Two SVM tutorials linked in class website (please, read both):
- High-level presentation with applications (Hearst 1998)
- Detailed tutorial (Burges 1998)

SVMs, Duality and the Kernel Trick

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

February 27th, 2006
Announcements

- Third homework
  - is out
  - Due March 1\textsuperscript{st}  

- Final assigned by registrar:
  - May 12, 1-4p.m \underline{Friday}
  - Location TBD

- Midterm
  - March 8\textsuperscript{th}, a week from Wednesday
  - Open book, notes, papers, etc. No computers
SVMs reminder

\[
\begin{align*}
\text{minimize}_{w} & \quad w \cdot w + C \sum_j \xi_j \\
- (w \cdot x_j + b) y_j & \geq 1 - \xi_j, \quad \forall j \\
\xi_j & \geq 0, \quad \forall j
\end{align*}
\]
Today’s lecture

- Learn one of the most interesting and exciting recent advancements in machine learning
  - The “kernel trick”
  - High dimensional feature spaces at no extra cost!
- But first, a detour
  - Constrained optimization!
Constrained optimization

\[ \min_x \quad x^2 \]
\[ \text{s.t.} \quad x \geq b \]
Lagrange multipliers – Dual variables

\[ \begin{align*} 
\min_x & \quad x^2 \\
\text{s.t.} & \quad x \geq b 
\end{align*} \]

Moving the constraint to objective function Lagrangian:

\[ L(x, \alpha) = x^2 - \alpha(x - b) \]

\[ \text{s.t.} \quad \alpha \geq 0 \]

Solve:

\[ \begin{align*} 
\min_x & \quad \max_\alpha \quad L(x, \alpha) \\
\text{s.t.} & \quad \alpha \geq 0 
\end{align*} \]
Lagrange multipliers – Dual variables

Solving: \( \min_x \max_\alpha \; x^2 - \alpha(x - b) \)
\[ \text{s.t. } \alpha \geq 0 \]
Dual SVM derivation (1) – the linearly separable case

\[
\text{minimize}_w \quad \frac{1}{2} w \cdot w \\
(w \cdot x_j + b) y_j \geq 1, \ \forall j
\]
Dual SVM derivation (2) – the linearly separable case

\[ L(w, \alpha) = \frac{1}{2} w \cdot w - \sum_j \alpha_j \left[ (w \cdot x_j + b) y_j - 1 \right] \]
\[ \alpha_i \geq 0, \forall j \]

\[ w = \sum_i \alpha_i y_i x_i \]

minimize \( w \quad \frac{1}{2} w \cdot w \)
\( (w \cdot x_j + b) y_j \geq 1, \forall j \)

\[ b = y_k - w \cdot x_k \]
for any \( k \) where \( \alpha_k > 0 \)
Dual SVM interpretation

\[ w \cdot x + b = 0 \]

\[ w = \sum_{i} \alpha_i y_i x_i \]
Dual SVM formulation – the linearly separable case

\[
\text{minimize}_{\alpha} \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j \\
\sum_i \alpha_i y_i = 0 \\
\alpha_i \geq 0
\]

\[
w = \sum_i \alpha_i y_i x_i \\
b = y_k - w \cdot x_k
\]
for any \( k \) where \( \alpha_k > 0 \)
Dual SVM derivation – the non-separable case

\[ \text{minimize}_{w} \quad w \cdot w + C \sum_j \xi_j \]

\[ (w \cdot x_j + b) y_j \geq 1 - \xi_j, \quad \forall j \]

\[ \xi_j \geq 0, \quad \forall j \]
Dual SVM formulation – the non-separable case

\[
\text{minimize} \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j
\]

\[
\sum_i \alpha_i y_i = 0
\]

\[
C \geq \alpha_i \geq 0
\]

\[
\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i
\]

\[
b = y_k - \mathbf{w} \cdot \mathbf{x}_k
\]

for any \( k \) where \( C > \alpha_k > 0 \)
Why did we learn about the dual SVM?

There are some quadratic programming algorithms that can solve the dual faster than the primal.

But, more importantly, the “kernel trick”!!!

Another little detour…
Reminder from last time: What if the data is not linearly separable?

Use features of features of features of features...

\[ \Phi(x) : \mathbb{R}^m \rightarrow F \]

Feature space can get really large really quickly!
Higher order polynomials

\[ \text{num. terms} = \binom{d + m - 1}{d} = \frac{(d + m - 1)!}{d!(m - 1)!} \]

\( m \) – number of input features
\( d \) – degree of polynomial

\( d = 2 \)
\( d = 3 \)
\( d = 4 \)

number of input dimensions

number of monomial terms

grows fast!
\( d = 6, m = 100 \)
about 1.6 billion terms
Dual formulation only depends on dot-products, not on \( \mathbf{w} \!\). 

