SVM Material

- SVM material in books for this class:
  - Brief discussion in Duda, Hort & Stork, pg 262-264.
    - Read Problems 29-33, pg 275-277.
  - Not mentioned in Devroye or Mitchell.
  - Hastie, Tibshirani & Friedman, Section 4.5 and Chapter 12.

- Additional References:
  - Introductory chapters in
    - Scholkopf et al (eds) - "Advances in Kernel Methods"
    - Smola et al (eds) - "Advances in Large Margin Classifiers"
What an SVM does

Input:
- Training set \( \{(x_i, y_i)\} \) containing \( r \) labelled examples
  - \( x_i \in X \subseteq \mathbb{R}^d \), \( x_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \)
  - \( y_i = +1 \) or \(-1\)
- For more than 2 classes, use methods discussed before, e.g. binary classifier for each pair of classes, or each class vs all others, etc.

Output:
- A classifier given by \( \text{sign}(f(x)) \), where \( f \) is chosen to yield the "best" classifier in some sense.
Classifiers are hyperplanes separating positive from negative examples.

"Best" hyperplane is the one with maximum *MARGIN*, i.e. maximum DISTANCE FROM THE CLOSEST EXAMPLE.

Solving the lin-sep case will allow the non-lin-sep case to be solved "easily".
Why Max Margin?

- Intuition: Classification is less sensitive to exact location of training point - Lower Variance
- Theory: Generalization error of hyperplane can be bounded (probabilistically) by an expression depending on $1/\text{margin}^2$
- Theorem
  - Let:
    - $D$ be a distribution on $X \times \{-1, 1\}$
    - $R$ be the radius of a ball containing the support of $D$
    - $r$ random examples be drawn from $D$
    - $h$ be a separating hyperplane with margin $> \gamma$
    - $\text{err}(h) = \Pr_D(h(x) \neq y)$
  - Then, for any $\delta > 0$,
    - If $r$ is "sufficiently large"
      - depending on $R$ and $\gamma$, but not on $d=\text{dimension of } X$
      - $\Pr\{\text{err}(h) < O((1/r)((R^2/\gamma^2)+\log(1/\delta )) \} > 1-\delta$
Hyperplanes

- Points $x$ on hyperplane $h$ satisfy $w \cdot x + b = 0$
  - $w$ is normal to the hyperplane
  - $w \cdot x = \sum_{i=1}^{d} w_i x_i$
- Distance of $x_i$ from $h$ is $\text{dist} = y_i(w \cdot x_i + b) / \|w\|$
  - because $x_i$ satisfies $w \cdot x + b = ? = |\text{dist}| / \|w\|$
  - $\|b\| / \|w\| = $ distance of $h$ from origin

$w \cdot x + b = ? \quad w \cdot x + b = 1$

$w$

$\text{dist}$

$x_i$
Max Margin Hyperplane

- Margin of $h = \text{distance to closest example}$
  
  \[ = \min_{i} y_i(w \cdot x_i + b) / \|w\| \]

- Max Margin hyperplane:
  
  \[ \max_{w,b} \min_{i} y_i(w \cdot x_i + b) / \|w\| \]

Approach 1: Fix denominator, maximize numerator:

- \[ \max_{w,b} \min_{i} y_i(w \cdot x_i + b) \text{ such that } \|w\| = 1 \]
- Constrained max of complex nonlinear function - difficult

Approach 2: Fix numerator, MINIMIZE denominator:

- \[ \min_{w,b} \|w\| \text{ such that } \min_{i} y_i(w \cdot x_i + b) = 1 \]
  
  - Equivalent to:
  
  \[ \min_{w,b} \|w\|^2 = w \cdot w = \sum_{i} d w_i^2 \text{ such that } y_i(w \cdot x_i + b) \geq 1 \ \forall i \]
  
