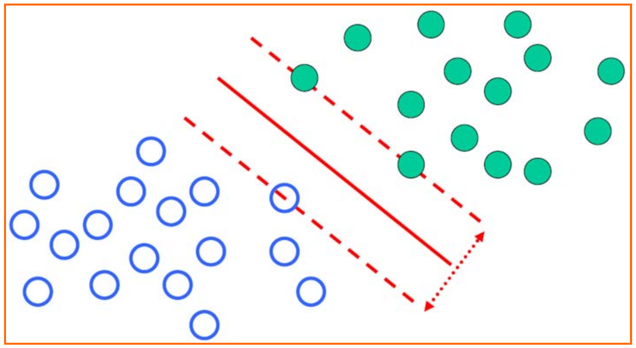
Machine Learning

10-701, Fall 2015

Support Vector Machines

Eric Xing

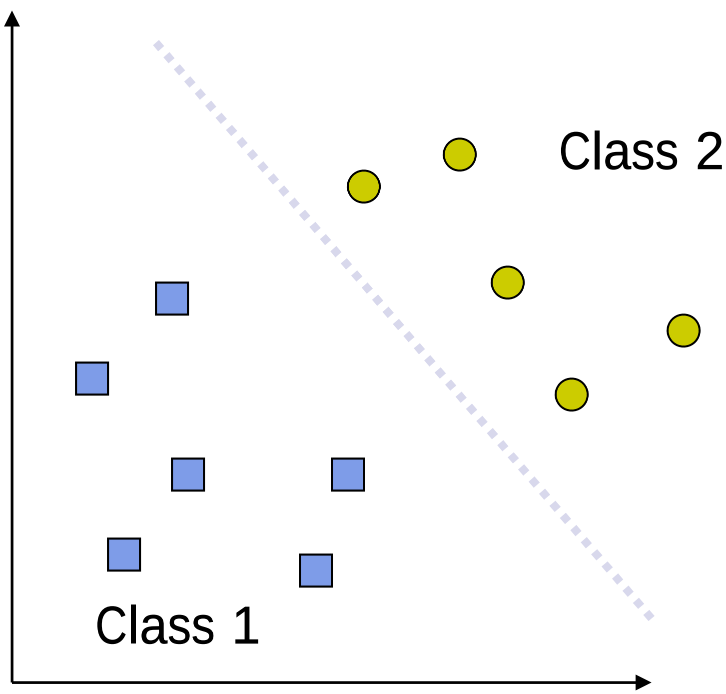


Lecture 9, October 6, 2015

Reading: Chap. 6&7, C.B book, and listed papers

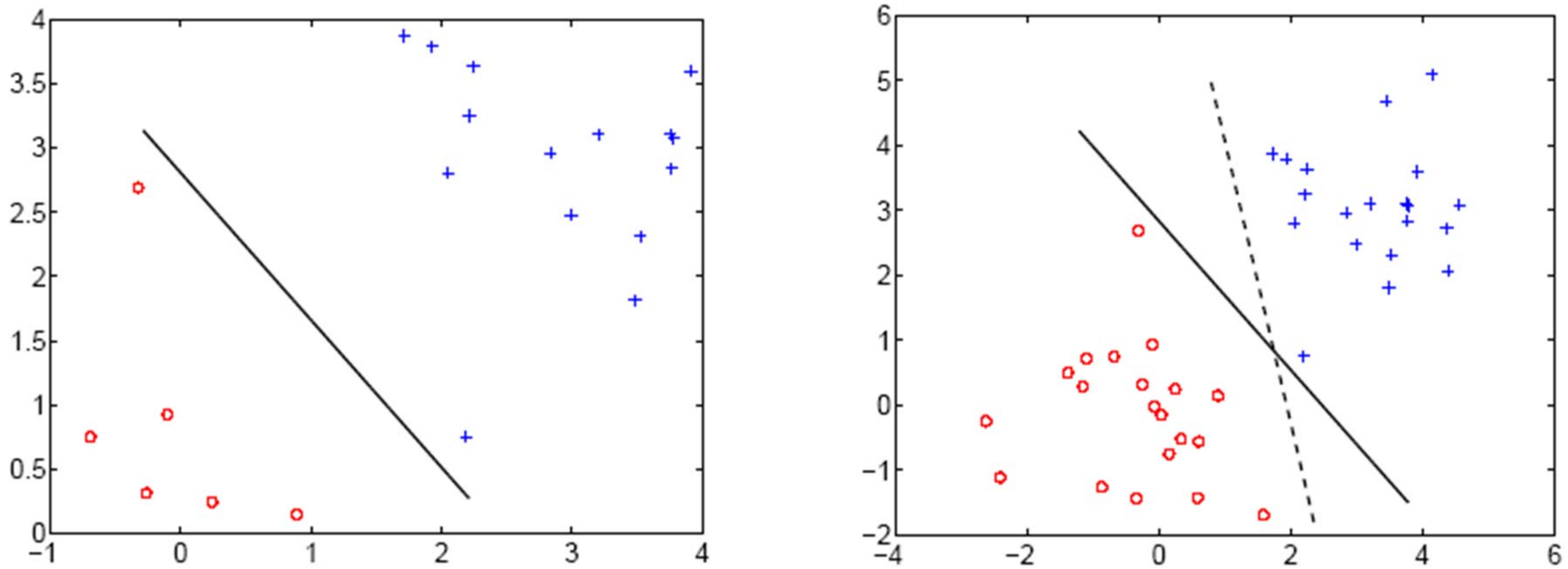
What is a good Decision Boundary?

