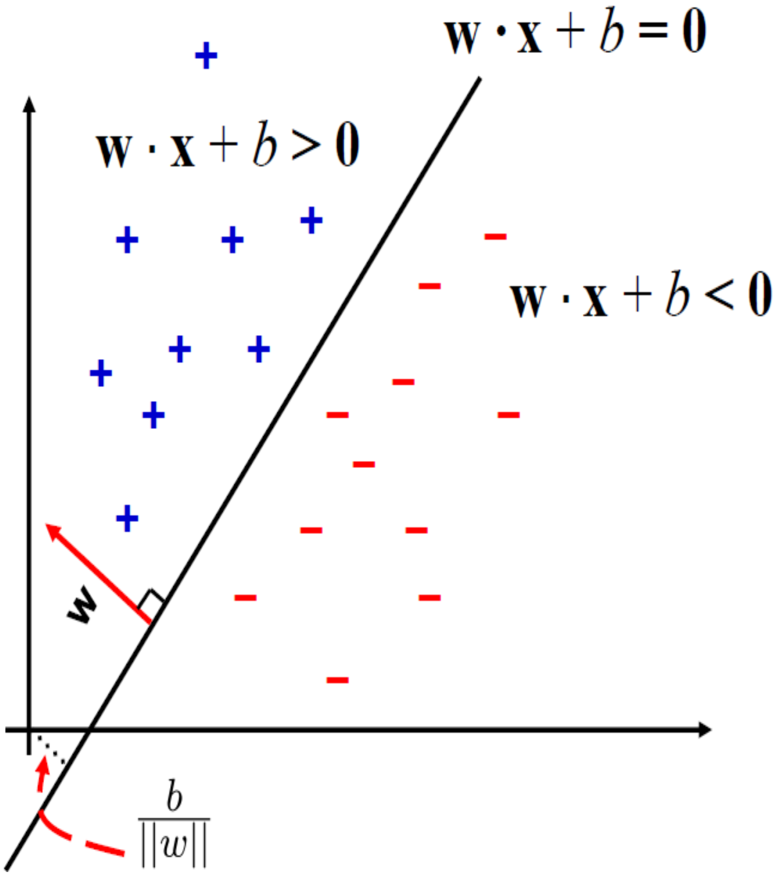
Support Vector Machines

Warning!

- This is just a first introduction to SVMs – very basic version of SVM introduced
- We will discuss more on the advanced techniques later this semester
- Strong assumptions in this case

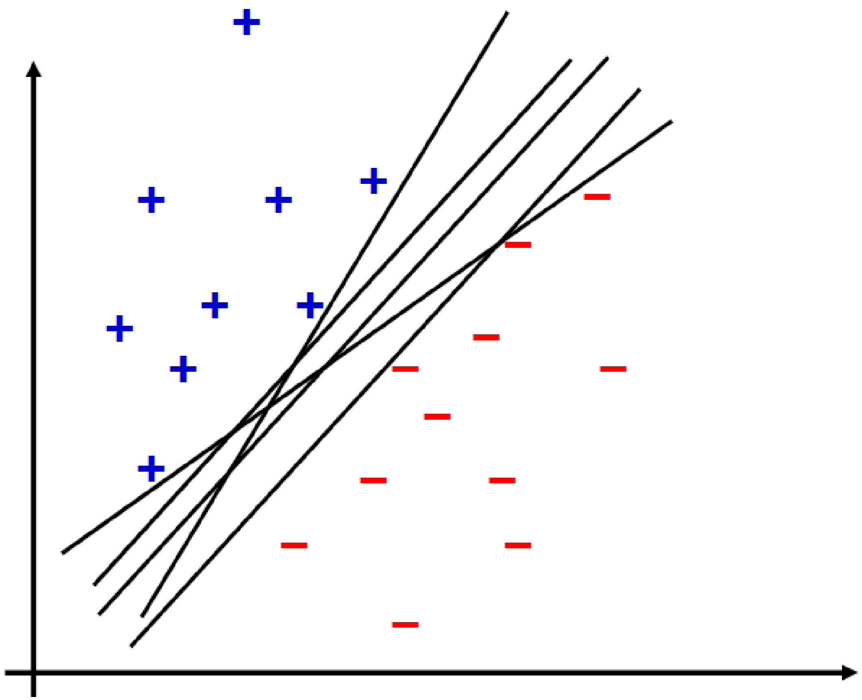
The Classification Problem



- Binary classification can be viewed as the task of separating classes in feature space
- w is the slope of the line
- b is the intercept

