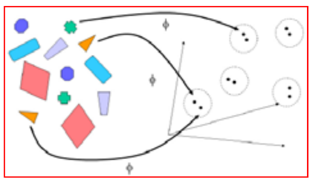
# Machine Learning

**10-701/15-781, Fall 2011**

## Advanced topics in Max-Margin Learning

Eric Xing



Lecture 20, November 21, 2011





# Recap: the SVM problem

- We solve the following constrained opt problem:

$$\max_\alpha \quad \mathcal{J}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

$$\text{s.t.} \quad \alpha_i \geq 0, \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0.$$

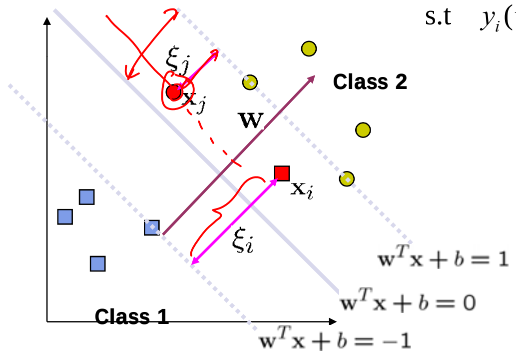
- This is a **quadratic programming** problem.
  - A global maximum of $\alpha_i$ can always be found.
  - The solution: $w = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = \sum_{i \in SV} \alpha_i y_i x_i$
  - How to predict: $\mathbf{w}^T \mathbf{x}_{\text{new}} + b \lessgtr 0$

# Non-linearly Separable Problems

s.t $\quad y_i(w^T x_i + b) \geq 1, \quad \forall i$

$-\xi_i$

$1-\xi_i$




- We allow "error" $\xi_i$ in classification; it is based on the output of the discriminant function $\boldsymbol{w}^T\boldsymbol{x}+b$
- $\xi_i$ approximates the number of misclassified samples



# Soft Margin Hyperplane

- Now we have a slightly different opt problem:

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

$$\text{s.t} \quad \begin{aligned} & y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \forall i \\ & \xi_i \geq 0, \quad \forall i \end{aligned}$$

  - $\xi_i$ are "slack variables" in optimization
  - Note that $\xi_i$=0 if there is no error for $\mathbf{x}_i$
  - $\xi_i$ is an upper bound of the number of errors
  - $C$ : tradeoff parameter between error and margin

## Lagrangian Duality, cont.

- Recall the Primal Problem:

$$\min_w \max_{\alpha,\beta,\alpha_i \geq 0} \mathcal{L}(w,\alpha,\beta)$$ $= f() + \alpha g + \beta h_i$

- The Dual Problem:

$$\max_{\alpha,\beta,\alpha_i \geq 0} \min_w \mathcal{L}(w,\alpha,\beta)$$

- **Theorem (weak duality):**

$$d^* = \max_{\alpha,\beta,\alpha_i \geq 0} \min_w \mathcal{L}(w,\alpha,\beta) \;\leq\; \min_w \max_{\alpha,\beta,\alpha_i \geq 0} \mathcal{L}(w,\alpha,\beta) = p^*$$

- **Theorem (strong duality):**

  Iff there exist a saddle point of $\mathcal{L}(w,\alpha,\beta)$, we have

$$d^* = p^*$$



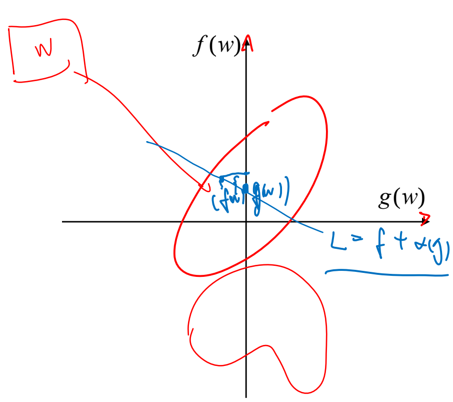
## A sketch of strong and weak duality

$h(w) = 0$

$L$

- Now, ignoring $h(x)$ for simplicity, let's look at what's happening graphically in the duality theorems.

$$d^* = \max_{\alpha_i \geq 0} \min_w f(w) + \alpha^T g(w) \;\leq\; \min_w \max_{\alpha_i \geq 0} f(w) + \alpha^T g(w) = p^*$$




