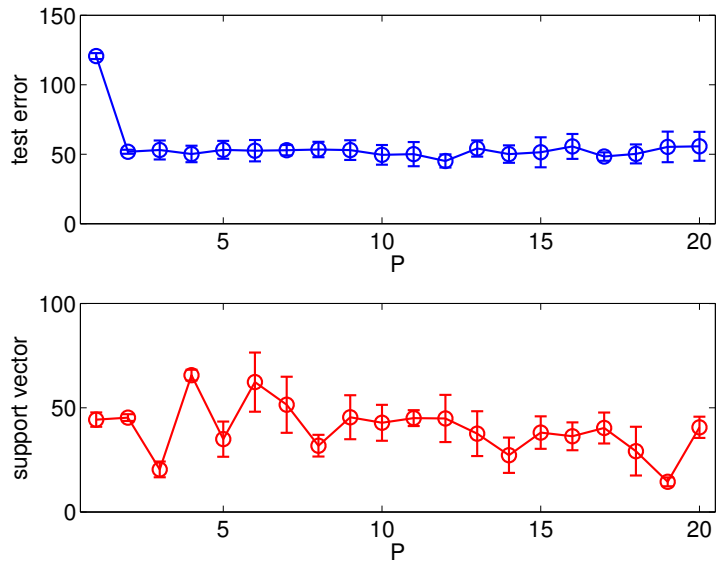
Method	MSE	Support Vector
SVR (k_L)	120.61±2.15	44.31±3.46
SVR (k_P and $q = 2$)	92.57±4.19	36.22±1.24
SVR (k_P and $q = 3$)	58.32±3.66	73.16±1.21
SVR (k_P and $q = 4$)	63.83±9.58	52.52±2.40
SVR (k_P and $q = 5$)	61.26±5.31	52.17±2.23
LMKL (softmax)	44.80±6.33	64.28±4.02
LMKL (sigmoid)	48.18±5.22	49.20±2.05

Combining Multiple Kernels

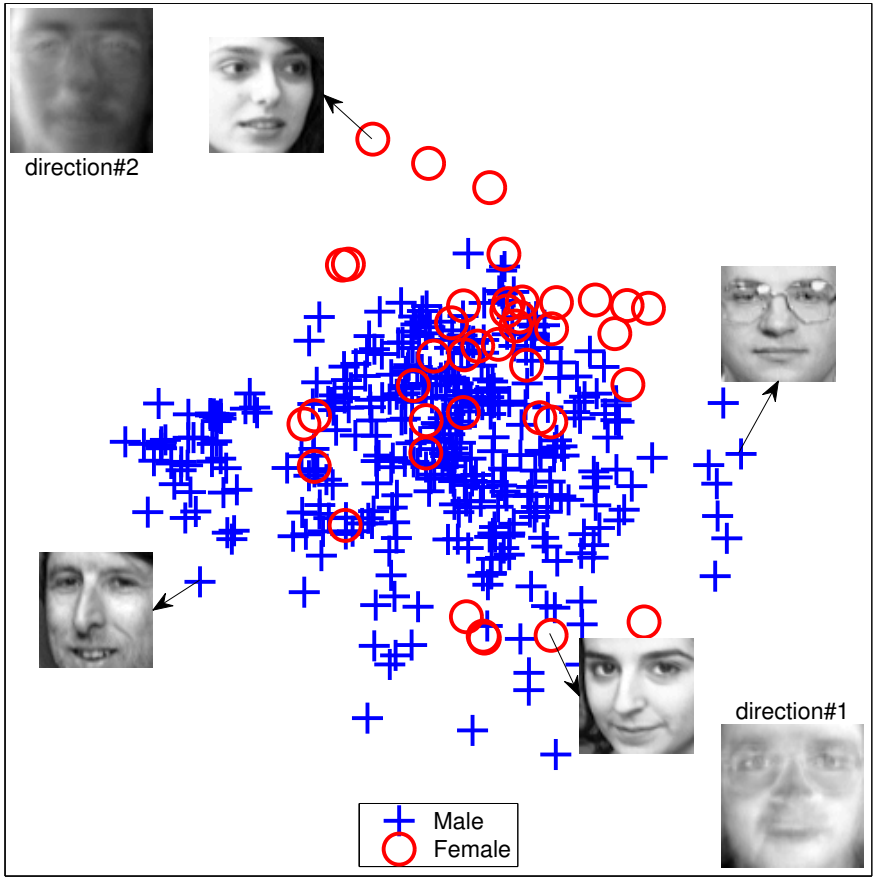
- Wine Quality (WHITEWINE) data set
- Concrete Compressive Strength (CONCRETE) data set