- Consider a binary classification task with y = ±1 labels (not 0/1 as before).
- When the training examples are linearly separable, we can set the parameters of a linear classifier so that all the training examples are classified correctly
- Many decision boundaries!
  - Generative classifiers
  - Logistic regressions ...
- Are all decision boundaries equally good?



What is a good Decision Boundary?

Not All Decision Boundaries Are Equal!

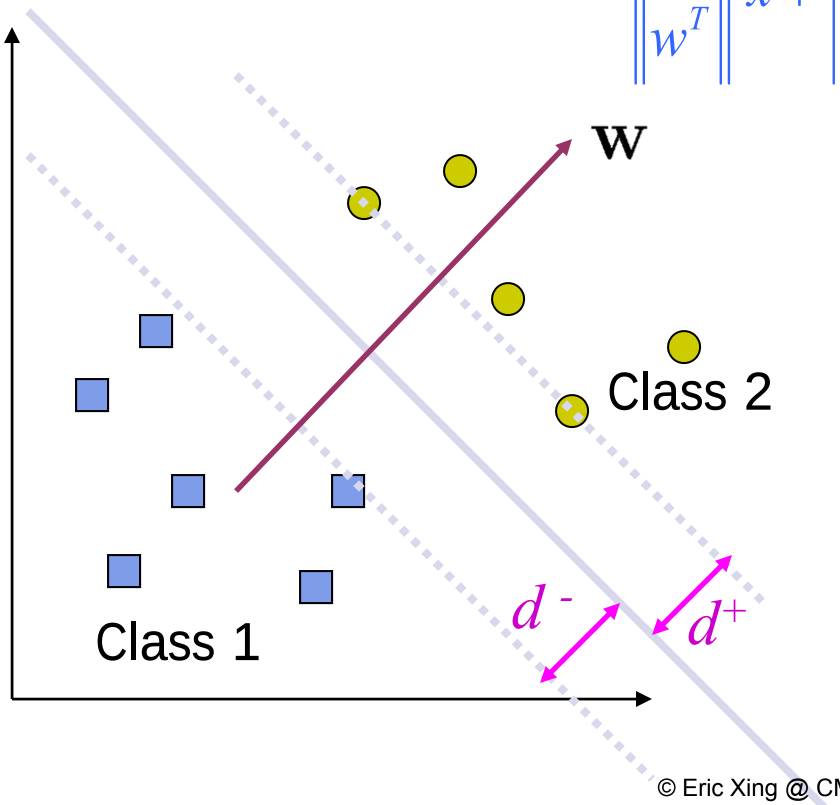


- Why we may have such boundaries?
  - Irregular distribution
  - Imbalanced training sizes
  - outliners

Classification and Margin

- Parameterzing decision boundary
  - Let w denote a vector orthogonal to the decision boundary, and b denote a scalar "offset" term, then we can write the decision boundary as:

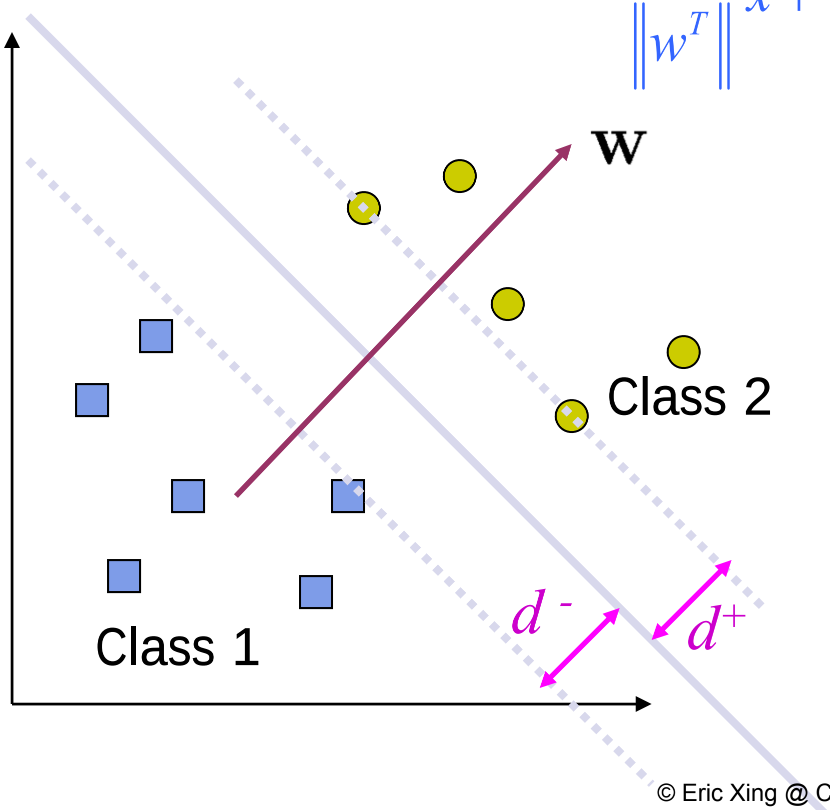
$$\frac{w^T}{\|w^T\|}x + \frac{b}{\|w^T\|} = 0$$



