# SVM and Review

Thursday Oct 11

# SVM

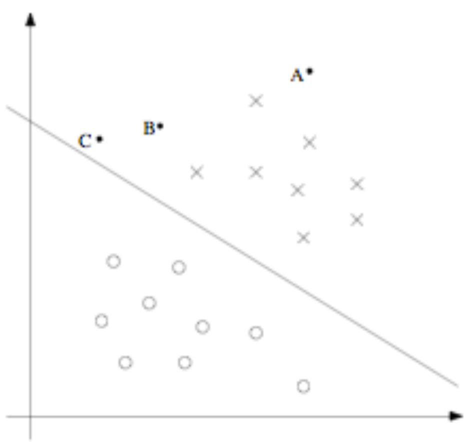
- Optimal margin classifier
  - Separates data with large “gap”
- Lagrange duality
- Kernels
- Non linearly separable case
- SMO algorithm
- Review

# Optimal Margin Classifier

- Recall: Logistic regression

$$p(y=1 \mid x; w) = \frac{1}{1+\exp(-w^T x)} = g(w^T x)$$

- Predict y = 1 if p(y=1|x;w) >= 0.5 (or $w^T x >= 0$)
- More confident that y = 1 if $w^T x >> 0$
- Similarly




More confident on our prediction of A than our prediction of C

# Optimal Margin Classifier

- Define a classifier $h_{w,b}(x)$:

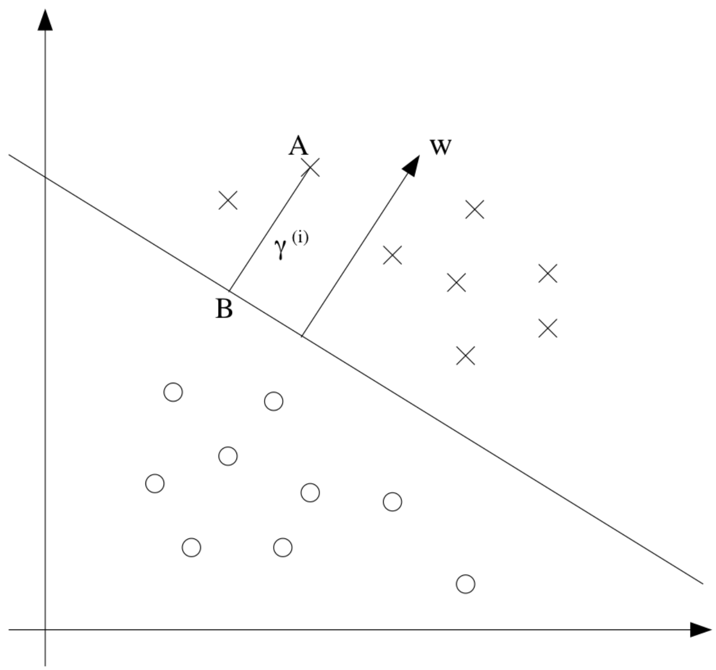
$$y \in \{-1,1\}$$

$$h_{w,b}(x) = g(w^T x + b)$$

g(z) = 1 if z >= 0 (more confident if z >> 0)

g(z) = -1 otherwise (more confident if z << 0)

# Margins




Functional margin

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x + b).$$

We can scale w and b, changing functional margin but not output of $h_{w,b}(x)$

- Geometric margin: distance from our training point to the decision boundary

$$\gamma^{(i)} = y^{(i)} \left( \left( \frac{w}{||w||} \right)^T x^{(i)} + \frac{b}{||w||} \right).$$

# Margins

- Given a training set:

$$S = \{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\}$$

- Function margin of (w,b) w.r.t. S:

$$\hat{\gamma} = \min_{i=1,\ldots,m} \hat{\gamma}^{(i)}.$$

- Geometric margin of (w,b) w.r.t S:

$$\gamma = \min_{i=1,\ldots,m} \gamma^{(i)}.$$

# Optimal Margin Classifier

- Find decision boundary that maximizes the geometric margin (the "gap")

$$\max_{\gamma,w,b} \quad \frac{\hat{\gamma}}{||w||}$$
$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq \hat{\gamma}, \quad i = 1, \ldots, m$$