Method	MSE	Support Vector
SVR (k_L)	0.59±0.00	66.83± 0.57
SVR (k_P and $q = 2$)	0.54±0.01	66.22± 0.67
SVR (k_P and $q = 3$)	0.54±0.00	66.14± 1.13
SVR (k_P and $q = 4$)	0.52±0.01	66.55± 1.03
SVR (k_P and $q = 5$)	0.52±0.01	66.27± 1.24
LMKL (softmax)	0.52±0.01	18.66±13.41
LMKL (sigmoid)	0.51±0.00	38.29± 2.34

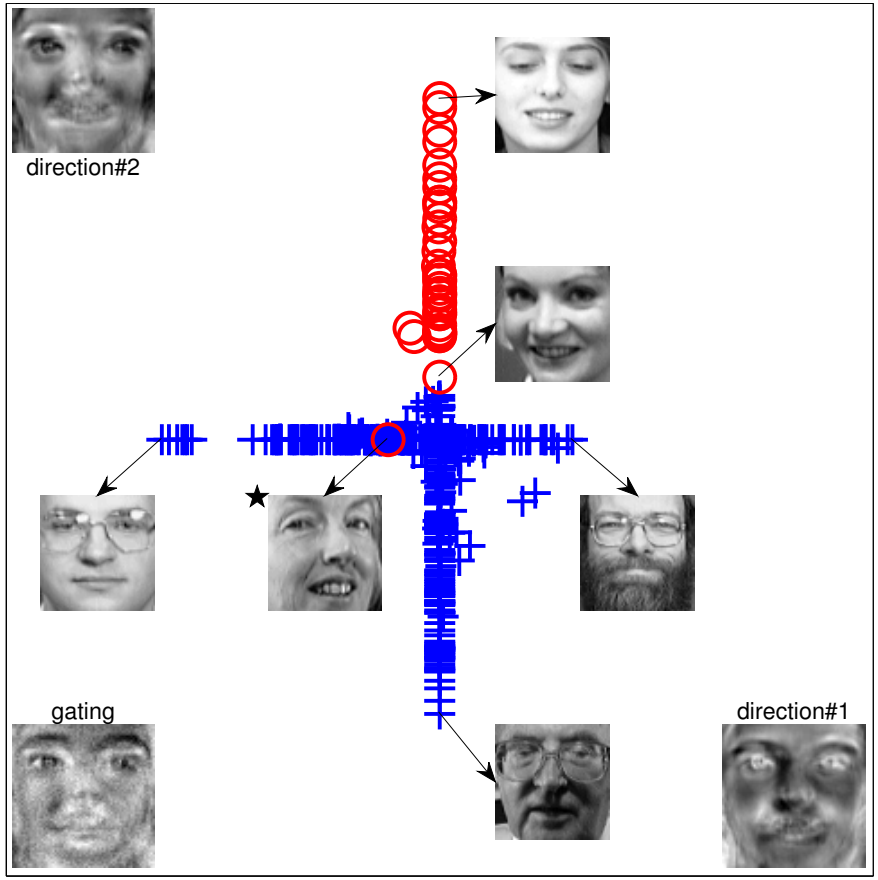
Method	MSE	Support Vector
SVR (k_L)	120.61±2.15	44.31±3.46
SVR (k_P and $q = 2$)	92.57±4.19	36.22±1.24
SVR (k_P and $q = 3$)	58.32±3.66	73.16±1.21
SVR (k_P and $q = 4$)	63.83±9.58	52.52±2.40
SVR (k_P and $q = 5$)	61.26±5.31	52.17±2.23
LMKL (softmax)	44.80±6.33	64.28±4.02
LMKL (sigmoid)	48.18±5.22	49.20±2.05