Classification and Margin

- Parameterzing decision boundary
  - Let w denote a vector orthogonal to the decision boundary, and b denote a scalar "offset" term, then we can write the decision boundary as:

$$\frac{w^T}{\|w^T\|}x + \frac{b}{\|w^T\|} = 0$$



- Margin

$(w^T x_i + b)/\|w\| > +c/\|w\|$ for all x_i in class 2

$(w^T x_i + b)/\|w\| < -c/\|w\|$ for all x_i in class 1

Or more compactly:

$$(w^T x_i + b) y_i / \|w\| > c/\|w\|$$

The margin between any two points

$\boldsymbol{m} = d^- + d^+ =$

Maximum Margin Classification

- The minimum permissible margin is:

$$m = \frac{w^T}{\|w\|}\left(x_{i^*} - x_{j^*}\right) = \frac{2c}{\|w\|}$$

- Here is our Maximum Margin Classification problem:

$$\max_w \quad \frac{2c}{\|w\|}$$

$$\text{s.t} \quad y_i(w^T x_i + b)/\|w\| \geq c/\|w\|, \quad \forall i$$

Maximum Margin Classification, con'd.

- The optimization problem:

$$\max_{w,b} \quad \frac{c}{\|w\|}$$
$$\text{s.t} \quad y_i(w^T x_i + b) \geq c, \quad \forall i$$

- But note that the magnitude of c merely scales w and b, and does not change the classification boundary at all! (why?)
- So we instead work on this cleaner problem:

$$\max_{w,b} \quad \frac{1}{\|w\|}$$
$$\text{s.t} \quad y_i(w^T x_i + b) \geq 1, \quad \forall i$$

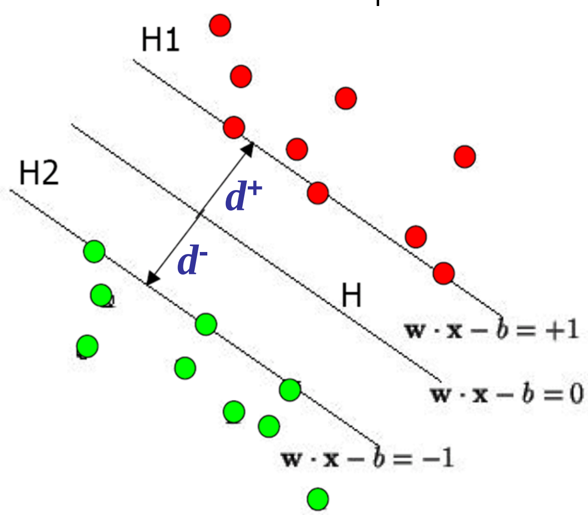
- The solution to this leads to the famous **Support Vector Machines** --- believed by many to be the best "off-the-shelf" supervised learning algorithm

Support vector machine

- A convex quadratic programming problem with linear constrains:

$$\max_{w,b} \frac{1}{\|w\|}$$
$$\text{s.t} \quad y_i(w^T x_i + b) \ge 1, \quad \forall i$$

  - The attained margin is now given by $\frac{1}{\|w\|}$
  - Only a few of the classification constraints are relevant ➔ **support vectors**



- Constrained optimization
  - We can directly solve this using commercial quadratic programming (QP) code
  - But we want to take a more careful investigation of Lagrange duality, and the solution of the above in its dual form.

➔ deeper insight: support vectors, kernels …

➔ more efficient algorithm

Digression to Lagrangian Duality

- The Primal Problem

Primal:

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

The generalized Lagrangian:

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

the α's ($\alpha_i \ge 0$) and β's are called the Lagarangian multipliers

Lemma:

$$\max_{\alpha, \beta, \alpha_i \ge 0} \mathcal{L}(w, \alpha, \beta) = \begin{cases} f(w) & \text{if } w \text{ satisfies primal constraints} \\ \infty & \text{o/w} \end{cases}$$

A re-written Primal:

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

Lagrangian Duality, cont.

