Lecture 17:

- Multi-class SVMs
- Kernels
Administrative

• We will have a make-up lecture on Saturday December 17, 2016 (I will check the date today).

• Project progress reports are due today!
Last time... Support Vector Machines

\[ \langle w, x \rangle + b \leq -1 \]

linear function

\[ f(x) = \langle w, x \rangle + b \]

\[ \langle w, x \rangle + b \geq 1 \]
Last time… **Support Vector Machines**

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

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

optimization problem

\[
\max_{w,b} \frac{1}{\|w\|} \quad \text{subject to } y_i \left[ \langle x_i, w \rangle + b \right] \geq 1
\]
Last time... Support Vector Machines

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

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

optimization problem

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|w\|^2 \\
\text{subject to} & \quad y_i \left[ \langle x_i, w \rangle + b \right] \geq 1
\end{align*}
\]
Last time... **Support Vector Machines**

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|w\|^2 \quad \text{subject to} \quad y_i \left[ \langle x_i, w \rangle + b \right] \geq 1 \\
\end{align*}
\]

\[
\begin{align*}
w & = \sum_i y_i \alpha_i x_i \\
\end{align*}
\]

\[
\begin{align*}
\text{maximize} & \quad - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i \\
\text{subject to} & \quad \sum \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0
\end{align*}
\]
Last time... **Large Margin Classifier**

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

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

\[ f(x) = \sum_i \alpha_i y_i (x_i^T x) + b \]

\[ \alpha_i > 0 \implies \text{support vectors} \]
Last time... **Soft-margin Classifier**

\[ \langle w, x \rangle + b \leq -1 \]

\[ \langle w, x \rangle + b \geq 1 \]

Theorem (Minsky & Papert)
Finding the minimum error separating hyperplane is NP hard

minimum error separator is impossible
Last time... Adding Slack Variables

\[ \xi_i \geq 0 \]

\[ \langle w, x \rangle + b \leq -1 + \xi \]

minimize amount of slack

Convex optimization problem
Last time... Adding Slack Variables

- for $0 < \xi \leq 1$ point is between the margin and correctly classified
- for $\xi_i \geq 0$ point is misclassified

\[ \langle w, x \rangle + b \leq -1 + \xi \]

\[ \langle w, x \rangle + b \geq 1 - \xi \]

Convex optimization problem

minimize amount of slack

adopted from Andrew Zisserman
Last time... Adding Slack Variables

• Hard margin problem

\[
\begin{align*}
&\underset{w,b}{\text{minimize}} \quad \frac{1}{2} \|w\|^2 \\
&\text{subject to} \quad y_i [\langle w, x_i \rangle + b] \geq 1
\end{align*}
\]

• With slack variables

\[
\begin{align*}
&\underset{w,b}{\text{minimize}} \quad \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \\
&\text{subject to} \quad y_i [\langle w, x_i \rangle + b] \geq 1 - \xi_i \text{ and } \xi_i \geq 0
\end{align*}
\]

Problem is always feasible. Proof:
\[
w = 0 \text{ and } b = 0 \text{ and } \xi_i = 1 \quad \text{(also yields upper bound)}
\]
Soft-margin classifier

• Optimisation problem:

$$\min_{w, b} \frac{1}{2} \|w\|^2 + C \sum_i \xi_i$$

subject to $$y_i \left[ \langle w, x_i \rangle + b \right] \geq 1 - \xi_i$$ and $$\xi_i \geq 0$$

C is a **regularization** parameter:

• small $C$ allows constraints to be easily ignored → **large margin**

• large $C$ makes constraints hard to ignore → **narrow margin**

• $C = \infty$ enforces all constraints: **hard margin**

adopted from Andrew Zisserman
Demo time...
This week

• Multi-class classification
• Introduction to kernels
Multi-class classification
Multi-class classification
Multi-class classification
One versus all classification

• Learn 3 classifiers:
  - - vs. \{o,+,\}, weights \(w_-\)
  - + vs. \{o,-\}, weights \(w_+\)
  - o vs. \{+,+\}, weights \(w_o\)

• Predict label using:
  \[ \hat{y} \leftarrow \arg\max_k w_k \cdot x + b_k \]

• Any problems?