Linear Separators

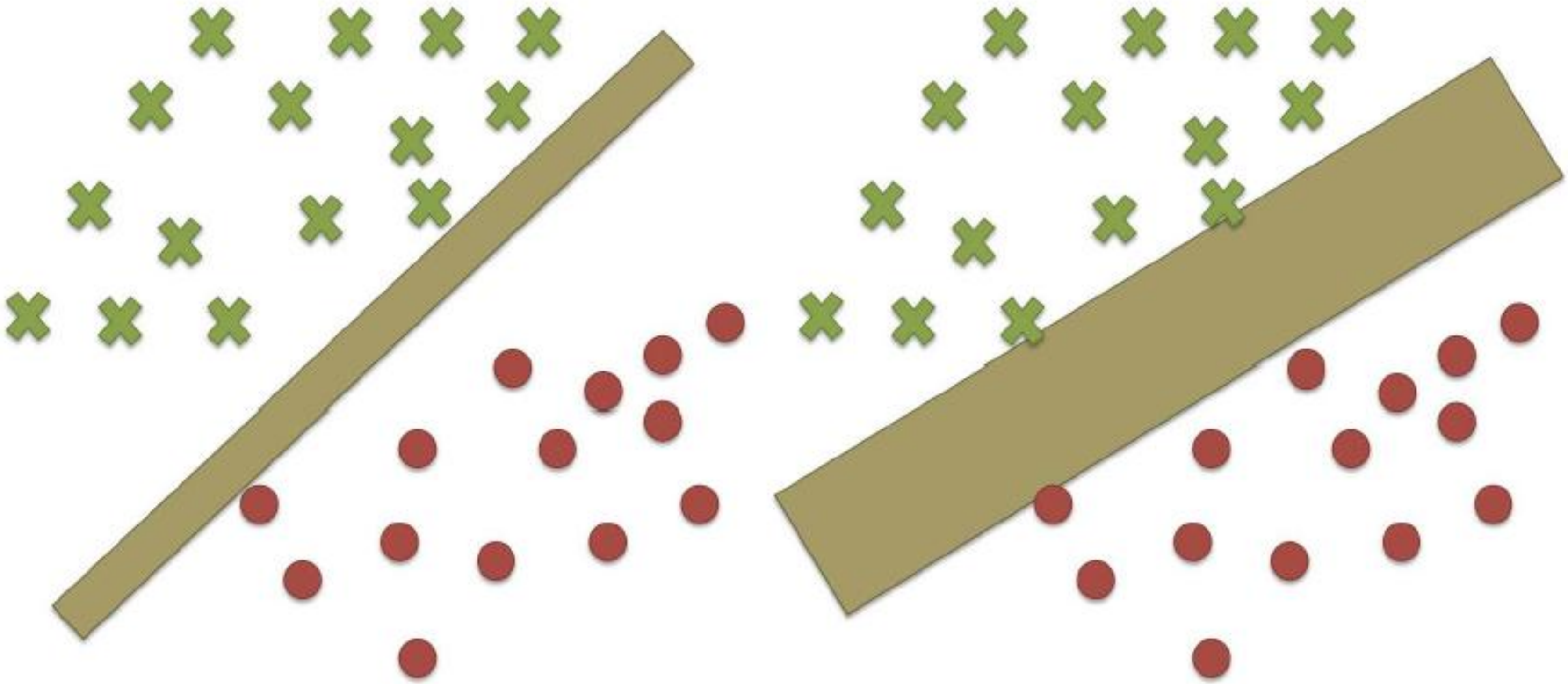
- Which is the best linear separator?



- Depends on the goal
- Goal is to classify **accurately** and **generalize** to new examples.