- Recall the Primal Problem:

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

- 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):**

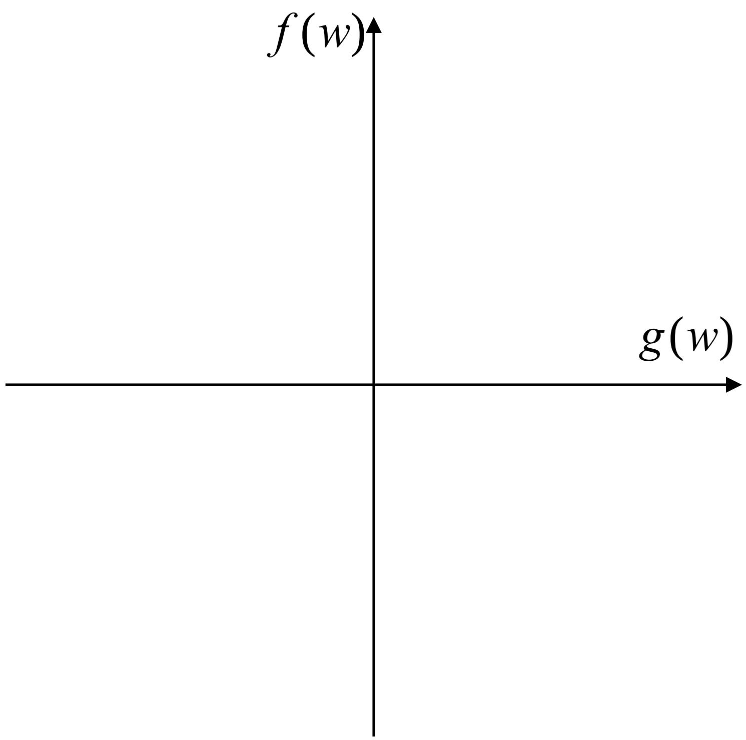
  Iff there exist a saddle point of $\mathcal{L}(w,\alpha,\beta)$, we have

$$d^* = p^*$$

A sketch of strong and weak duality

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

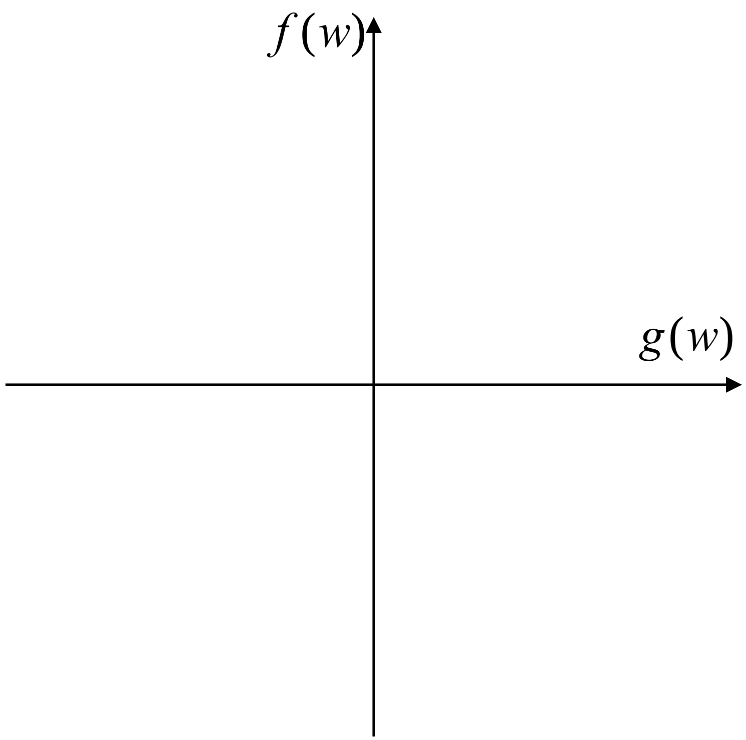
$$d^* = \max_{\alpha_i \geq 0} \min_w f(w) + \alpha^T g(w) \;\leq\; \min_w \max_{\alpha_i \geq 0} f(w) + \alpha^T g(w) = p^*$$



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.