• Could we learn this dataset?
Multi-class SVM

- Simultaneously learn 3 sets of weights:
  - How do we guarantee the correct labels?
  - Need new constraints!

The “score” of the correct class must be better than the “score” of wrong classes:

\[ w(y_j) \cdot x_j + b(y_j) > w(y) \cdot x_j + b(y) \quad \forall j, y \neq y_j \]
Multi-class SVM

- As for the SVM, we introduce slack variables and maximize margin:

\[
\begin{align*}
\text{minimize}_{w, b} & \quad \sum_y w(y) \cdot w(y) + C \sum_j \xi_j \\
\text{subject to} & \quad w(y_j) \cdot x_j + b(y_j) \geq w(y') \cdot x_j + b(y') + 1 - \xi_j, \quad \forall y' \neq y_j, \forall j \\
& \quad \xi_j \geq 0, \forall j
\end{align*}
\]

To predict, we use:

\[
\hat{y} \leftarrow \arg \max_k w_k \cdot x + b_k
\]

Now can we learn it?
Kernels
Non-linear features

- Regression
  We got nonlinear functions by preprocessing
- Perceptron
  - Map data into feature space \( x \rightarrow \phi(x) \)
  - Solve problem in this space
  - Query replace \( \langle x, x' \rangle \) by \( \langle \phi(x), \phi(x') \rangle \) for code
- Feature Perceptron
  - Solution in span of \( \phi(x_i) \)
Non-linear features

- Separating surfaces are Circles, hyperbolae, parabolae
Solving XOR

- XOR not linearly separable
- Mapping into 3 dimensions makes it easily solvable
SVM with a polynomial Kernel visualization

Created by: Udi Aharoni
Quadratic Features in $\mathbb{R}^2$

$$\Phi(x) := \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$$

Dot Product

$$\left\langle \Phi(x), \Phi(x') \right\rangle = \left\langle \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right), \left(x'_1^2, \sqrt{2}x'_1x'_2, x'_2^2\right) \right\rangle$$

$$= \langle x, x' \rangle^2.$$  

Insight

Trick works for any polynomials of order via $\langle x, x' \rangle^d$. 

![Heatmaps demonstrating quadratic features](image)
Computational Efficiency

Problem
- Extracting features can sometimes be very costly.
- Example: second order features in 1000 dimensions. This leads to $5 \cdot 10^5$ numbers. For higher order polynomial features much worse.

Solution
Don’t compute the features, try to compute dot products implicitly. For some features this works . . .

Definition
A kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a symmetric function in its arguments for which the following property holds

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

for some feature map $\Phi$. If $k(x, x')$ is much cheaper to compute than $\Phi(x)$ . . .
Recap: The Perceptron

initialize $w = 0$ and $b = 0$
repeat
  if $y_i [\langle w, x_i \rangle + b] \leq 0$ then
    $w \leftarrow w + y_i x_i$ and $b \leftarrow b + y_i$
  end if
until all classified correctly

- Nothing happens if classified correctly
- Weight vector is linear combination $w = \sum_{i \in I} y_i x_i$
- Classifier is linear combination of inner products $f(x) = \sum_{i \in I} y_i \langle x_i, x \rangle + b$
Recap: The Perceptron on features

initialize $w, b = 0$
repeat
  Pick $(x_i, y_i)$ from data
  if $y_i(w \cdot \Phi(x_i) + b) \leq 0$ then
    $w' = w + y_i\Phi(x_i)$
    $b' = b + y_i$
  until $y_i(w \cdot \Phi(x_i) + b) > 0$ for all $i$

• Nothing happens if classified correctly
• Weight vector is linear combination
  $w = \sum_{i \in I} y_i \phi(x_i)$
• Classifier is linear combination of inner products
  $f(x) = \sum_{i \in I} y_i \langle \phi(x_i), \phi(x) \rangle + b$
The Kernel Perceptron

initialize $f = 0$

repeat

Pick $(x_i, y_i)$ from data

if $y_i f(x_i) \leq 0$ then

$$f(\cdot) \leftarrow f(\cdot) + y_i k(x_i, \cdot) + y_i$$

until $y_i f(x_i) > 0$ for all $i$

• Nothing happens if classified correctly
• Weight vector is linear combination $w = \sum_{i \in I} y_i \phi(x_i)$
• Classifier is linear combination of inner products

$$f(x) = \sum_{i \in I} y_i \langle \phi(x_i), \phi(x) \rangle + b = \sum_{i \in I} y_i k(x_i, x) + b$$
Processing Pipeline