  - Quadratic optimization with linear constraints
Examples

\[ w = x_1 - x_2 \]

\[ w = 0.5x_1 + 0.5x_3 - x_2 \]

\[ w = 0.9x_1 + 0.1x_3 - x_2 = 0.2x_1 + 0.8x_3 - x_4 \]
The solution to \( \min_{w,b} \| w \|^2 = w \cdot w = \sum_{i=1}^{n} w_i^2 \) such that \( y_i(w \cdot x_i + b) \geq 1 \) occurs at \( w = \sum a_i y_i x_i \)

- \( a_i \geq 0 \)
- \( \sum a_i y_i = 0 \) i.e. \( \sum_{+ve} a_i = \sum_{-ve} a_i \)
- \( a_i[y_i(w \cdot x_i + b) - 1] = 0 \) (Karush-Kuhn-Tucker conditions)
- \( a_i > 0 \Rightarrow y_i(w \cdot x_i + b) = 1 \) i.e. \( a_i = 0 \) for inactive constraints
  - \( x_i \) is a "support vector" MEANS \( a_i > 0 \)
  - All support vectors are "on the margin"
  - CONVERSE IS FALSE
- \( b \) can be recovered from any active constraint \( y_i(w \cdot x_i + b) = 1 \)

The solution is (usually) **Sparse** - number of support vectors is small

Why is the solution of this form?

- Perceptron
- Convex Hull
- Lagrangian (primal/dual)
Perceptron

- Finds hyperplane for linearly separable data:
  - $w=0$
  - Repeat
    - for each training point $(x_i, y_i)$
      - if $x_i$ is incorrectly classified do $w = w + y_i x_i$

- Converges to SOME separating hyperplane
  - not max margin

- Maintains $w$ of the form $w = \sum_i a_i y_i x_i$
  - $a_i$ reflects how often a point was updated - its 'difficulty'

- Can be made to converge to max margin hyperplane
  - pick worst-classified point at each iteration
  - Computationally too expensive
Convex hull of $Z = \{\Sigma t_i z_i | z_i \text{ in } Z, \ 0 \leq t_i \leq 1, \Sigma t_i = 1\}$

Normal vector of max margin hyperplane joins closest pair of points in Convex hull of positive and negative training points

$$w = x^+ - x^- = \Sigma_{+ve} s_i x_i - \Sigma_{-ve} t_i x_i,$$
where $\Sigma s_i = \Sigma t_i = 1$

$$= \Sigma_1^r a_i y_i x_i, \ \text{and } \Sigma_{+ve} a_i = \Sigma_{-ve} a_i, \ \text{i.e. } \Sigma_1^r a_i y_i = 0$$
Lagrangian (primal/dual)

"Primal" Problem: Min\(_{w,b} \ w \cdot w\) such that \(y_i(w \cdot x_i + b) \geq 1\)

\(L(w,b,a) = \frac{1}{2}(w \cdot w) - \sum \alpha_i [y_i(w \cdot x_i + b) - 1], \ \alpha_i \geq 0\)

- Min \(L\) as a function of \(w,b\)
- Max \(L\) as a function of \(\alpha\)

- Constraints satisfied \(\Rightarrow L \leq \frac{1}{2}(w \cdot w)\)

- \(\frac{\partial L}{\partial w} = w - \sum \alpha_i y_i x_i = 0\) when \(w = \sum \alpha_i y_i x_i\)

- \(\frac{\partial L}{\partial b} = \sum \alpha_i y_i = 0\)

Substitute into \(L\):

- \(L(\alpha) = \frac{1}{2}(\sum \alpha_i y_i x_i) \cdot (\sum \alpha_i y_j x_j) - \sum \alpha_i [y_i(\sum \alpha_j y_j x_j) \cdot x_i + b] - 1\)

- \(= \frac{1}{2} \sum \alpha_i y_i y_j (x_i \cdot x_j) - \sum \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum \alpha_i\)

- \(= \sum \alpha_i - \frac{1}{2} \sum \alpha_i \sum \alpha_j y_i y_j (x_i \cdot x_j)\)

"Dual" Problem:

Max\(_{\alpha} \sum \alpha_i - \frac{1}{2} \sum \alpha_i \sum \alpha_j y_i y_j (x_i \cdot x_j)\) such that \(\alpha_i \geq 0, \sum \alpha_i y_i = 0\)
Data dot products only!