Visualization Experiments



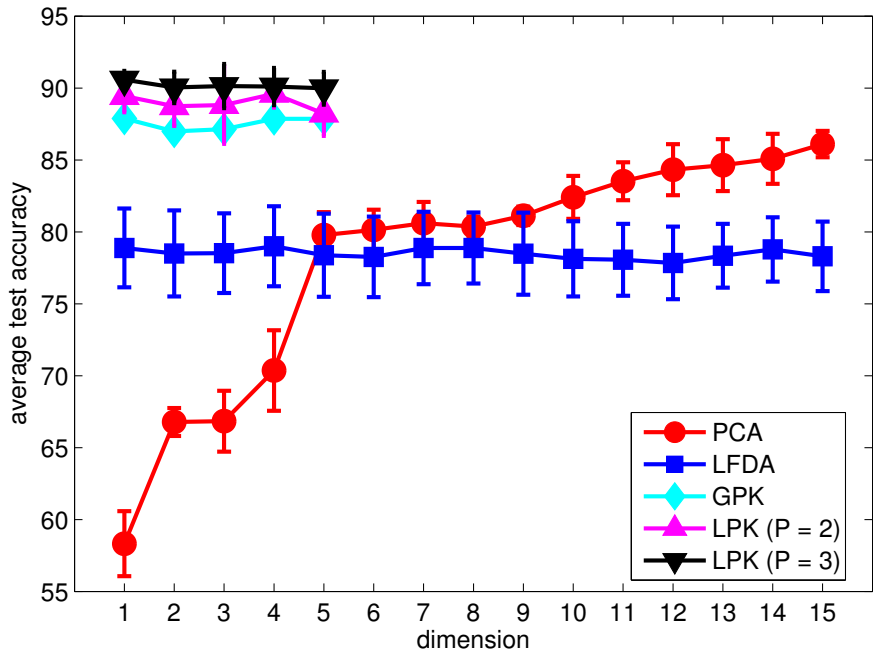
PCA

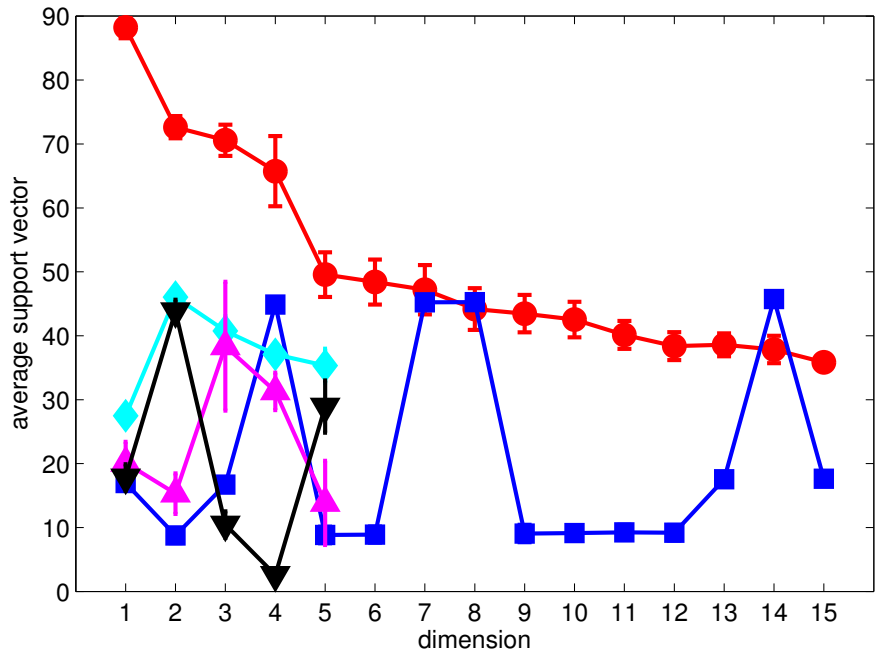


LPK

Classification Experiments

- We compare PCA, LFDA, GPK, and LPK
- USPS digit recognition data set
- Separate even digits from odd digits