- Original data
- Data in feature space (implicit)
- Solve in feature space using kernels
Polynomial Kernels

Idea

- We want to extend \( k(x, x') = \langle x, x' \rangle^2 \) to

\[
k(x, x') = (\langle x, x' \rangle + c)^d \quad \text{where } c > 0 \text{ and } d \in \mathbb{N}.
\]

- Prove that such a kernel corresponds to a dot product.

Proof strategy

Simple and straightforward: compute the explicit sum given by the kernel, i.e.

\[
k(x, x') = (\langle x, x' \rangle + c)^d = \sum_{i=0}^{m} \binom{d}{i} (\langle x, x' \rangle)^i c^{d-i}
\]

Individual terms \( (\langle x, x' \rangle)^i \) are dot products for some \( \Phi_i(x) \).
Kernel Conditions

**Computability**
We have to be able to compute \( k(x, x') \) efficiently (much cheaper than dot products themselves).

**“Nice and Useful” Functions**
The features themselves have to be useful for the learning problem at hand. Quite often this means smooth functions.

**Symmetry**
Obviously \( k(x, x') = k(x', x) \) due to the symmetry of the dot product \( \langle \Phi(x), \Phi(x') \rangle = \langle \Phi(x'), \Phi(x) \rangle \).

**Dot Product in Feature Space**
Is there always a \( \Phi \) such that \( k \) really is a dot product?
Mercer’s Theorem

The Theorem
For any symmetric function \( k : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \) which is square integrable in \( \mathcal{X} \times \mathcal{X} \) and which satisfies
\[
\int_{\mathcal{X} \times \mathcal{X}} k(x, x') f(x) f(x') \, dx \, dx' \geq 0 \quad \text{for all} \quad f \in L_2(\mathcal{X})
\]
there exist \( \phi_i : \mathcal{X} \to \mathbb{R} \) and numbers \( \lambda_i \geq 0 \) where
\[
k(x, x') = \sum_i \lambda_i \phi_i(x) \phi_i(x') \quad \text{for all} \quad x, x' \in \mathcal{X}.
\]

Interpretation
Double integral is the continuous version of a vector-matrix-vector multiplication. For positive semidefinite matrices we have
\[
\sum \sum k(x_i, x_j) \alpha_i \alpha_j \geq 0
\]
Properties

Distance in Feature Space
Distance between points in feature space via
\[
d(x, x')^2 := \|\Phi(x) - \Phi(x')\|^2
\]
\[
= \langle \Phi(x), \Phi(x) \rangle - 2 \langle \Phi(x), \Phi(x') \rangle + \langle \Phi(x'), \Phi(x') \rangle
\]
\[
=k(x, x) + k(x', x') - 2k(x, x)
\]

Kernel Matrix
To compare observations we compute dot products, so we study the matrix $K$ given by
\[
K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j)
\]
where $x_i$ are the training patterns.

Similarity Measure
The entries $K_{ij}$ tell us the overlap between $\Phi(x_i)$ and $\Phi(x_j)$, so $k(x_i, x_j)$ is a similarity measure.
**Properties**

\( K \) is Positive Semidefinite

**Claim:** \( \alpha^\top K \alpha \geq 0 \) for all \( \alpha \in \mathbb{R}^m \) and all kernel matrices \( K \in \mathbb{R}^{m \times m} \). Proof:

\[
\sum_{i,j} \alpha_i \alpha_j K_{ij} = \sum_{i,j} \alpha_i \alpha_j \langle \Phi(x_i), \Phi(x_j) \rangle \\
= \left\langle \sum_{i=1}^m \alpha_i \Phi(x_i), \sum_{j=1}^m \alpha_j \Phi(x_j) \right\rangle = \left\| \sum_{i=1}^m \alpha_i \Phi(x_i) \right\|^2
\]