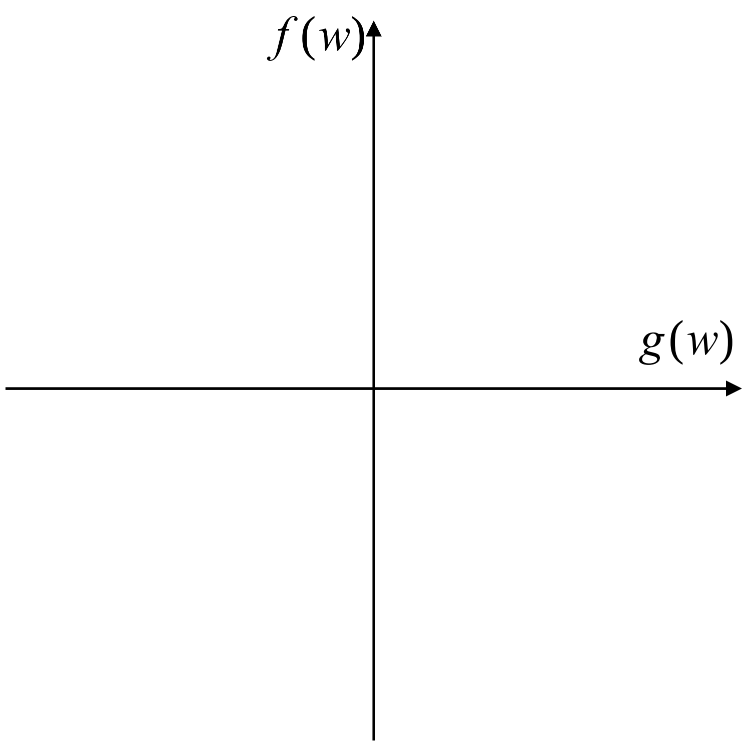
$$d^* = \max_{\alpha_i \geq 0} \min_w f(w) + \alpha^T g(w) \;\leq\; \min_w \max_{\alpha_i \geq 0} f(w) + \alpha^T g(w) = p^*$$

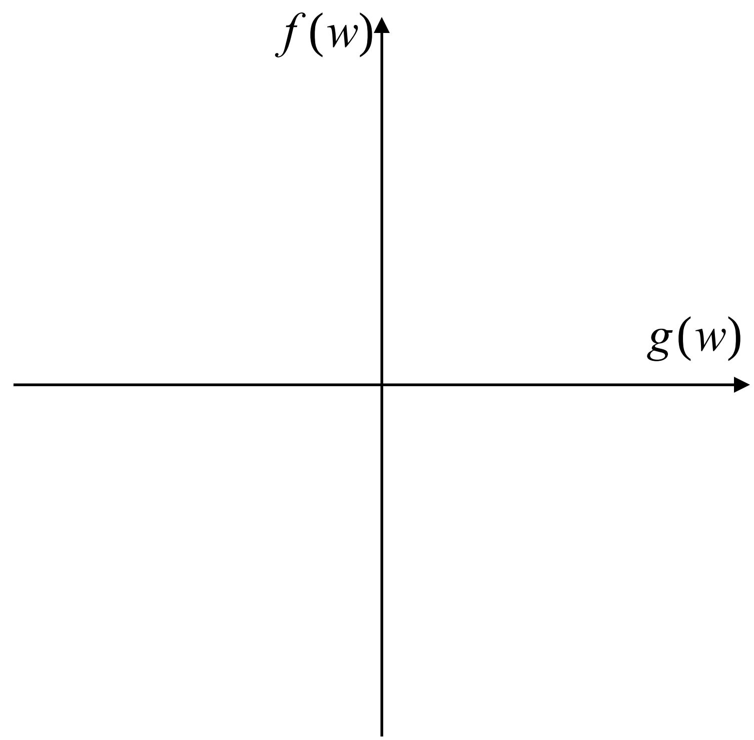


A sketch of strong and weak duality

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

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





The KKT conditions

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

$$\frac{\partial}{\partial w_i}\mathcal{L}(w,\alpha,\beta)=0, \quad i=1,\dots,k$$

$$\frac{\partial}{\partial \beta_i}\mathcal{L}(w,\alpha,\beta)=0, \quad i=1,\dots,l$$

$$\alpha_i g_i(w)=0, \quad i=1,\dots,m$$ Complementary slackness

$$g_i(w)\le 0, \quad i=1,\dots,m$$ Primal feasibility

$$\alpha_i \ge 0, \quad i=1,\dots,m$$ Dual feasibility

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

Solving optimal margin classifier

- Recall our opt problem:

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

- This is equivalent to

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^T w \\ \text{s.t} \quad & 1 - y_i(w^T x_i + b) \leq 0, \quad \forall i \end{aligned} \qquad (*)$$

- Write the Lagrangian:

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

  - Recall that (*) can be reformulated as $\min_{w,b} \max_{\alpha_i \geq 0} \mathcal{L}(w,b,\alpha)$
    Now we solve its **dual problem**: $\max_{\alpha_i \geq 0} \min_{w,b} \mathcal{L}(w,b,\alpha)$

The Dual Problem

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

$$\max_{\alpha_i \geq 0} \min_{w,b} \mathcal{L}(w,b,\alpha)$$

- We minimize $\mathcal{L}$ with respect to w and b first:

$$\nabla_w \mathcal{L}(w,b,\alpha)=w-\sum_{i=1}^{m}\alpha_i y_i x_i=0, \qquad (*)$$

$$\nabla_b \mathcal{L}(w,b,\alpha)=\sum_{i=1}^{m}\alpha_i y_i=0, \qquad (**)$$

Note that **(*)** implies:

$$w=\sum_{i=1}^{m}\alpha_i y_i x_i \qquad (***)$$

- Plug (***) back to $\mathcal{L}$, and using (**), we have:

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

The Dual problem, cont.