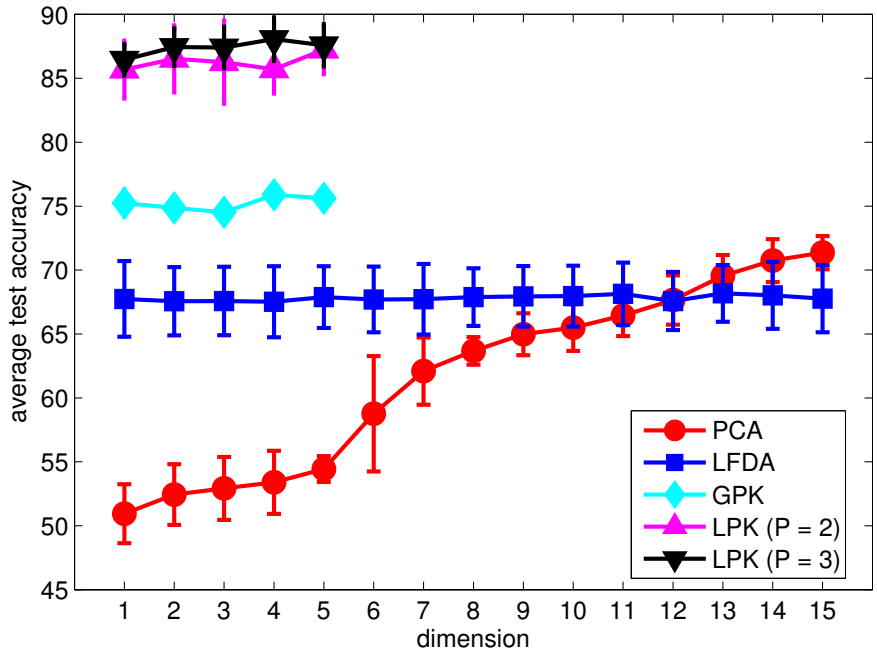


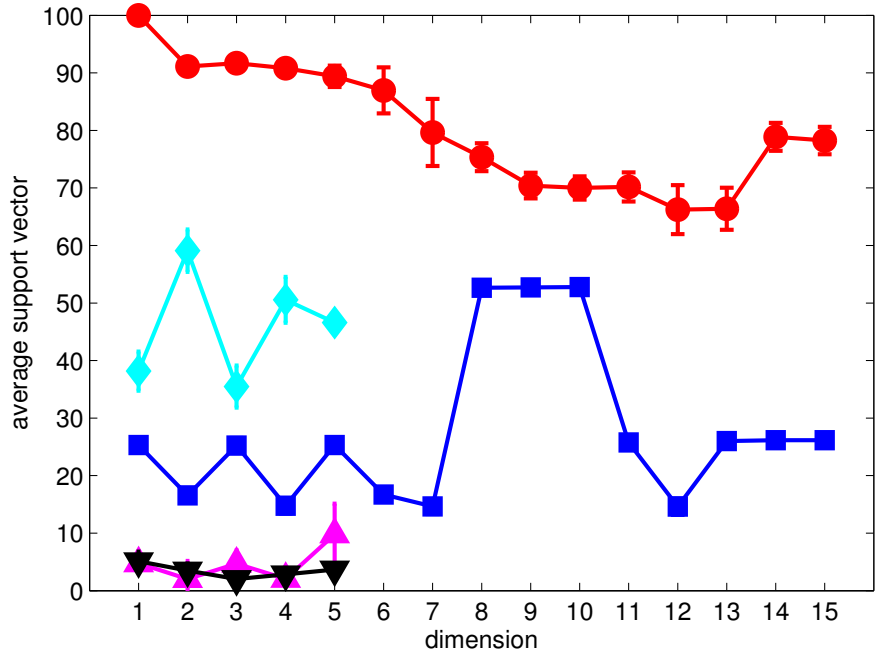


Classification Experiments

- USPS digit recognition data set

- Separate small (‘0’ - ‘4’) digits from large (‘5’ - ‘9’) digits