- Dual Problem is usually easier to solve
  - the constraints $a_i \geq 0, \sum_i a_i y_i = 0$ are simpler
  - Usually solved iteratively:
    - start with constraints satisfied
    - increase objective function while maintaining constraints
- Note that in $\sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j (x_i \cdot x_j)$
- THE TRAINING DATA ONLY APPEAR AS DOT PRODUCTS
- $w=\sum_i a_i y_i x_i \Rightarrow w \cdot w = \sum_i \sum_j a_i a_j y_i y_j (x_i \cdot x_j)$
  - The max margin hyperplane for linearly separable data is of the form
  - $h(x) = w \cdot x + b = (\sum_i a_i y_i x_i) \cdot x + b = \sum_i a_i y_i (x_i \cdot x) + b$
If training data is not linearly-separable:
- map into a space $F$ so that training data becomes linearly-separable
- find max margin hyperplane in $F$
- this gives (non-hyperplane) decision surface in $X$

Define $\phi:X \rightarrow \mathbb{R}^2$ by $\phi(x) = \phi((x_1, x_2)) = (x_1^2, x_2^2)$.
Hyperplane in $\phi(X)$ is $ax_1^2 + bx_2^2 + c = 0$
Max margin hyperplane in $\phi(X)$ gives separating ellipse in $X$. 
Different choices for φ correspond to different families of decision surfaces in the original space X
  - Such a φ can always be found (homework)
  - F can be very high-dimensional
  - φ need not be continuous, 1-1 ...

Surface in X that corresponds to the max margin hyperplane in F

Surface that would be obtained by "maximizing the margin" in X.

The family being searched in X is changed by just changing the mapping φ
  - However in practice explicitly computing φ is difficult
Using $\phi$ Implicitly

- **Lin-Sep:**
  - Max $\alpha \sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j (x_i \cdot x_j)$ such that $a_i \geq 0$, $\sum a_i y_i = 0$

- **Non-Lin-Sep:**
  - Find $\phi : X \rightarrow F$
    - so that $\{(\phi(x_i), y_i)\}$ is linearly separable:
      - Max $\alpha \sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j (\phi(x_i) \cdot \phi(x_j))$ such that $a_i \geq 0$, $\sum a_i y_i = 0$
  - Suppose $K$ is a "kernel" function,
    - i.e. $K(x, x') = \phi(x) \cdot \phi(x')$ for some $\phi$

- Then the max margin hyperplane in $F$ is found by:
  - Max $\alpha \sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j K(x_i, x_j)$ such that $a_i \geq 0$, $\sum a_i y_i = 0$

- The resulting decision surface is of the form
  - $f(x) = \sum a_i y_i K(x_i, x) + b$

- Compare with max margin hyperplane:
  - $h(x) = \sum a_i y_i (x_i \cdot x) + b$
SVM: Main Ideas

- Max margin
  - min $\|w\|$  
    - constrained optimization
- Lin-sep case:
  - solve equivalent "dual" problem (1950s)
  - training data only appear as dot products
- General case:
  - map into high-dim space
  - replace dot products by kernel values (Aizerman, 1964)
- These ideas all existed independently before SVMs
- Putting them together
What an SVM does

- **Input:**
  - Training set \( \{(x_i, y_i)\}_{i=1}^{r} \)
    - \( x_i \in X \subseteq \mathbb{R}^d \)
    - \( y_i = +1 \) or \(-1\)
  - Kernel function \( K: X \times X \rightarrow \mathbb{R} \)

- **Output:**
  - A classifier given by \( \text{sign}(f(x)) \)
    - \( f \) is of the form \( f(x) = \sum_i a_i y_i K(x_i, x) + b, \ a_i \geq 0 \) for all \( i \)
    - \( f \) corresponds to the max margin hyperplane in the space implicitly defined by \( K \)
  - \( a_i \) are computed
    - by solving \( \max_{a_i} \sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j K(x_i, x_j) \) such that \( a_i \geq 0, \sum_i a_i y_i = 0 \)

- **How do we pick \( K \)?**
- **How do we solve the constrained optimization?**
"Kernel" has many meanings/uses:
- Linear maps
- Integral Operators
- Operating Systems
- ...