- Now we have the following dual opt problem:

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

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

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

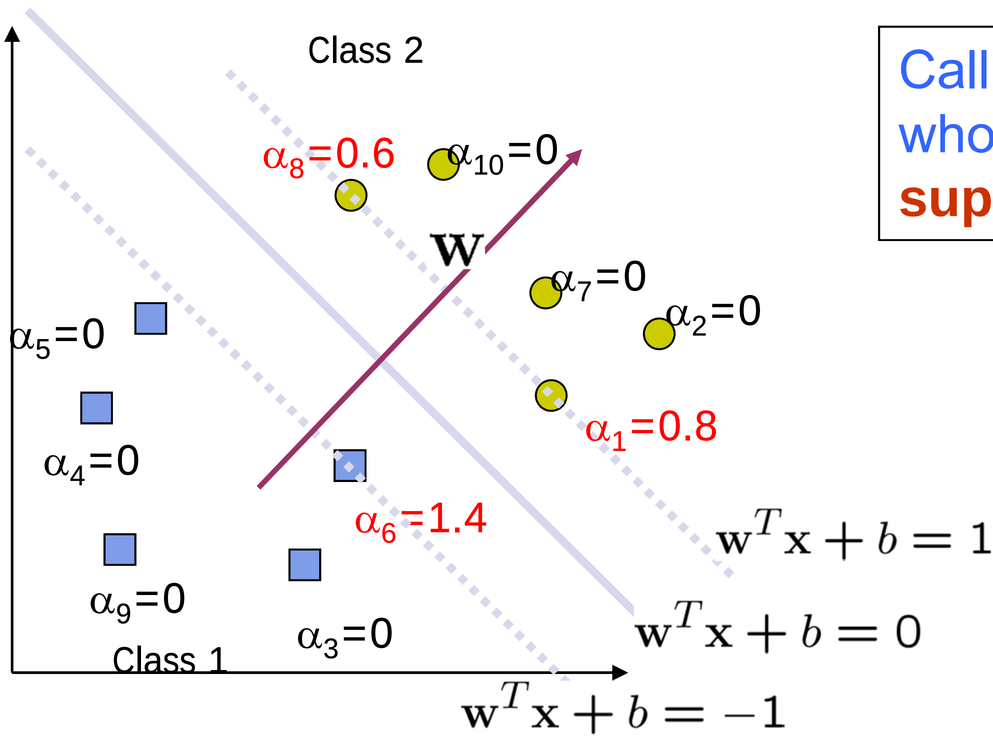
- This is, (again,) a **quadratic programming** problem.
  - A global maximum of α_i can always be found.
  - But what's the big deal??
  - Note two things:

1. w can be recovered by $w = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$ See next …
2. The "kernel" $\mathbf{x}_i^T \mathbf{x}_j$ More later …

Support vectors

- Note the KKT condition --- only a few α_i's can be nonzero!!

$$\alpha_i g_i(w) = 0, \quad i = 1, \ldots, m$$



Call the training data points whose α_i's are nonzero the **support vectors** (SV)

Support vector machines

- Once we have the Lagrange multipliers $\{\alpha_i\}$, we can reconstruct the parameter vector w as a weighted combination of the training examples:

$$w = \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i$$

- For testing with a new data $\boldsymbol{z}$
  - Compute

$$w^T z + b = \sum_{i \in SV} \alpha_i y_i \left(\mathbf{x}_i^T z \right) + b$$

  and classify $\boldsymbol{z}$ as class 1 if the sum is positive, and class 2 otherwise