- Recall: we can add arbitrary scaling constraint on w and b without changing anything

$$\min_{\gamma,w,b} \quad \frac{1}{2}||w||^2$$
$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, m$$

Can be solved with QP

# Lagrange Duality

- Why?
  - To formulate our optimization objective in its dual form, that allows us to use kernels
  - To derive efficient algorithm for solving the optimization problem
- Primal Optimization Problem

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0, \quad i = 1, \ldots, k \\ & h_i(w) = 0, \quad i = 1, \ldots, l. \end{aligned}$$

# Primal

- Define the **generalized Lagrangian** to solve the primal optimization problem

$$\mathcal{L}(w,\alpha,\beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w).$$

- Define

$$\theta_{\mathcal{P}}(w) = \max_{\alpha,\beta\,:\,\alpha_i \geq 0} \mathcal{L}(w,\alpha,\beta).$$

$$\theta_{\mathcal{P}}(w) = \begin{cases} f(w) & \text{if } w \text{ satisfies primal constraints} \\ \infty & \text{otherwise.} \end{cases}$$

# Primal

- Hence, we can rewrite the primal optimization problem as:

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \le 0, \quad i = 1, \dots, k \\ & h_i(w) = 0, \quad i = 1, \dots, l. \end{aligned}$$

$$\min_w \theta_{\mathcal{P}}(w) = \min_w \max_{\alpha, \beta : \alpha_i \ge 0} \mathcal{L}(w, \alpha, \beta),$$

- Define the value of our primal problem as p*

$$p^* = \min_w \theta_{\mathcal{P}}(w)$$

# Dual

- Define

$$\theta_{\mathcal{D}}(\alpha, \beta) = \min_{w} \mathcal{L}(w, \alpha, \beta).$$

- And the Dual Optimization Problem as:

$$\max_{\alpha,\beta\,:\,\alpha_i \geq 0} \theta_{\mathcal{D}}(\alpha, \beta) = \max_{\alpha,\beta\,:\,\alpha_i \geq 0} \min_{w} \mathcal{L}(w, \alpha, \beta).$$

- Define the value of our dual problem as d*

$$d^* = \max_{\alpha,\beta\,:\,\alpha_i \geq 0} \theta_{\mathcal{D}}(w)$$

# Primal and Dual

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

- Under certain condition d* = p*
- Karush-Kuhn-Tucker conditions

$$\begin{aligned}
\frac{\partial}{\partial w_i}\mathcal{L}(w^*,\alpha^*,\beta^*) &= 0, \quad i = 1,\ldots,n \\
\frac{\partial}{\partial \beta_i}\mathcal{L}(w^*,\alpha^*,\beta^*) &= 0, \quad i = 1,\ldots,l \\
\alpha_i^* g_i(w^*) &= 0, \quad i = 1,\ldots,k \\
g_i(w^*) &\leq 0, \quad i = 1,\ldots,k \\
\alpha^* &\geq 0, \quad i = 1,\ldots,k
\end{aligned}$$

"**Dual Complimentarity**"
if $\alpha_i^* > 0$ then $g_i(w^*) = 0$
i.e. the constraint is "active"

# Optimal Margin Classifier - Dual

- Recall our "primal" optimization problem:

$$\min_{\gamma,w,b} \quad \frac{1}{2}||w||^2$$
$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, m$$

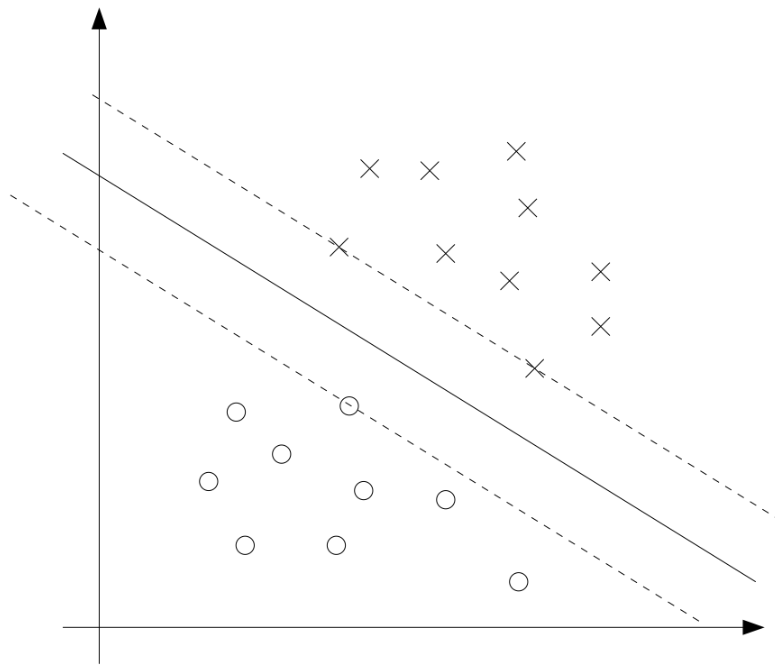
- Rewrite the constraints as:

$$g_i(w) = -y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0.$$

- By "dual complimentarity" condition, $\alpha_i > 0$ only for training examples that have functional margin = 1

# Optimal Margin Classifier - Dual

- Our **support vectors**



# Optimal Margin Classifier - Dual

- The Lagrangian for our optimization problem:

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 \\ \text{s.t.} \quad & g_i(w) = -y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0. \end{aligned}$$

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i \left[ y^{(i)}(w^T x^{(i)} + b) - 1 \right].$$

# Optimal Margin Classifier - Dual

- First we minimize w.r.t w and b:

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i \left[ y^{(i)}(w^T x^{(i)} + b) - 1 \right].$$

$$\nabla_w \mathcal{L}(w,b,\alpha) = w - \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} = 0$$

$$w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}.$$

$$\frac{\partial}{\partial b} \mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i y^{(i)} = 0.$$

# Optimal Margin Classifier - Dual

- Plugging it back to our Lagrangian:

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i \left[ y^{(i)}(w^T x^{(i)} + b) - 1 \right].$$

$$\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)} - b \sum_{i=1}^{m} \alpha_i y^{(i)}.$$

$$\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)}.$$

- And then maximize w.r.t. α

$$\begin{aligned} \max_\alpha \quad & W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle. \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, m \\ & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0, \end{aligned}$$

Inner product

# Optimal Margin Classifier - Dual

- Once solved for α, you can find:

$$w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}.$$

$$b^* = -\frac{\max_{i:y^{(i)}=-1} {w^*}^T x^{(i)} + \min_{i:y^{(i)}=1} {w^*}^T x^{(i)}}{2}.$$

- Given a new point, classify using:

$$\begin{aligned} w^T x + b &= \left( \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \right)^T x + b \\ &= \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b. \end{aligned}$$

Inner product

# Kernels

- Dual form of our optimization problem allows to write our algorithm in terms of inner products
- Exploit this using kernels
- The resulting algorithm – Support Vector Machines – can learn efficiently in very high dimensional spaces

# Kernels

- Given an input feature x, we can define a feature mapping:

$$\phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}.$$

- Apply SVM using this feature – replace all inner products <x,z> with <ϕ(x), ϕ(z)> or the kernel

$$K(x,z) = \phi(x)^T \phi(z).$$

# Kernels

- Although ϕ(x) may be *expensive (O($n^2$))* to compute, K(x,z) may be *inexpensive (O(n))*

$$K(x,z) = (x^T z)^2.$$

$$\begin{aligned} K(x,z) &= \left(\sum_{i=1}^n x_i z_i\right)\left(\sum_{j=1}^n x_i z_i\right) \\ &= \sum_{i=1}^n \sum_{j=1}^n x_i x_j z_i z_j \\ &= \sum_{i,j=1}^n (x_i x_j)(z_i z_j) \end{aligned}$$

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}.$$

- Hence we can get SVM to learn in the high dimensional feature space without ever having to explicitly represent vectors ϕ(x)