\[
\text{minimize}_{\alpha} \quad \sum_{i} \alpha_{i} - \frac{1}{2} \sum_{i,j} \alpha_{i} \alpha_{j} y_{i} y_{j} x_{i} x_{j} \\
\sum_{i} \alpha_{i} y_{i} = 0 \\
C \geq \alpha_{i} \geq 0
\]

\[
\text{minimize}_{\alpha} \quad \sum_{i} \alpha_{i} - \frac{1}{2} \sum_{i,j} \alpha_{i} \alpha_{j} y_{i} y_{j} K(\mathbf{x}_{i}, \mathbf{x}_{j}) \\
K(\mathbf{x}_{i}, \mathbf{x}_{j}) = \Phi(\mathbf{x}_{i}) \cdot \Phi(\mathbf{x}_{j}) \\
\sum_{i} \alpha_{i} y_{i} = 0 \\
C \geq \alpha_{i} \geq 0
\]
Dot-product of polynomials

\[ \Phi(u) \cdot \Phi(v) = \text{polynomials of degree } d \]
Finally: the “kernel trick”!

\[
\begin{align*}
\text{minimize } & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\
K(x_i, x_j) & = \Phi(x_i) \cdot \Phi(x_j) \\
\sum_i \alpha_i y_i & = 0 \\
C' & \geq \alpha_i \geq 0
\end{align*}
\]

- Never represent features explicitly
  - Compute dot products in closed form
- Constant-time high-dimensional dot-products for many classes of features
- Very interesting theory – Reproducing Kernel Hilbert Spaces
  - Not covered in detail in 10701/15781, more in 10702

\[
\begin{align*}
w & = \sum_i \alpha_i y_i \Phi(x_i) \\
b & = y_k - w \cdot \Phi(x_k) \\
& \text{for any } k \text{ where } C > \alpha_k > 0
\end{align*}
\]
Polynomial kernels

- All monomials of degree $d$ in $O(d)$ operations:
  $\Phi(u) \cdot \Phi(v) = (u \cdot v)^d = \text{polynomials of degree } d$

- How about all monomials of degree up to $d$?
  - Solution 0:
  
  - Better solution:
Common kernels

- Polynomials of degree $d$
  \[ K(u, v) = (u \cdot v)^d \]

- Polynomials of degree up to $d$
  \[ K(u, v) = (u \cdot v + 1)^d \]

- Gaussian kernels
  \[ K(u, v) = \exp\left(-\frac{||u - v||}{2\sigma^2}\right) \]

- Sigmoid
  \[ K(u, v) = \tanh(\eta u \cdot v + \nu) \]
Overfitting?

- Huge feature space with kernels, what about overfitting???
  - Maximizing margin leads to sparse set of support vectors
  - Some interesting theory says that SVMs search for simple hypothesis with large margin
  - Often robust to overfitting
What about at classification time

- For a new input $\mathbf{x}$, if we need to represent $\Phi(\mathbf{x})$, we are in trouble!
- Recall classifier: $\text{sign}(\mathbf{w} \cdot \Phi(\mathbf{x}) + b)$
- Using kernels we are cool!

$$K(\mathbf{u}, \mathbf{v}) = \Phi(\mathbf{u}) \cdot \Phi(\mathbf{v})$$

$$\mathbf{w} = \sum_i \alpha_i y_i \Phi(\mathbf{x}_i)$$

$$b = y_k - \mathbf{w} \cdot \Phi(\mathbf{x}_k)$$

for any $k$ where $C > \alpha_k > 0$
SVMs with kernels

- Choose a set of features and kernel function
- Solve dual problem to obtain support vectors $\alpha_i$
- At classification time, compute:

$$w \cdot \Phi(x) = \sum_i \alpha_i y_i K(x, x_i)$$

$$b = y_k - \sum_i \alpha_i y_i K(x_k, x_i)$$

for any $k$ where $C > \alpha_k > 0$

Classify as $\text{sign} (w \cdot \Phi(x) + b)$
What’s the difference between SVMs and Logistic Regression?

<table>
<thead>
<tr>
<th></th>
<th>SVMs</th>
<th>Logistic Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss function</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High dimensional features with kernels</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Logistic Regression

High-dimensional features with kernels
Kernels in logistic regression

\[ P(Y = 1 \mid x, w) = \frac{1}{1 + e^{-(w \cdot \Phi(x) + b)}} \]

- Define weights in terms of support vectors:
  \[ w = \sum_{i} \alpha_i \Phi(x_i) \]

\[ P(Y = 1 \mid x, w) = \frac{1}{1 + e^{-\left(\sum_{i} \alpha_i \Phi(x_i) \cdot \Phi(x) + b\right)}} = \frac{1}{1 + e^{-\left(\sum_{i} \alpha_i K(x,x_i) + b\right)}} \]

- Derive simple gradient descent rule on \( \alpha_i \)
What’s the difference between SVMs and Logistic Regression? (Revisited)

<table>
<thead>
<tr>
<th></th>
<th>SVMs</th>
<th>Logistic Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss function</td>
<td>Hinge loss</td>
<td>Log-loss</td>
</tr>
<tr>
<td>High dimensional features with kernels</td>
<td>Yes!</td>
<td>Yes!</td>
</tr>
</tbody>
</table>

Almost always no! Often yes!
Solution sparse
Real probabilities "margin"
Semantics of output
Loss function
Logistic Regression
SVMs
What you need to know

- Dual SVM formulation
  - How it’s derived
- The kernel trick
- Derive polynomial kernel
- Common kernels
- Kernelized logistic regression
- Differences between SVMs and logistic regression
Acknowledgment

- SVM applet:
  - [http://www.site.uottawa.ca/~gcaron/applets.htm](http://www.site.uottawa.ca/~gcaron/applets.htm)