  - Note: w need not be formed explicitly

Interpretation of support vector machines

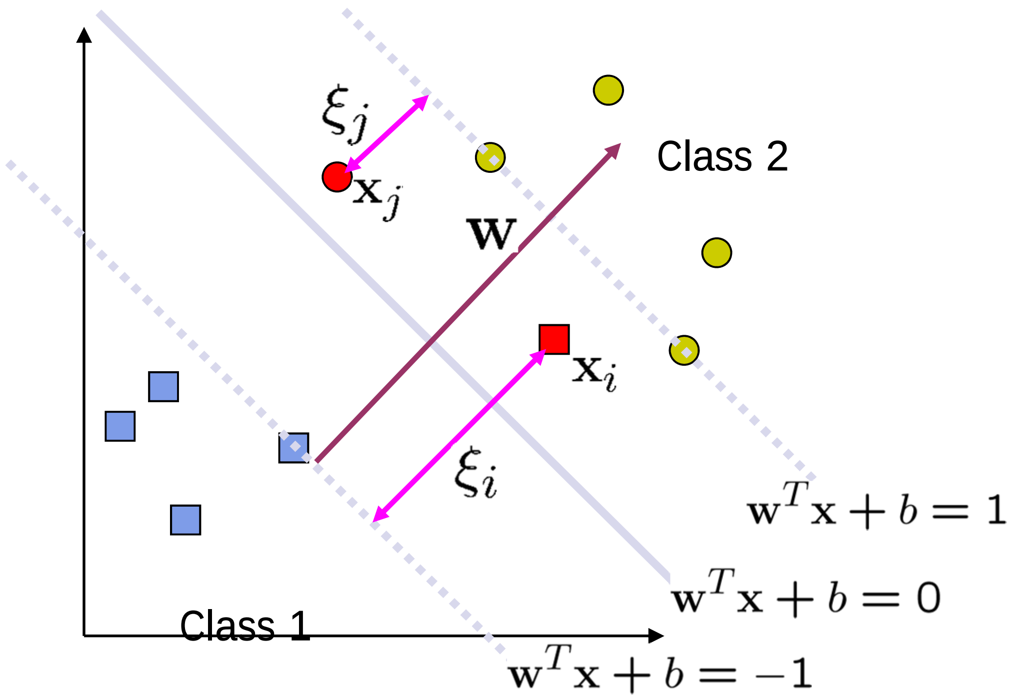
- The optimal w is a linear combination of a small number of data points. This "sparse" representation can be viewed as data compression as in the construction of kNN classifier

- To compute the weights $\{\alpha_i\}$, and to use support vector machines we need to specify only the inner products (or kernel) between the examples $\mathbf{x}_i^T \mathbf{x}_j$

- We make decisions by comparing each new example z with only the support vectors:

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

Non-linearly Separable Problems



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

Soft Margin Hyperplane

- Now we have a slightly different opt problem:

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

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

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

The Optimization Problem

- The dual of this new constrained optimization problem is

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

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

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

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

The SMO algorithm

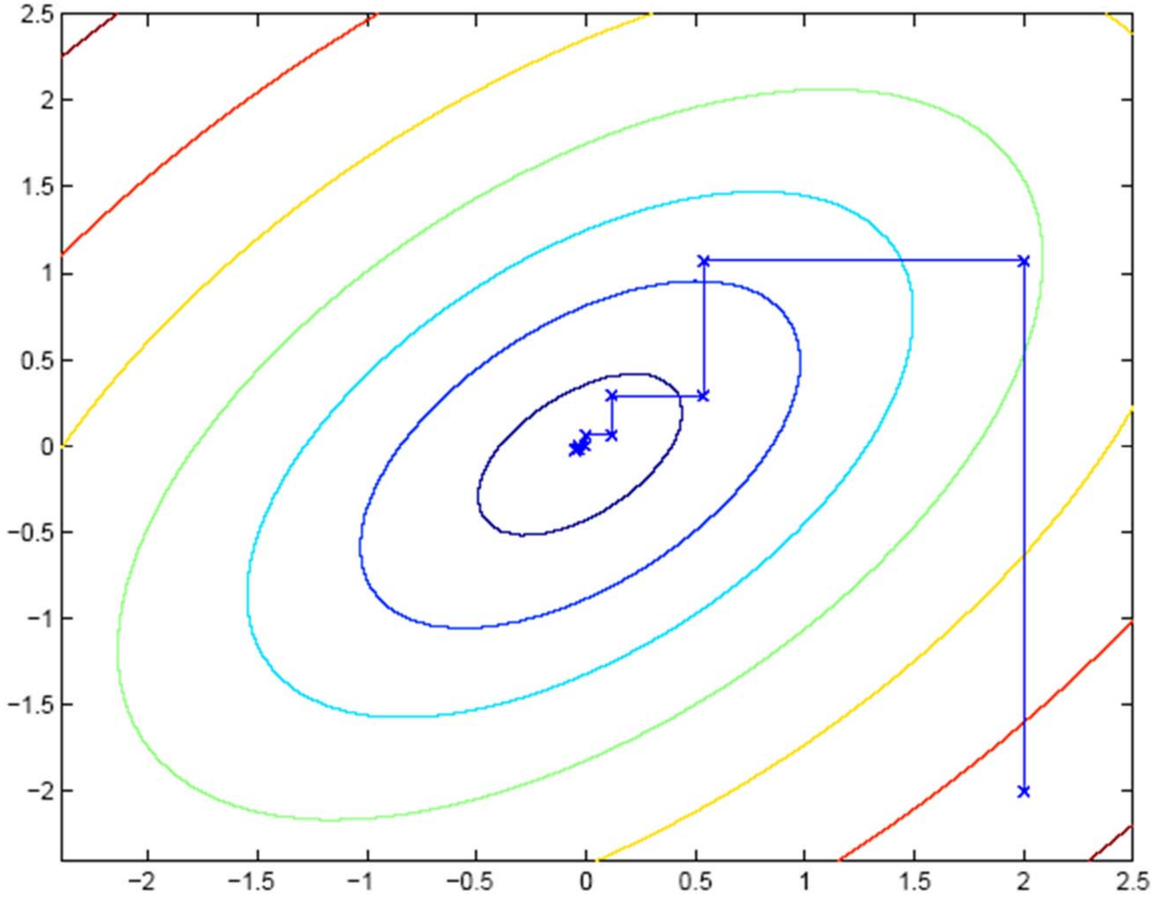
- Consider solving the unconstrained opt problem:

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

- We've already see three opt algorithms!
  - ?
  - ?
  - ?