"Kernel" of a nut
- core, seed
- central/essential part
- base on which everything else is built

If you know what happens in the kernel, you know "everything"
Polynomial Kernels

- How to find $K$ such that $K(x,x') = \phi(x)\cdot \phi(x')$ for some $\phi$?

- Examples:
  - $K(x,x') = x \cdot x'$ - original data is linearly separable
  - $K(x,x') = (x \cdot x')^2 = (x_1x'_1 + x_2x'_2)^2$
    \[= x_1^2x'_1^2 + 2x_1x_2x'_1x'_2 + x_2^2x'_2^2\]
    \[= \phi(x)\cdot \phi(x')?\]
    - $\phi(x) = \phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$ works
    - $\phi(x) = \phi(x_1, x_2) = (x_1^2, x_1x_2, x_1x_2, x_2^2)$ also works
  - $K(x,x') = ((x \cdot x') + 1)^2$
    \[= (x \cdot x')^2 + 2(x \cdot x') + 1\]
    - $\phi(x) = \phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$

- $K(x,x') = (x \cdot x')^k$ corresponds to using all terms of degree $k$
- $K(x,x') = ((x \cdot x') + 1)^k$ corresponds to using all terms of degree $\leq k$
  i.e. polynomials of degree $k$. 
Radial Basis Functions

- $K(x,x') = \exp(-\|(x-x')^2\|/c)$
  - Place Gaussian at certain points
  - Classifier is linear combination of Gaussians
    - $f(x) = \sum_{i}^{m} a_i K(x_i,x) + b$
  - Neural network with Gaussians at the hidden layer

- SVM automatically finds
  - number and location of points $x_i$ (support vectors)
  - weights $a_i$

- $\exp(-\|(x-x')^2\|/c)$ is an exponential kernel
  - How do we know it is a valid kernel?
    - rather than try find $\phi$
    - use theory to build kernels from simpler kernels.
Characterisation of Kernels

- **Proposition:**
  - If $X$ is finite, $K:X \times X \rightarrow \mathbb{R}$ is a kernel if and only if
    - $K$ is symmetric
    - $K(x_i,x_j)_{1 \times 1}^n$ is positive semi-definite
      - $z^TKz \geq 0$ for all $z$
      - all eigenvalues of $K \geq 0$

- **Proof:**
  - Suppose $K$ is symmetric and positive semi-definite
  - Write $K = V^{-1}DV$
    - $D = \text{diag}(\lambda_i)$, where $\lambda_i \geq 0$
    - $V$ orthogonal, $v_i$ is the $t^{th}$ column of $V$.
  - Define $\phi:X \rightarrow \mathbb{R}^n$ by $\phi(x) = (\sqrt{\lambda_i}v_i)^n$
  - Then $\phi(x_i) \cdot \phi(x_j) = \Sigma^n_{i=1} \lambda_i v_i v_j$
    - $= (V^{-1}DV)_{ij}$
    - $= K(x_i,x_j)$
  - Conversely, if $K$ is a kernel with a negative eigenvalue $\lambda_s$ and corresponding eigenvector $v_s$, then $z = \Sigma^n_{i=1} v_s \phi(x_i)$ has norm $\lambda_s < 0$
Constructing Kernels

- Mercer's Theorem:
  - $K$ is a kernel if and only if
    - $K$ is symmetric
    - $K(x_i, x_j)_1^n$ is positive semi-definite for every finite subset of $X$.