$L(w, \alpha, \beta) = f(w) + \alpha g(w)$

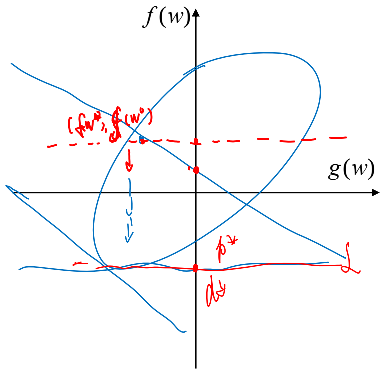
feasible of $L$

convex

## A sketch of strong and weak duality

- Now, ignoring $h(x)$ for simplicity, let's look at what's happening graphically in the duality theorems.

$$d^* = \max_{\alpha_i \geq 0} \min_w f(w) + \alpha^T g(w) \;\leq\; \min_w \max_{\alpha_i \geq 0} f(w) + \alpha^T g(w) = p^*$$




primal:
max$_\alpha$ first $\alpha \to 0$
then min$_w$ second go south until
on the border of space $w$.

dual:
min $w$
max $\alpha$
the solution to the dual is always
on the tangent of the feasible space
the slope of the tangent $\alpha$



## A sketch of strong and weak duality

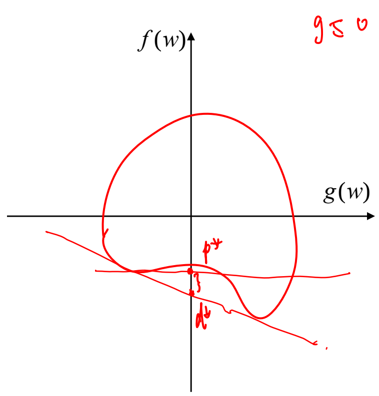
$h(w) = 0$

$L$

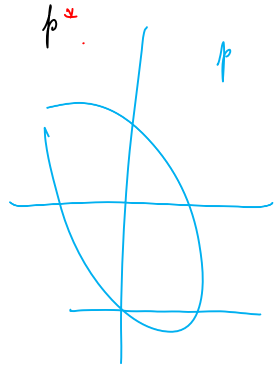
- Now, ignoring $h(x)$ for simplicity, let's look at what's happening graphically in the duality theorems.

$$d^* = \max_{\alpha_i \geq 0} \min_w f(w) + \alpha^T g(w) \;\leq\; \min_w \max_{\alpha_i \geq 0} f(w) + \alpha^T g(w) = p^*$$

$g \leq 0$

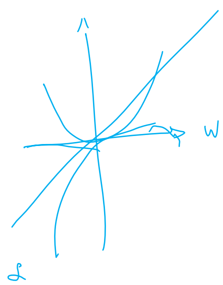



$p^*$.



## The KKT conditions

- If there exists some saddle point of $\mathcal{L}$, then the saddle point satisfies the following "Karush-Kuhn-Tucker" (KKT) conditions:




$$\frac{\partial}{\partial w_i}\mathcal{L}(w,\alpha,\beta)=0, \quad i=1,\ldots,k$$
$$\frac{\partial}{\partial \beta_i}\mathcal{L}(w,\alpha,\beta)=0, \quad i=1,\ldots,l$$
$$\alpha_i g_i(w)=0, \quad i=1,\ldots,m$$
$$g_i(w)\le 0, \quad i=1,\ldots,m$$
$$\alpha_i \ge 0, \quad i=1,\ldots,m$$

- **Theorem**: If $w^*$, $\alpha^*$ and $\beta^*$ satisfy the KKT condition, then it is also a solution to the primal and the dual problems.