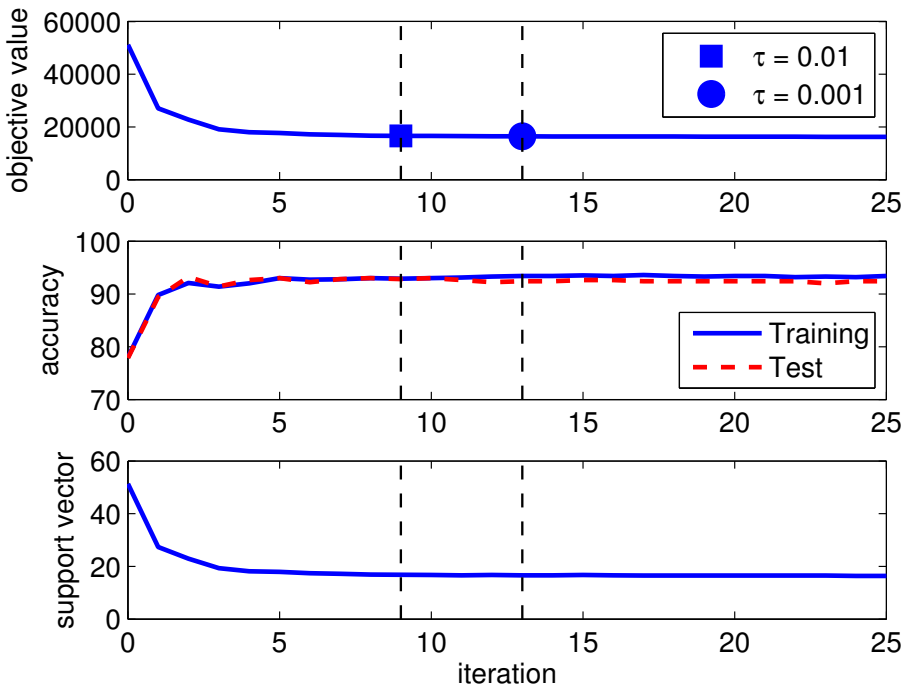




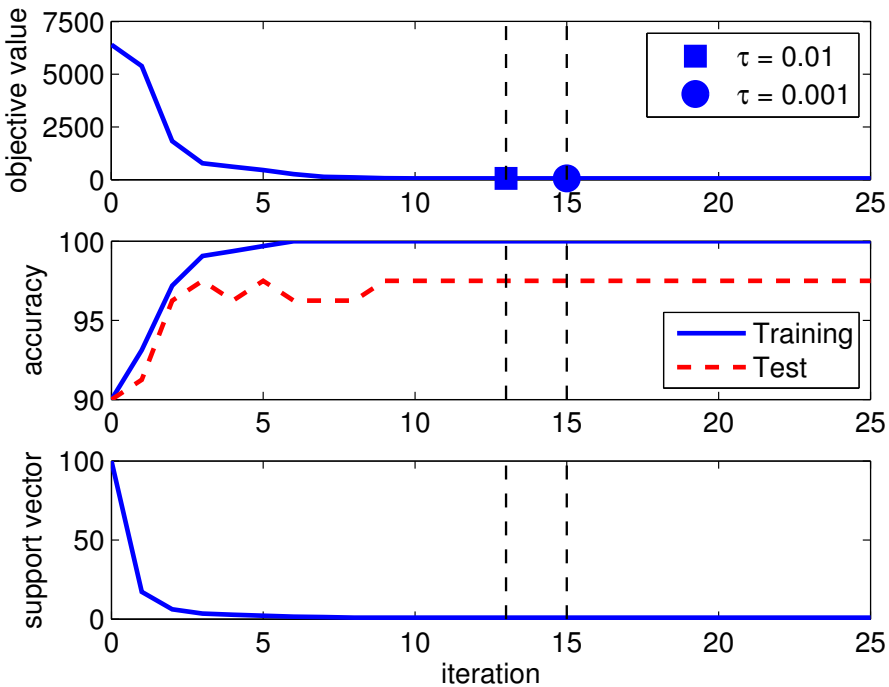
Convergence Analysis

- Convergence parameter τ

- `stop if (objective value)`$^{(t)}$ `>` $(1-\tau)$`(objective value)`$^{(t-1)}$