# Kernels

- More generally,

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

Corresponds to a feature mapping to $O(n^d)$-dimensional space

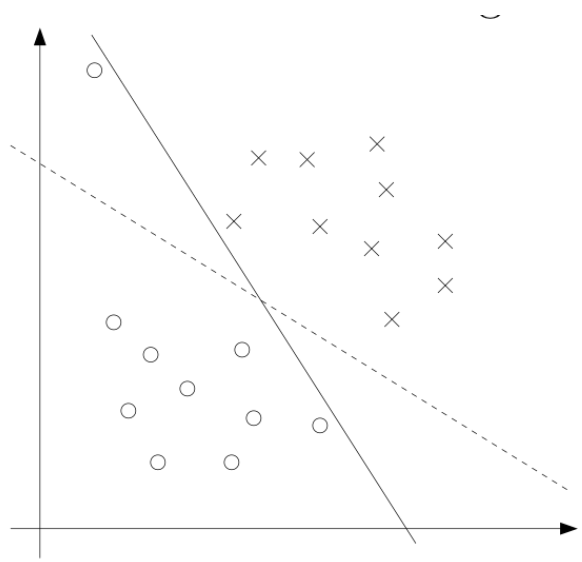
- But computing K(x,z) still takes O(n) time

# Kernels

- Intuitively, $K(x,z)$ is some measure of how similar x and z
- A kernel is a valid kernel is if there exists some feature mapping ϕ such that $K(x,z) = \phi(x)^T\phi(z)$ for all x, z.
- **Theorem (Mercer)**: Given any *m* points $\{x^{(1)},\ldots,x^{(m)}\}$, and an *m*-by-*m* Kernel matrix, where its (i,j) entry is $K(x^{(i)}, x^{(j)})$, K is a valid kernel if and only if the corresponding kernel matrix is symmetric positive semi definite

# Non Linearly Separable Case

- Mapping to high dimensional feature space increases the likelihood that the data is separable (but not always)



Sometimes we don't want to separate training data exactly
Recall: Overfitting

# Non Linearly Separable Case

- We want to allow for some mistakes:

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m \\ & \xi_i \geq 0, \quad i = 1, \ldots, m. \end{aligned}$$

- Allow functional margin to be less than 1
- Whenever that happens pay the cost of $C\xi_i$
- C is the tradeoff between making large margin and making mistakes

# Non Linearly Separable Case

- Lagrangian

$$\mathcal{L}(w,b,\xi,\alpha,r) = \frac{1}{2}w^T w + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left[y^{(i)}(x^T w + b) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i.$$

- Dual form

$$\begin{aligned}
\max_\alpha \quad & W(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle \\
\text{s.t.} \quad & 0 \le \alpha_i \le C, \quad i = 1,\ldots,m \\
& \sum_{i=1}^{m}\alpha_i y^{(i)} = 0,
\end{aligned}$$

# Non Linearly Separable Case

- KKT dual complimentarity conditions:

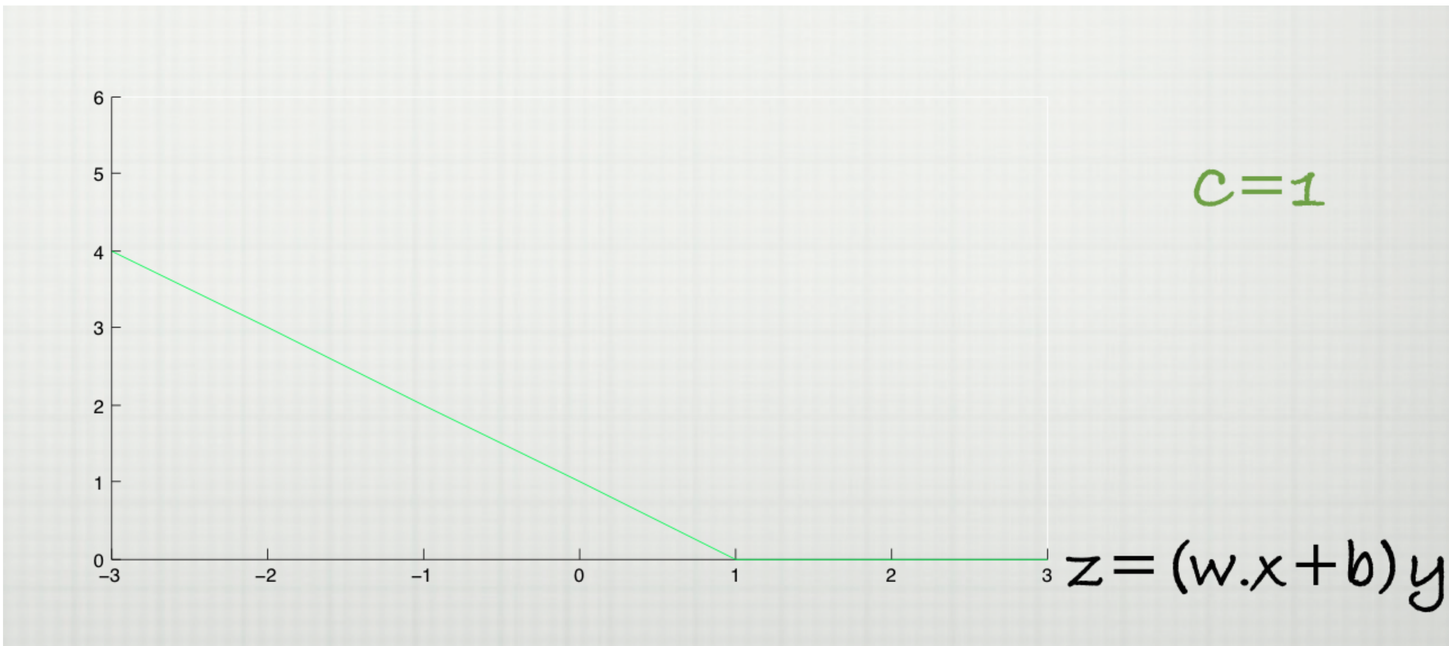
$$
\begin{aligned}
\alpha_i = 0 \quad &\Rightarrow \quad y^{(i)}(w^T x^{(i)} + b) \geq 1 \\
\alpha_i = C \quad &\Rightarrow \quad y^{(i)}(w^T x^{(i)} + b) \leq 1 \\
0 < \alpha_i < C \quad &\Rightarrow \quad y^{(i)}(w^T x^{(i)} + b) = 1.
\end{aligned}
$$

# Non Linearly Separable Case

- Loss part: $C\Sigma\xi_i$
- $\xi \geq 0$ only if the functional margin, $(wx+b)y < 1$
- From our constraints, we want $\xi \geq 1 - (wx+b)y$ and minimize $\xi$ at the same time
  - Hence, $\xi = 1 - (wx+b)y$
- Loss = $C(1 - (wx+b)y)$ only if $(wx+b)y < 1$