## The Optimization Problem

- The dual of this new constrained optimization problem is

$$\max_\alpha \quad \mathcal{J}(\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i^T\mathbf{x}_j)$$
$$\text{s.t.} \quad 0\le\alpha_i\le C, \quad i=1,\ldots,m$$
$$\sum_{i=1}^{m}\alpha_i y_i=0.$$

- This is very similar to the optimization problem in the linear separable case, except that there is an upper bound $C$ on $\alpha_i$ now
- Once again, a QP solver can be used to find $\alpha_i$

# The SMO algorithm

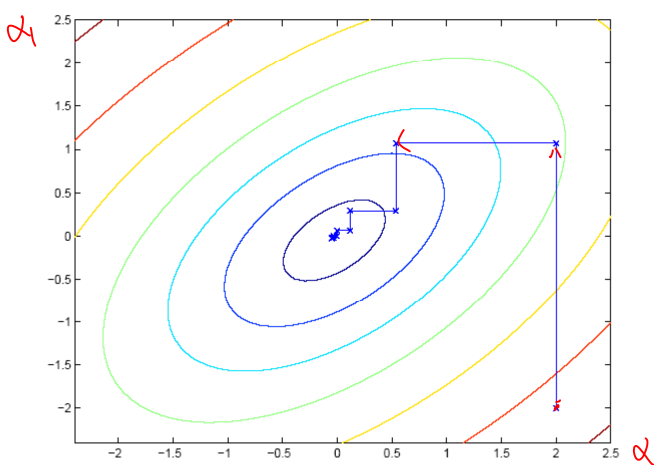
- Consider solving the unconstrained opt problem:

$$\max_{\alpha} W(\alpha_1, \alpha_2, \ldots, \alpha_m)$$

- We've already see three opt algorithms!
  - Coordinate ascent
  - Gradient ascent
  - Newton-Raphson
- Coordinate ascend:



# Coordinate ascend



## Sequential minimal optimization

- Constrained optimization:

$$\max_{\alpha} \quad \mathcal{J}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C, \quad i = 1, \dots, m$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0.$$

- Question: can we do coordinate along one direction at a time (i.e., hold all $\alpha_{[-i]}$ fixed, and update $\alpha_i$?)

$\alpha_1 \; ---- \; \alpha_n$

$\alpha_i \to \alpha_i + \Delta\alpha_i \; , \; \alpha_{-i} = \alpha_{-i}$

$\vec{\alpha}_{-j,-i} = \vec{\alpha}_{-j,-i}$

$\alpha_i, \; \alpha_j$

$\alpha_i = \alpha_i + \Delta\alpha_i$




## The SMO algorithm

Repeat till convergence

1. Select some pair $\alpha_i$ and $\alpha_j$ to update next (using a heuristic that tries to pick the two that will allow us to make the biggest progress towards the global maximum).

2. Re-optimize J($\alpha$) with respect to $\alpha_i$ and $\alpha_j$, while holding all the other $\alpha_k$ 's ($k \neq i; j$) fixed.

Will this procedure converge?

## Convergence of SMO

$$\max_\alpha \quad \mathcal{J}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

**KKT:**

$$\text{s.t.} \quad 0 \le \alpha_i \le C, \quad i = 1, \ldots, k$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0.$$

- Let's hold $\alpha_3, \ldots, \alpha_m$ fixed and reopt J w.r.t. $\alpha_1$ and $\alpha_2$



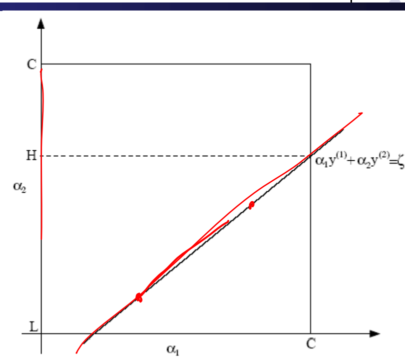

## Convergence of SMO

- The constraints:

$$\alpha_1 y_1 + \alpha_2 y_2 = \xi$$

$$0 \le \alpha_1 \le C$$

$$0 \le \alpha_2 \le C$$




- The objective:

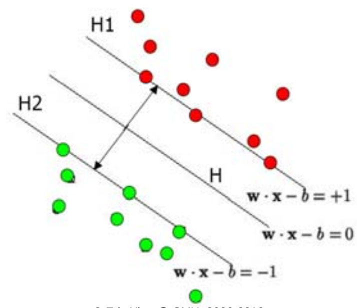
$$\mathcal{J}(\alpha_1, \alpha_2, \ldots, \alpha_m) = \mathcal{J}((\xi - \alpha_2 y_2) y_1, \alpha_2, \ldots, \alpha_m)$$

- Constrained opt:

