Lecture 17:

- Multi-class SVMs
Administrative

- We will have a **make-up lecture** on Saturday April 23, 2016.

- **Project progress reports** are due April 21, 2016
  - 2 days left!
Recap: Support Vector Machines

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

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

linear function

\[ f(x) = \langle w, x \rangle + b \]
Recap: Support Vector Machines

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

optimization problem

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

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

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

optimization problem

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

\[
\minimize_{w, b} \frac{1}{2} \|w\|^2 \quad \text{subject to} \quad y_i [\langle x_i, w \rangle + b] \geq 1
\]

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

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

\[
\text{subject to } \sum \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0
\]
Recap: Large Margin Classifier

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

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

\[ f(x) = \sum_{i} \alpha_i y_i (x_i^T x) + b \]

\[ \alpha_i > 0 \implies \text{support vectors} \]
Recap: Soft-margin Classifier

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

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

\[ \langle w, x \rangle + b \geq 1 \]
Recap: Adding Slack Variables

\[ \xi_i \geq 0 \]

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

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

Convex optimization problem

minimize amount of slack
Recap: 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
Adding Slack Variables

- Hard margin problem
  \[
  \min_{w,b} \frac{1}{2} \|w\|^2 \quad \text{subject to } y_i [\langle w, x_i \rangle + b] \geq 1
  \]

- With slack variables
  \[
  \min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \\
  \text{subject to } y_i [\langle w, x_i \rangle + b] \geq 1 - \xi_i \text{ and } \xi_i \geq 0
  \]

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

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

\[C\] is a **regularization** parameter:

- small \(C\) allows constraints to be easily ignored
  \(\rightarrow\) large margin
- large \(C\) makes constraints hard to ignore
  \(\rightarrow\) 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

- Real-world problems often have multiple classes: text, speech, image, biological sequences.
- Algorithms studied so far: designed for binary classification problems.
- How do we design multi-class classification algorithms?
  - can the algorithms used for binary classification be generalized to multi-class classification?
  - can we reduce multi-class classification to binary 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:

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

To predict, we use:
\[
\hat{y} \leftarrow \arg \max_k \ w_k \cdot x + b_k
\]

Now can we learn it?
Kernels
Solving XOR

- XOR not linearly separable
- Mapping into 3 dimensions makes it easily solvable
Quadratic Features in $\mathbb{R}^2$

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

**Dot Product**

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

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

**Insight**

Trick works for any polynomials of order via $\langle x, x' \rangle^d$. 
SVM with a polynomial Kernel visualization

Created by:
Udi Aharoni
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

\[
\begin{align*}
w' &= w + y_i \Phi(x_i) \\
b' &= b + y_i
\end{align*}
\]

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

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