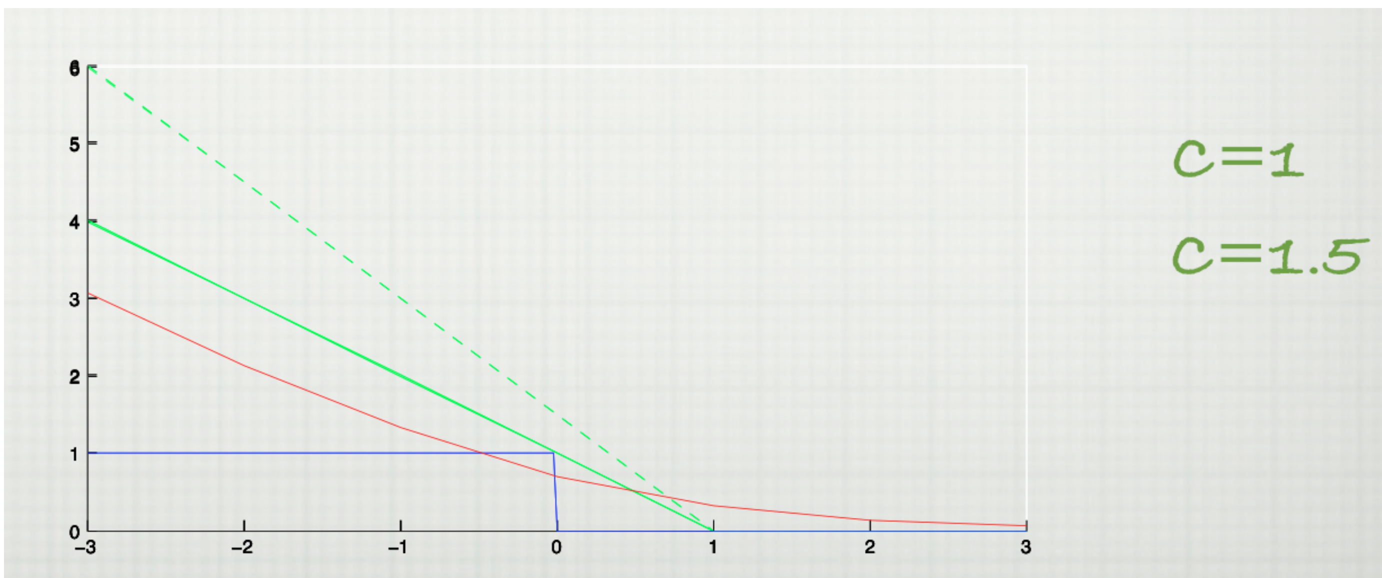
# Non Linearly Separable Case

- Hinge Loss



# Other Loss function

- Hinge Loss L = 1 – (wx+b)y only if (wx+b)y < 1
- 0/1 loss L = 1 if (wx+b)y < 0, 0 otherwise
- Logistic Loss L = log(1 + exp((-wx+b)y))



# SMO
# (Sequential Minimal Optimization)

- One efficient way to solve the dual problem
- A kind of coordinate ascent algorithm:

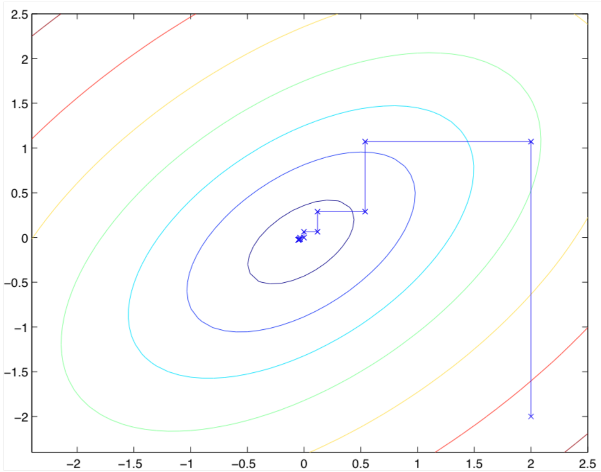
Loop until convergence: {

For $i = 1, \ldots, m$, {

$$\alpha_i := \arg\max_{\hat{\alpha}_i} W(\alpha_1, \ldots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \ldots, \alpha_m).$$

}

}



# SMO

- The dual optimization problem:

$$\max_\alpha \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle.$$

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

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0.$$

- 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. Reoptimize $W(\alpha)$ with respect to $\alpha_i$ and $\alpha_j$, while holding all the other $\alpha_k$'s ($k \neq i, j$) fixed.

}

# SMO

- Dual optimization problem:

$$\max_\alpha \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle.$$

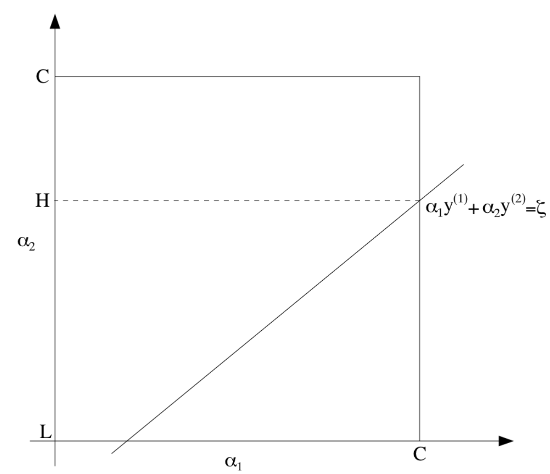
$$\text{s.t.} \quad 0 \le \alpha_i \le C, \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0.$$

- Say we have picked $\alpha_1$ and $\alpha_2$ to optimize:

$$\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = -\sum_{i=3}^{m} \alpha_i y^{(i)}.$$

$$\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = \zeta.$$



# SMO

- We can express $\alpha_1$ in terms of $\alpha_2$

$$\alpha_1 = (\zeta - \alpha_2 y^{(2)})/y^{(1)}.$$

- Substituting this back to our optimization objective W(α):

$$W(\alpha_1, \alpha_2, \ldots, \alpha_m) = W((\zeta - \alpha_2 y^{(2)})/y^{(1)}, \alpha_2, \ldots, \alpha_m).$$