- Coordinate ascend:

Coordinate ascend



Sequential minimal optimization

- Constrained optimization:

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

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

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

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

The SMO algorithm

Repeat till convergence

1. Select some pair α_i and α_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(α) with respect to α_i and α_j, while holding all the other α_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 α_3 ,…, α_m fixed and reopt J w.r.t. α_1 and α_2

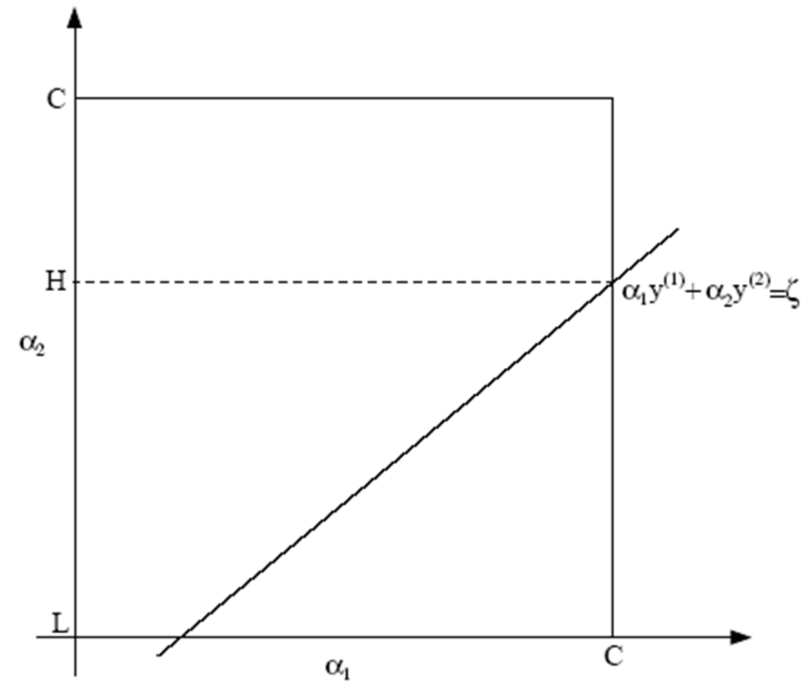
Convergence of SMO

- The constraints:

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

$$0 \leq \alpha_1 \leq C$$

$$0 \leq \alpha_2 \leq C$$



- The objective:

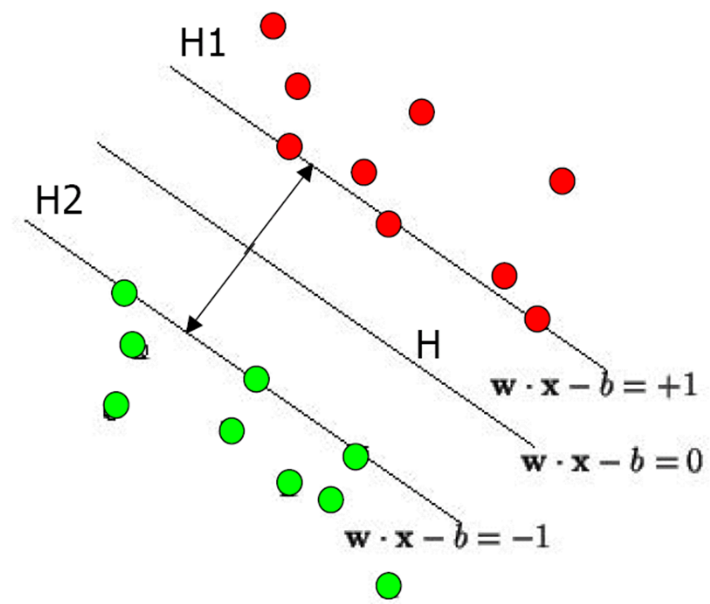
$$\mathcal{J}(\alpha_1, \alpha_2, \ldots, \alpha_m) = \mathcal{J}((\xi - \alpha_2 y_2) y_1, \alpha_2, \ldots, \alpha_m)$$

- 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!

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



Summary

- Max-margin decision boundary
- Constrained convex optimization
  - Duality
  - The KTT conditions and the support vectors
  - Non-separable case and slack variables
  - The SMO algorithm