Notion of Margins

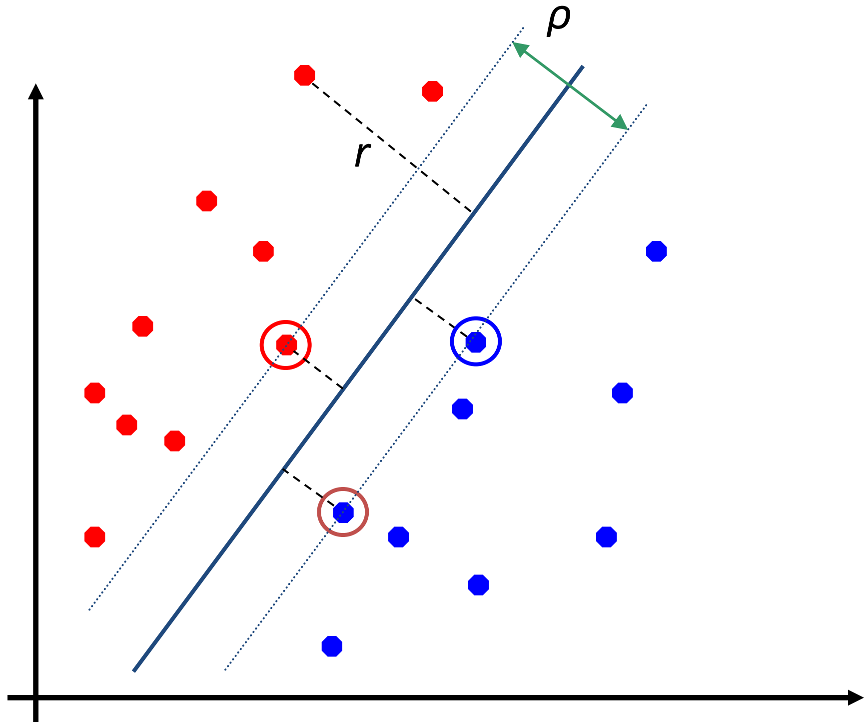
Many different hyperplanes $\mathbf{w}'\mathbf{x} = b$ can classify the data. Which one will work best?



The hyperplane that maximizes the separation between the two classes (*the margin*)

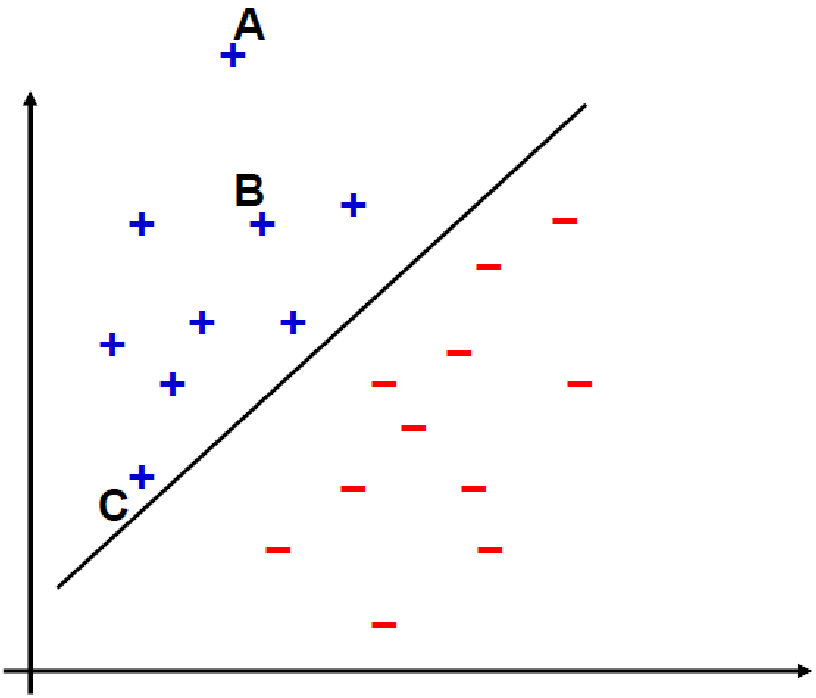
Margins

- Distance from example $\mathbf{x}_i$ to the separator is $r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are ***support vectors***.
- ***Margin*** *ρ* of the separator is the distance between support vectors.



Intuition of a Margin

- Consider points A, B, and C
- We are quite confident in our prediction for A because it is far from the decision boundary.
- In contrast, we are not so confident in our prediction for C because a slight change in the decision boundary may flip the decision.



Given a training set, we would like to make all predictions correct and confident! This leads to the concept of margin.