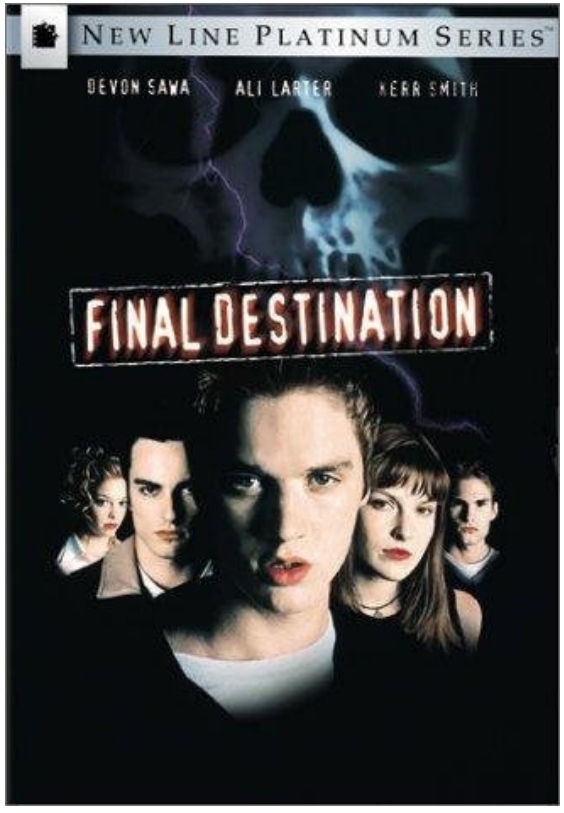


Waveform



Olivetti

Outline



Conclusions

- Regularized Multiple Kernel Learning
  - optimizes regularization parameters using RSM
  - obtains smoother decision functions
  - uses fewer kernels and support vectors
  - has advantage of RSM over grid search

Conclusions

- Regularized Multiple Kernel Learning
  - optimizes regularization parameters using RSM
  - obtains smoother decision functions
  - uses fewer kernels and support vectors
  - has advantage of RSM over grid search
- Cost-Conscious Multiple Kernel Learning
  - includes the costs of kernels
  - eliminates costlier data representations

Conclusions

- Localized Multiple Kernel Learning
  - input-dependent selection among multiple kernels or feature representations
  - two main ingredients
    - gating model for data-dependent kernel weights
    - kernel-based learning algorithm
  - two-step alternating optimization algorithm
  - applied to different gating models and learning problems

Conclusions

- Localized Multiple Kernel Learning
  - input-dependent selection among multiple kernels or feature representations
  - two main ingredients
    - gating model for data-dependent kernel weights
    - kernel-based learning algorithm
  - two-step alternating optimization algorithm
  - applied to different gating models and learning problems
  - obtains better learners compared to MKL
  - improves accuracy and/or stores fewer support vectors
  - can combine multiple copies of the same kernel
  - acts as a saliency detector on image recognition problems

Conclusions

- Localized Multiple Kernel Learning
  - computational complexity
    - complexity of canonical solver & number of iterations
    - Armijo's rule
    - improvement in test time

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \underbrace{\sum_{m=1}^{P} \eta_m(\boldsymbol{x}_i|\mathbf{V}) k_m(\boldsymbol{x}_i^m, \boldsymbol{x}^m) \eta_m(\boldsymbol{x}|\mathbf{V})}_{k_\eta(\boldsymbol{x}_i, \boldsymbol{x})} + b$$

Conclusions

- Localized Multiple Kernel Learning
  - computational complexity
    - complexity of canonical solver & number of iterations
    - Armijo's rule
    - improvement in test time

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \underbrace{\sum_{m=1}^{P} \eta_m(\boldsymbol{x}_i|\mathbf{V}) k_m(\boldsymbol{x}_i^m, \boldsymbol{x}^m) \eta_m(\boldsymbol{x}|\mathbf{V})}_{k_\eta(\boldsymbol{x}_i, \boldsymbol{x})} + b$$

  - knowledge extraction
    - data-dependent relative importances

Conclusions

- Localized Multiple Kernel Learning
  - computational complexity
    - complexity of canonical solver & number of iterations
    - Armijo's rule
    - improvement in test time

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \underbrace{\sum_{m=1}^{P} \eta_m(\boldsymbol{x}_i|\mathbf{V}) k_m(\boldsymbol{x}_i^m, \boldsymbol{x}^m) \eta_m(\boldsymbol{x}|\mathbf{V})}_{k_\eta(\boldsymbol{x}_i, \boldsymbol{x})} + b$$

  - knowledge extraction
    - data-dependent relative importances
  - regularization
    - regularization on locally combined kernel
    - no overfitting with increasing number of kernels