- Use this to prove that
  - sums of kernels are kernels
  - positive scalar products of kernels are kernels
  - .... (homework)
  - a polynomial with positive coefficients applied to a kernel gives a kernel
  - limits of kernels are kernels
  - ....
  - therefore $\exp(-\|x-x'\|^2/c)$ is a kernel
Solving the Optimization

- "Primal" problem:
\[
\text{Min}_a \sum_i \alpha_i y_i K(x_i, x_i) \text{ such that } y_i (\sum_i \alpha_i y_i K(x_i, x_i)) + b \geq 1
\]

- "Dual" problem:
\[
\text{Max}_a \sum_i \alpha_i - \frac{1}{2} \sum_i \alpha_i \sum_j \alpha_j y_i y_j K(x_i, x_j) \text{ such that } \alpha_i \geq 0, \sum_i \alpha_i y_i = 0
\]

- Iterative Methods are used
  - start with constraints satisfied
  - increase dual objective function while maintaining constraints

- No local optima

- Terminate when:
  - objective function stops increasing - unreliable
  - KKT conditions satisfied

- Running time usually \(~ O(dr^2)\)
  - Size of \(K(x_i, x_j)\) is \(~ O(r^2)\)
    - do not want \(K\) sparse
  - problem may need to be decomposed into "chunks"
Soft Margins

- May not want + and -points completely separated
  - noisy data
  - avoid overfitting
- Allow hypothesis to make some errors on the training set in order to avoid more complex hypothesese.
Soft Margins: 1-Norm

Minimize $\xi, w, b$ such that $y_i (w \cdot x_i + b) \geq 1 - \xi_i$, $\xi_i \geq 0$

- $\xi_i$ are "slack variables"
  - $x_i$ is misclassified $\iff \xi_i > 1$
- $C$ modulates the trade-off between:
  - simplicity of the decision surface
  - number of misclassified training points.
  - regularization
- Good value of $C$ determined empirically, e.g. by cross-validation

Dual problem:

- Maximize $\sum a_i - \frac{1}{2}\sum a_i a_j y_i y_j K(x_i, x_j)$ such that $C \geq a_i \geq 0$, $\sum a_i y_i = 0$
- "Box" constraint on the $a_i$
- $\xi_i > 0 \implies a_i = C$
Soft Margins: 2-Norm

- Min$_{\xi,w,b}$ $w \cdot w + C\sum_1 \xi_i^2$ such that $y_i(w \cdot x_i + b) \geq 1 - \xi_i$
  - $\xi_i \geq 0$ constraint not needed
  - Role of C as before
- Dual Problem:
  - Max$_a \sum_1 a_i - \frac{1}{2} \sum_1 \sum_1 a_i a_j y_i y_j (K(x_i, x_j) + (1/C)\delta_{ij})$
    - such that $a_i \geq 0, \sum_1 a_i y_i = 0$
    - $\delta_{ij} = 1$ if $i = j$, 0 otherwise
    - Change of kernel
      - add 1/C to all diagonal elements
SVM Resources

- **Downloadable Software:**
  - svmlight
    - C code available at http://ais.gmd.de/~thorsten/svm_light/
  - weka (Waikato Environment for Knowledge Analysis)
    - Java code available at http://www.cs.waikato.ac.nz/~ml/weka/
  - SVMTorch
    - SVM for regression problems
  - ....

- **Applet:** http://svm.research.bell-labs.com/
- **General:** http://www.kernel-machines.org/
- **History:** http://www.kyb.tuebingen.mpg.de/bu/people/bs/svm.html
- **Applications:** http://www.clopinet.com/isabelle/Projects/SVM/applist.html
SVMs: Pros and Cons

- **Kernel Function**
  - No other parameter-fiddling needed
  - Allows incorporation of prior knowledge
  - How to choose?

- **Classification Accuracy** usually good

- **Convergence**
  - No local minima
  - Often slow in practice

- **Theoretical Foundations**
  - Structured research framework
  - Practical applications are much messier

- **Sparseness**
  - Only support vectors needed for solution
  - Many data points may be support vectors