## Cross-validation error of SVM

- The leave-one-out cross-validation error does not depend on the dimensionality of the feature space but only on the # of support vectors!

$$\text{Leave-one-out CV error} = \frac{\#\ \text{support vectors}}{\#\ \text{of training examples}}$$







## Advanced topics in Max-Margin Learning

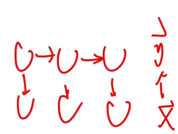
$$\max_\alpha \mathcal{J}(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j (\mathbf{x}_i^T\mathbf{x}_j)$$

$$\mathbf{w}^T\mathbf{x}_{\text{new}} + b \lessgtr 0$$

$\sum_i \alpha_i (x_i \cdot x_{new}) + b$ $\qquad w = \sum_{i \in SV} \alpha_i x_i$

$f(x_i, x_j) = \phi(x_i)\,\phi(x_j)$

- Kernel
- Point rule or average rule $\quad w^* \Rightarrow p^*(w)$
- Can we predict vec(y)? $\quad y = \{+1, -1\}$



## Outline

- The Kernel trick
- Maximum entropy discrimination
- Structured SVM, aka, Maximum Margin Markov Networks




## (1) Non-linear Decision Boundary

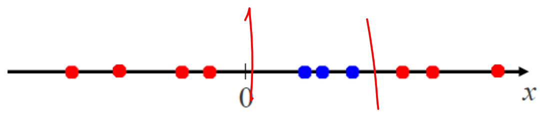
- So far, we have only considered large-margin classifier with a linear decision boundary
- How to generalize it to become nonlinear?
- Key idea: transform $\mathbf{x}_i$ to a higher dimensional space to "make life easier"
  - Input space: the space the point $\mathbf{x}_i$ are located
  - Feature space: the space of $\phi(\mathbf{x}_i)$ after transformation
- Why transform?

  $x \to f(x)$

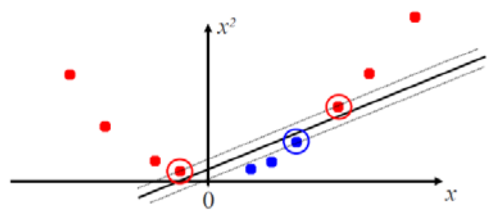
  - Linear operation in the feature space is equivalent to non-linear operation in input space
  - Classification can become easier with a proper transformation. In the XOR problem, for example, adding a new feature of $x_1x_2$ make the problem linearly separable (homework)

# The Kernel Trick

- Is this data linearly-separable?




- How about a quadratic mapping $\phi(x_i)$? $= x_i^2$






# The Kernel Trick

- Recall the SVM optimization problem

$$\max_{\alpha} \quad \mathcal{J}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C, \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0.$$

- The data points only appear as inner product
- As long as we can calculate the inner product in the feature space, we do not need the mapping explicitly
- Many common geometric operations (angles, distances) can be expressed by inner products

implicitly dealing with transf.

- Define the kernel function $K$ by $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$

# II. The Kernel Trick

- Computation depends on feature space
  - Bad if its dimension is much larger than input space

$$\max_\alpha \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t.} \quad \alpha_i \geq 0, \quad i = 1, \ldots, k$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0.$$

$K = x_i^T x_j$

$\Downarrow$

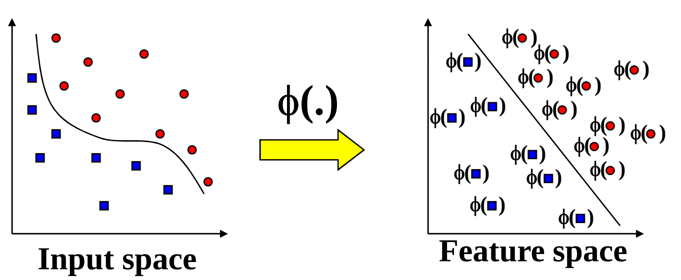
$K = \phi(x_i)\phi(x_j)$

**Where K(x$_i$,x$_j$) = $\phi$(x$_i$)$^t$ $\phi$(x$_j$)**

$$y^*(z) = \text{sign}\left(\sum_{i \in SV} \alpha_i y_i K(\mathbf{x}_i, z) + b\right)$$



# Transforming the Data




Note: feature space is of higher dimension than the input space in practice

- Computation in the feature space can be costly because it is high dimensional
  - The feature space is typically infinite-dimensional!
- The kernel trick comes to rescue

