Binary Classification

$y_i = +1$

$y_i = -1$
Motivation

Binary Classification

\[ y_i = +1 \]

\[ y_i = -1 \]
Binary Classification

\[ y_i = -1 \]

\[ y_i = +1 \]
Binary Classification

Motivation

\[ y_i = +1 \]

\[ \langle w, x_1 \rangle + b = +1 \]

\[ \langle w, x_2 \rangle + b = -1 \]

\[ \langle w, x_1 - x_2 \rangle = 2 \]

\[ \langle \frac{w}{\|w\|}, x_1 - x_2 \rangle = \frac{2}{\|w\|} \]

\[ \{ x \mid \langle w, x \rangle + b = 1 \} \]

\[ \{ x \mid \langle w, x \rangle + b = -1 \} \]

\[ \|w\| \]
Motivation

Linear Support Vector Machines

Optimization Problem

\[
\min_{w,b,\xi} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i \\
\text{s.t.} \quad y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \text{ for all } i \\
\quad \xi_i \geq 0
\]
The Kernel Trick
The Kernel Trick
Motivation

The Kernel Trick

\[ x^2 + y^2 \]
Kernel Trick

Optimization Problem

$$\min_{w, b, \xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i$$

s.t. \quad y_i (\langle w, \phi(x_i) \rangle + b) \geq 1 - \xi_i \text{ for all } i$

\quad \xi_i \geq 0$
Motivation

Kernel Trick

Optimization Problem

\[
\begin{align*}
\max_{\alpha} & \quad -\frac{1}{2} \alpha^\top H \alpha + 1^\top \alpha \\
\text{s.t.} & \quad 0 \leq \alpha_i \leq C \\
& \quad \sum_i \alpha_i y_i = 0 \\
H_{ij} & = y_i y_j \langle \phi(x_i), \phi(x_j) \rangle
\end{align*}
\]
Optimization Problem

\[
\begin{align*}
\max_{\alpha} & \quad -\frac{1}{2} \alpha^\top H\alpha + 1^\top \alpha \\
\text{s.t.} & \quad 0 \leq \alpha_i \leq C \\
& \quad \sum_i \alpha_i y_i = 0 \\

H_{ij} & = y_i y_j k(x_i, x_j)
\end{align*}
\]
Key Question

Which kernel should I use?

The Multiple Kernel Learning Answer

- Cook up as many (base) kernels as you can
- Compute a data dependent kernel function as a linear combination of base kernels

\[
k(x, x') = \sum_k d_k k_k(x, x') \quad \text{s.t. } d_k \geq 0
\]
Key Question

Which kernel should I use?

The Multiple Kernel Learning Answer

- Cook up as many (base) kernels as you can
- Compute a data dependent kernel function as a linear combination of base kernels

\[ k(x, x') = \sum_k d_k k_k(x, x') \quad \text{s.t.} \quad d_k \geq 0 \]
Object Detection

Localize a specified object of interest if it exists in a given image
Some Examples of MKL Detection
Summary of Our Results

- Sonar Dataset with 800 kernels

<table>
<thead>
<tr>
<th>$p$</th>
<th>Training Time (s)</th>
<th># Kernels Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SMO-MKL</td>
<td>Shogun</td>
</tr>
<tr>
<td>1.1</td>
<td>4.71</td>
<td>47.43</td>
</tr>
<tr>
<td>1.33</td>
<td>3.21</td>
<td>19.94</td>
</tr>
<tr>
<td>2.0</td>
<td>3.39</td>
<td>34.67</td>
</tr>
</tbody>
</table>

- Web dataset: $\approx 50,000$ points and 50 kernels $\approx 30$ minutes

- Sonar with a hundred thousand kernels
  - Precomputed: $\approx 8$ minutes
  - Kernels computed on-the-fly: $\approx 30$ minutes
The Setup

- We are given $K$ kernel functions $k_1, \ldots, k_n$ with corresponding feature maps $\phi_1(\cdot), \ldots, \phi_n(\cdot)$
- We are interested in deriving the feature map

$$
\phi(x) = \begin{bmatrix}
\sqrt{d_1}\phi_1(x) \\
\vdots \\
\sqrt{d_n}\phi_n(x)
\end{bmatrix}
$$
Setting up the Optimization Problem - I

The Setup

- We are given $K$ kernel functions $k_1, \ldots, k_n$ with corresponding feature maps $\phi_1(\cdot), \ldots, \phi_n(\cdot)$
- We are interested in deriving the feature map

$$
\phi(x) = \begin{bmatrix}
\sqrt{d_1} \phi_1(x) \\
\vdots \\
\sqrt{d_n} \phi_n(x)
\end{bmatrix} \implies w = \begin{bmatrix}
w_1 \\
\vdots \\
w_n
\end{bmatrix}
$$
Setting up the Optimization Problem

Optimization Problem

\[
\begin{align*}
\min_{w, b, \xi} & \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i \\
\text{s.t.} & \quad y_i (\langle w, \phi(x_i) \rangle + b) \geq 1 - \xi_i \text{ for all } i \\
& \quad \xi_i \geq 0
\end{align*}
\]
Setting up the Optimization Problem

Optimization Problem

\[ \min_{w, b, \xi, d} \frac{1}{2} \sum_k \| w_k \|^2 + C \sum_{i=1}^{m} \xi_i \]

s.t. \[ y_i \left( \sum_k \sqrt{d_k} \langle w_k, \phi_k(x_i) \rangle + b \right) \geq 1 - \xi_i \text{ for all } i \]

\[ \xi_i \geq 0 \]

\[ d_k \geq 0 \text{ for all } k \]
Setting up the Optimization Problem

Optimization Problem

\[
\begin{align*}
\min_{w,b,\xi,d} & \quad \frac{1}{2} \sum_{k} \|w_k\|^2 + C \sum_{i=1}^{m} \xi_i + \frac{\rho}{2} \left( \sum_{k} d_k^p \right)^{\frac{2}{p}} \\
\text{s.t.} & \quad y_i \left( \sum_{k} \sqrt{d_k} \langle w_k, \phi_k(x_i) \rangle + b \right) \geq 1 - \xi_i \quad \text{for all } i \\
& \quad \xi_i \geq 0 \\
& \quad d_k \geq 0 \quad \text{for all } k
\end{align*}
\]
Motivation