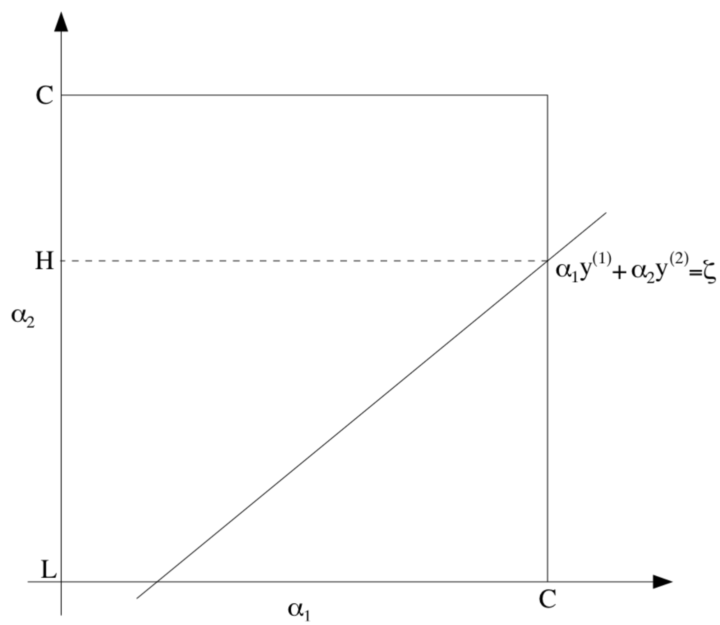
- Since $\alpha_2$ to $\alpha_m$ are held fixed, W(α) takes the form of a quadratic equation:

$$a\alpha_2^2 + b\alpha_2 + c$$

# SMO

- Solving this quadratic equation for $\alpha_2$

$$\alpha_2^{new} = \begin{cases} H & \text{if } \alpha_2^{new,unclipped} > H \\ \alpha_2^{new,unclipped} & \text{if } L \leq \alpha_2^{new,unclipped} \leq H \\ L & \text{if } \alpha_2^{new,unclipped} < L \end{cases}$$



# SVM

- Optimal Margin Classifier
- Dual form is useful
- Support vectors are neat
- Kernel trick is cool
- Different loss function

# Review

| Learning Method | Generative or Discriminative | Loss Function | Decision Boundary | Parameter Estimation Algorithm | Model Complexity Reduction |
|---|---|---|---|---|---|
| Gaussian Naïve Bayes | Generative | -log P(X,Y) | Equal variance: linear boundary<br>Unequal variance: quadratic boundary | Estimate μ and σ and prior P(Y) using maximum likelihood | Place prior on parameters and use MAP estimator |
| Logistic Regression | Discriminative | -log P(Y\|X) | Linear | No closed form estimate. Optimize objective function using gradient descent | $L_2$ regularization/ $L_1$ regularization |

# Review

| Learning Method | Generative or Discriminative | Loss Function | Decision Boundary | Parameter Estimation Algorithm | Model Complexity Reduction |
|---|---|---|---|---|---|
| Decision Trees | Discriminative | Either –log P(Y\|X) or zero-one loss | Axis-aligned partition of feature space | Many algorithms, ID3, CART, C4.5 | Prune tree or limit tree depth |
| K-NN | Discriminative | Zero-one loss | Arbitrarily complex | Must store all training data to classify new points. Choose K using cross validation | Increase K |
| SVM | Discriminative | Hinge-loss: C(1-y(wx+b)) only if y(wx+b) < 1, 0 otherwise | Linear (depends on kernel) | Solve using quadratic program (or SMO) to find boundary that maximizes margin | Reduce C |
| Linear Regression (Gaussian Noise) | Discriminative | Square loss: $(f(X) – Y)^2$ | Linear | Solve $\beta = (X^TX)^{-1}X^TY$ | $L_2$ regularization/ $L_1$ regularization |

# Acknowledgment


- Sue Ann recitation slides on SVM

http://www.cs.cmu.edu/~guestrin/Class/15781/recitations/r7/20071018svm.pdf

- Andrew Ng notes on SVM

http://cs229.stanford.edu/notes/cs229-notes3.pdf