Notation

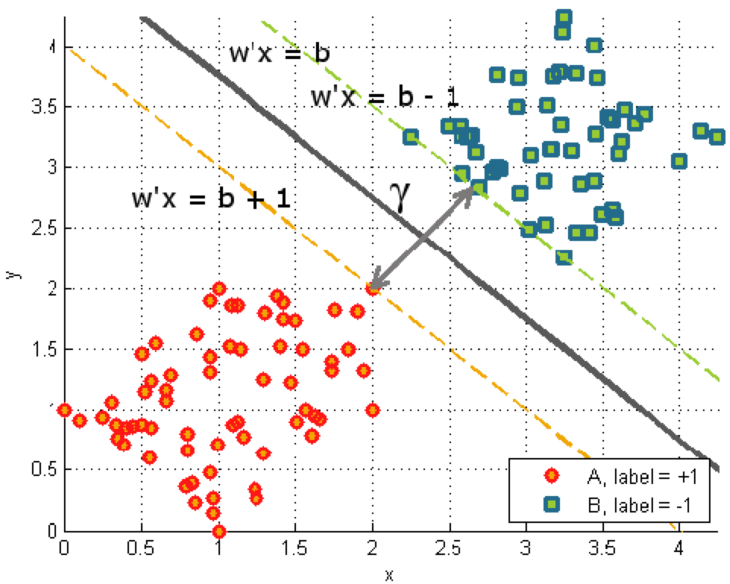
We denote the classifier, $f(\mathbf{x}) = \mathbf{w}'\mathbf{x} - b$, for all $\mathbf{x} \in \mathbb{R}^n$

assume supporting hyperplanes $\mathbf{w}'\mathbf{x} - b = \pm 1$

distance between the two hyperplanes is the *margin*, $\gamma = 2$

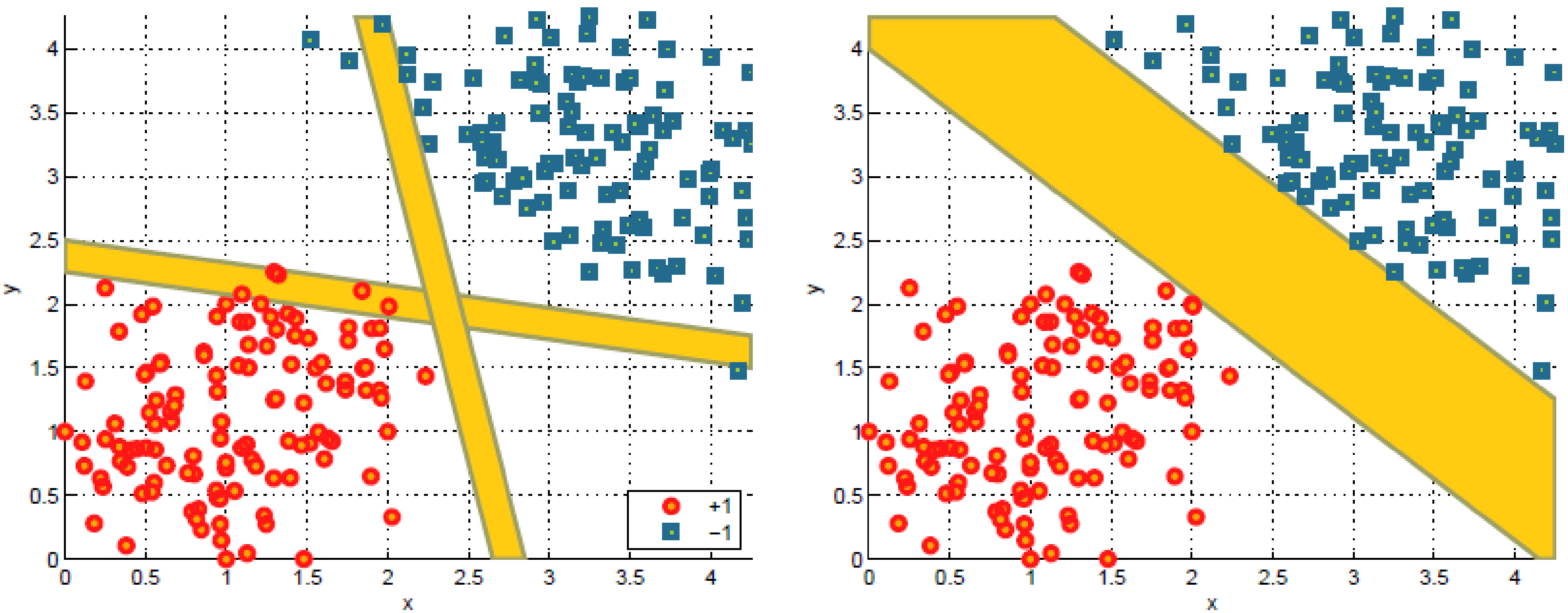
To achieve *scale invariance* divide the classifier by $\|\mathbf{w}\|_2$. Then the supporting hyperplanes are $\hat{\mathbf{w}}'\mathbf{x} - \hat{b} = \pm\frac{1}{\|\mathbf{w}\|_2}$.