**Kernel Expansion**

If \( w \) is given by a linear combination of \( \Phi(x_i) \) we get

\[
\langle w, \Phi(x) \rangle = \left\langle \sum_{i=1}^m \alpha_i \Phi(x_i), \Phi(x) \right\rangle = \sum_{i=1}^m \alpha_i k(x_i, x).
\]
### Examples of kernels $k(x, x')$

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$\langle x, x' \rangle$</td>
</tr>
<tr>
<td>Laplacian RBF</td>
<td>$\exp(-\lambda |x - x'|)$</td>
</tr>
<tr>
<td>Gaussian RBF</td>
<td>$\exp(-\lambda |x - x'|^2)$</td>
</tr>
<tr>
<td>Polynomial</td>
<td>$\left(\langle x, x' \rangle + c \right)^d, c \geq 0, d \in \mathbb{N}$</td>
</tr>
<tr>
<td>B-Spline</td>
<td>$B_{2n+1}(x - x')$</td>
</tr>
<tr>
<td>Cond. Expectation</td>
<td>$\mathbf{E}_c[p(x</td>
</tr>
</tbody>
</table>

### Simple trick for checking Mercer’s condition

Compute the Fourier transform of the kernel and check that it is nonnegative.
Linear Kernel

$k(x,y)$ for $x=1$
Laplacian Kernel

\[ k(x,y) \text{ for } y=1 \]
Gaussian Kernel
Polynomial of order 3
$B_3$ Spline Kernel

\[ k(x,y) \text{ for } y=1 \]
Kernels in Computer Vision

- Features $x = \text{histogram (of color, texture, etc)}$

- Common Kernels
  - Intersection Kernel
  - Chi-square Kernel

\[
K_{\text{intersect}}(u, v) = \sum_i \min(u_i, v_i)
\]

\[
K_{\chi^2}(u, v) = \sum_i \frac{2u_i v_i}{u_i + v_i}
\]
$K_{\text{linear}}(x,y) = xy$

$K_{\text{min}}(x,y) = \min(x,y)$

$K_{\chi^2} = \frac{2xy}{(x+y)}$

Image credit: Subhransu Maji
The Kernel Trick for SVMs
The Kernel Trick for SVMs

- Linear soft margin problem

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_i \xi_i$$

subject to $y_i [\langle w, x_i \rangle + b] \geq 1 - \xi_i$ and $\xi_i \geq 0$

- Dual problem

$$\max_{\alpha} -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_i \alpha_i$$

subject to $\sum_i \alpha_i y_i = 0$ and $\alpha_i \in [0, C]$

- Support vector expansion

$$f(x) = \sum_i \alpha_i y_i \langle x_i, x \rangle + b$$
The Kernel Trick for SVMs

• Linear soft margin problem

$$\text{minimize}_{w, b} \frac{1}{2} \|w\|^2 + C \sum_i \xi_i$$

subject to $$y_i \left[\langle w, \phi(x_i) \rangle + b \right] \geq 1 - \xi_i$$ and $$\xi_i \geq 0$$

• Dual problem

$$\text{maximize}_\alpha - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) + \sum_i \alpha_i$$

subject to $$\sum_i \alpha_i y_i = 0$$ and $$\alpha_i \in [0, C]$$

• Support vector expansion

$$f(x) = \sum_i \alpha_i y_i k(x_i, x) + b$$
C=1
C = 1

The diagram illustrates a classification problem with support vectors. The data points are divided into two classes: y = 1 and y = -1. The support vectors are indicated by green arrows connecting the decision boundary (y = 0) to the data points. The boundary is determined by the maximum margin principle, with the goal of maximizing the distance between the decision boundary and the closest data points from each class.
C = 20
$C=5$
C=10
$C=2$
And now with a narrower kernel
And now with a very wide kernel
Nonlinear Separation

- Increasing $C$ allows for more nonlinearities
- Decreases number of errors
- SV boundary need not be contiguous
- Kernel width adjusts function class
Overfitting?

• Huge feature space with kernels: should we worry about overfitting?

• SVM objective seeks a solution with large margin
  - Theory says that large margin leads to good generalization (we will see this in a couple of lectures)

• But everything overfits sometimes!!!

• Can control by:
  - Setting C
  - Choosing a better Kernel
  - Varying parameters of the Kernel (width of Gaussian, etc.)