Conclusions

- Local Projection Kernels
  - localized dimensionality reduction on multimodal data sets
  - improves accuracy and/or stores fewer support vectors
  - computational complexity

Conclusions

- Local Projection Kernels
  - localized dimensionality reduction on multimodal data sets
  - improves accuracy and/or stores fewer support vectors
  - computational complexity
- Future work
  - using RSM on validation performance for model selection problems
  - Bayesian reformulations of LMKL and LPK

Conclusions

- Local Projection Kernels
  - localized dimensionality reduction on multimodal data sets
  - improves accuracy and/or stores fewer support vectors
  - computational complexity
- Future work
  - using RSM on validation performance for model selection problems
  - Bayesian reformulations of LMKL and LPK
- MATLAB source codes

`http://www.cmpe.boun.edu.tr/~gonen`

References I

Bach, F. R., Lanckriet, G. R. G. and Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the SMO algorithm, *Proceedings of the 21st International Conference on Machine Learning*.

Bach, F. R., Thibaux, R. and Jordan, M. I. (2005). Computing regularization paths for learning multiple kernels, *Advances in Neural Information Processing Systems 17*.

Bredensteiner, E. J. and Bennett, K. P. (1999). Multicategory classification by support vector machines, *Computational Optimization and Applications* **12**(1-3): 53–79.

Chapelle, O., Vapnik, V., Bousquet, O. and Mukherjee, S. (2002). Choosing multiple parameters for support vector machines, *Machine Learning* **46**(1–3): 131–159.

Gönen, M. and Alpaydın, E. (2008). Localized multiple kernel learning, *Proceedings of the 25th International Conference on Machine Learning*.

Gönen, M. and Alpaydın, E. (2009). Localized multiple kernel learning for image recognition, *NIPS Workshop on Understanding Multiple Kernel Learning Methods*.

Gönen, M. and Alpaydın, E. (2010a). Cost-conscious multiple kernel learning, *Pattern Recognition Letters* **31**(9): 959–965.

Gönen, M. and Alpaydın, E. (2010b). Localized multiple kernel regression, *Proceedings of the 20th International Conference on Pattern Recognition*.

Gönen, M. and Alpaydın, E. (2010c). Supervised learning of local projection kernels, *Neurocomputing* **73**(10–12): 1694–1703.

Gönen, M., Tanuğur, A. G. and Alpaydın, E. (2008). Multiclass posterior probability support vector machines, *IEEE Transactions on Neural Networks* **19**(1): 130–139.

Gromiha, M. M., An, J., Kono, H., Oobatake, M., Uedaira, H., Prabakaran, P. and Sarai, A. (2000). Protherm, version 2.0: Thermodynamic database for proteins and mutants, *Nucleic Acids Research* **28**: 283–285.

References II

Jacobs, R. A., Jordan, M. I., Nowlan, S. J. and Hinton, G. E. (1991). Adaptive mixtures of local experts, *Neural Computation* **3**: 79–87.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E. and Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming, *Journal of Machine Learning Research* **5**: 27–72.

Noble, W. S. (2004). Support vector machine applications in computational biology, *in* B. Schölkopf, K. Tsuda and J.-P. Vert (eds), *Kernel Methods in Computational Biology*, The MIT Press, chapter 3.

Özen, A., Gönen, M., Alpaydın, E. and Haliloğlu, T. (2009). Machine learning integration for predicting the effect of single amino acid substitutions on protein stability, *BMC Structural Biology* **9**: 66.

Pavlidis, P., Weston, J., Cai, J. and Grundy, W. N. (2001). Gene functional classification from heterogeneous data, *Proceedings of the 5th Annual International Conference on Computational Molecular Biology*.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, Cambridge, MA.

Tao, Q., Wu, G.-W., Wang, F.-Y. and Wang, J. (2005). Posterior probability support vector machines for unbalanced data, *IEEE Transactions on Neural Networks* **16**(6): 1561–1573.

Vapnik, V. (1998). *The Nature of Statistical Learning Theory*, John Wiley & Sons.

Weston, J. and Watkins, C. (1998). Multi-class support vector machines, *Technical Report CSD-TR-98-04*, Royal Holloway, University of London, Department of Computer Science.