## An Example for feature mapping and kernels

- Consider an input $\mathbf{x}=[x_1,x_2]$
- Suppose $\phi(.)$ is given as follows

$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = 1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2$$

- An inner product in the feature space is

$$\left\langle \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right), \phi\left(\begin{bmatrix} x_1' \\ x_2' \end{bmatrix}\right) \right\rangle = 1 + 2x_1x_1' + 2x_2x_2' + x_1^2x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2x_2'^2$$
$$= (1 + x_1 x_2)^2$$

- So, if we define the **kernel function** as follows, there is no need to carry out $\phi(.)$ explicitly

$$K(\mathbf{x}, \mathbf{x}') = \left(1 + \mathbf{x}^T\mathbf{x}'\right)^2$$



## More examples of kernel functions

- Linear kernel (we've seen it)

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$$

- Polynomial kernel (we just saw an example)

$$K(\mathbf{x}, \mathbf{x}') = \left(1 + \mathbf{x}^T\mathbf{x}'\right)^p$$

$\phi' = (\quad)$
$K = O(m)$

where $p$ = 2, 3, … To get the feature vectors we concatenate all $p$th order polynomial terms of the components of x (weighted appropriately)

- Radial basis kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right) = \phi(x_i)^T \phi(x_j)$$

In this case the feature space consists of functions and results in a non-parametric classifier.

## The essence of kernel

- Feature mapping, but "without paying a cost"
  - E.g., polynomial kernel

$$K(x, z) = (x^T z + c)^d$$

  - How many dimensions we've got in the new space?
  - How many operations it takes to compute K()?

- Kernel design, any principle?
  - K(x,z) can be thought of as a similarity function between x and z
  - This intuition can be well reflected in the following "Gaussian" function (Similarly one can easily come up with other K() in the same spirit)

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$$

  - Is this necessarily lead to a "legal" kernel? (in the above particular case, K() is a legal one, do you know how many dimension $\phi(x)$ is?




## Kernel matrix

- Suppose for now that $K$ is indeed a valid kernel corresponding to some feature mapping $\phi$, then for $x_1, \ldots, x_m$, we can compute an $m \times m$ matrix $K = \{K_{i,j}\}$, where $K_{i,j} = \phi(x_i)^T \phi(x_j)$

- This is called a kernel matrix!

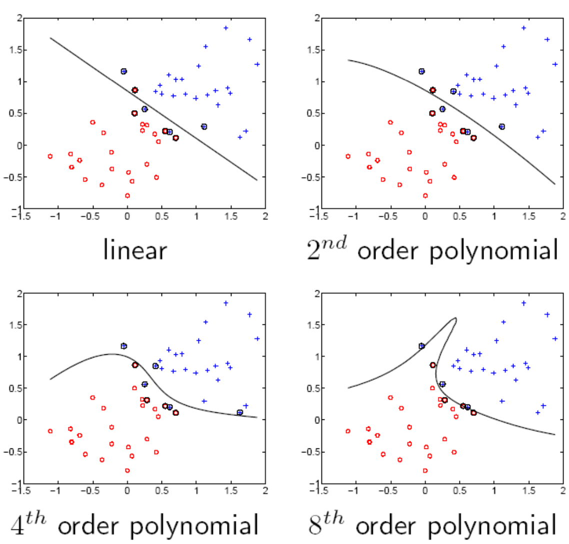
- Now, if a kernel function is indeed a valid kernel, and its elements are dot-product in the transformed feature space, it must satisfy:
  - Symmetry $K = K^T$
    proof $K_{i,j} = \phi(x_i)^T \phi(x_j) = \phi(x_j)^T \phi(x_i) = K_{j,i}$
  - Positive –semidefinite $y^T K y \geq 0 \quad \forall y$
    proof?

# Mercer kernel

**Theorem (Mercer):** Let $K: \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ be given. Then for $K$ to be a valid (Mercer) kernel, it is necessary and sufficient that for any $\{x_i, \ldots, x_m\}$, $(m < \infty)$, the corresponding kernel matrix is symmetric positive semi-denite.



# SVM examples




$x_i\, x_j$

$\downarrow$

$\underline{K}(x_i, x_j)$