Setting up the Optimization Problem

Optimization Problem

\[
\begin{align*}
\min_{w,b,\xi,d} & \quad \frac{1}{2} \sum_k \frac{\|w_k\|^2}{d_k} + C \sum_{i=1}^m \xi_i + \frac{\rho}{2} \left( \sum_k d_k^p \right)^{2/p} \\
\text{s.t.} & \quad y_i \left( \sum_k \langle w_k, \phi_k(x_i) \rangle + b \right) \geq 1 - \xi_i \text{ for all } i \\
& \quad \xi_i \geq 0 \\
& \quad d_k \geq 0 \text{ for all } k
\end{align*}
\]
Setting up the Optimization Problem

Optimization Problem

\[
\min_{\mathbf{d}} \max_{\mathbf{\alpha}} \quad -\frac{1}{2} \sum_k d_k \mathbf{\alpha}^\top H_k \mathbf{\alpha} + \mathbf{1}^\top \mathbf{\alpha} + \frac{\rho}{2} \left( \sum_k d_k^p \right)^{\frac{2}{p}} \\
\text{s.t.} \quad 0 \leq \alpha_i \leq C \\
\quad \sum_i \alpha_i y_i = 0 \\
\quad d_k \geq 0
\]
Saddle Point Problem
Solving the Saddle Point

Saddle Point Problem

$$\min_d \max_{\alpha} - \frac{1}{2} \sum_k d_k \alpha^\top H_k \alpha + 1^\top \alpha + \frac{\rho}{2} \left( \sum_k d_k^p \right)^{\frac{2}{p}}$$

s.t. \quad 0 \leq \alpha_i \leq C

$$\sum_i \alpha_i y_i = 0$$

$$d_k \geq 0$$
Our Approach

The Key Insight

Eliminate \( d \)

\[
D(\alpha) := \max_{\alpha} \left( -\frac{1}{8 \rho} \left( \sum_k \left( \alpha^\top H_k \alpha \right)^q \right)^{\frac{2}{q}} + 1^\top \alpha \right)
\]

s.t. \[0 \leq \alpha_i \leq C\]
\[\sum_i \alpha_i y_i = 0\]
\[\frac{1}{p} + \frac{1}{q} = 1\]

Not a QP but very close to one!
Our Approach

SMO-MKL: High Level Overview

\[
D(\alpha) := \max_{\alpha} \left( \sum_k \left( \alpha^\top H_k \alpha \right)^q \right)^{\frac{2}{q}} + \mathbf{1}^\top \alpha
\]

s.t. \[0 \leq \alpha_i \leq C\]
\[\sum_i \alpha_i y_i = 0\]

Algorithm

- Choose two variables \(\alpha_i\) and \(\alpha_j\) to optimize
- Solve the one dimensional reduced optimization problem
- Repeat until convergence
Our Approach

SMO-MKL: High Level Overview

Selecting the Working Set

- Compute directional derivative and directional Hessian
- Greedily select the variables

Solving the Reduced Problem

- Analytic solution for $p = q = 2$ (one dimensional quartic)
- For other values of $p$ use Newton Raphson
Our Approach

SMO-MKL: High Level Overview

Selecting the Working Set
- Compute directional derivative and directional Hessian
- Greedily select the variables

Solving the Reduced Problem
- Analytic solution for $p = q = 2$ (one dimensional quartic)
- For other values of $p$ use Newton Raphson
Experiments

Generalization Performance

![Australian test accuracy chart]

Test Accuracy (%)

- SMO-MKL
- Shogun

Australian

Test Accuracy (%)

- 1.1
- 1.33
- 1.66
- 2.0
- 2.33
- 2.66
- 3.0

80 82 84 86 88 90
Generalization Performance

Experiments

Test Accuracy (%) vs. SMO-MKL and Shogun for ionosphere dataset.
Scaling with Training Set Size

Adult: 123 dimensions, 50 RBF kernels, $p = 1.33$, $C = 1$
Experiments

Scaling with Training Set Size

Adult: 123 dimensions, 50 RBF kernels, $p = 1.33$, $C = 1$
Experiments

On Another Dataset

Web: 300 dimensions, 50 RBF kernels, $p = 1.33$, $C = 1$

![Graph showing CPU time in seconds vs. number of training examples. The x-axis represents the number of training examples in logarithmic scale, ranging from $10^{3.5}$ to $10^{4.5}$. The y-axis represents CPU time in logarithmic scale, ranging from $10^1$ to $10^3$. The data points show a trend of increasing CPU time with increasing number of training examples.](chart.png)
Scaling with Number of Kernels

Sonar: 208 examples, 59 dimensions, $p = 1.33$, $C = 1$
Scaling with Number of Kernels

Sonar: 208 examples, 59 dimensions, $p = 1.33$, $C = 1$

The graph shows the CPU time in seconds for SMO-MKL and $O(n)$ as a function of the number of kernels.

- SMO-MKL
- $O(n)$
Experiments

On Another Dataset

Real-sim: 72,309 examples, 20,958 dimensions, $p = 1.33$, $C = 1$

Number of Training Examples

CPU Time in seconds

$O(n)$

SMO-MKL
References