margin is $\gamma = \frac{2}{\|\mathbf{w}\|_2}$.



Why Max Margin?

- Minimizes generalization error. Works well on **Future data**
- Minimizes Complexity. **Fewer support vectors**
- Minimizes the capacity of the classifier. **Eliminates overfitting**



Max margin Classifier

- Given a linearly separable training set S={($\boldsymbol{x}^{(i)}$, $y^{(i)}$): *i=1, …, N*}, we would like to find a linear classifier with maximum margin.
- This can be represented as an optimization problem.

$$\max_{\mathbf{w},b,\gamma} \gamma$$

$$\text{subject to: } y^{(i)} \frac{(\mathbf{w}\cdot\mathbf{x}^{(i)}+b)}{\|\mathbf{w}\|} \geq \gamma, \quad i=1,\cdots,N$$

Nasty optimization problem! Let's make it look nicer!

- Let $\gamma' = \gamma \cdot \|w\|$, this is equivalent to

$$\max_{\mathbf{w},b,\gamma'} \frac{\gamma'}{\|\mathbf{w}\|}$$

$$\text{subject to: } y^{i}(\mathbf{w}\cdot\mathbf{x}^{i}+b) \geq \gamma', \quad i=1,\cdots,N$$

Max margin Classifier

- Note that rescaling **w** and *b* by ($1/\gamma$') will not change the classifier, we can thus further reformulate the optimization problem

$$
\boxed{\begin{array}{l}\max\limits_{\mathbf{w},b} \dfrac{\gamma'}{\|\mathbf{w}\|} \\ \text{subject to: } \; y^i(\mathbf{w}\cdot\mathbf{x}^i + b) \ge \gamma', \quad i = 1,\cdots,N\end{array}}
$$

$$
\Downarrow
$$

$$
\boxed{\begin{array}{l}\max\limits_{\mathbf{w},b} \dfrac{1}{\|\mathbf{w}\|} \text{ (or equivalently } \min\limits_{\mathbf{w},b} \|\mathbf{w}\|^2\text{)} \\ \text{subject to: } \; y^i(\mathbf{w}\cdot\mathbf{x}^i + b) \ge 1, \quad i = 1,\cdots,N\end{array}}
$$

Maximizing the geometric margin is equivalent to minimizing the magnitude of **w** subject to maintaining a functional margin of at least 1

Solving the problem

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{subject to: } y^i(\mathbf{w}\cdot\mathbf{x}^i + b) \geq 1, \quad i = 1,\cdots,N$$

- This results in a ***quadratic optimization problem*** with *linear inequality constraints.*
- This is a well-known class of mathematical programming problems for which several (non-trivial) algorithms exist.
  - One could solve for w using any of these methods
- We will see that it is useful to first formulate an equivalent dual optimization problem and solve it instead
  - This requires a bit of machinery

Constrained Optimization

- To solve the following optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to} \quad g_i(\mathbf{x}) \leq 0 \quad \text{for } i = 1, \ldots, m$$

- Consider the following function known as the Lagrangian

$$\mathcal{L}(x, \alpha) = f(\mathbf{x}) + \sum_i \alpha_i g_i(\mathbf{x})$$

- Under certain conditions it can be shown that for a solution x′ to the above problem we have

$$f(x') = \underbrace{\min_x \max_\alpha \mathcal{L}(x, \alpha)}_{\text{Primal form}} = \underbrace{\max_\alpha \min_x \mathcal{L}(x, \alpha)}_{\text{Dual form}}$$

$$\text{subject to } \alpha_i \geq 0$$

Dual Problem

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to}: 1 - y^i(\mathbf{w}\cdot\mathbf{x}^i + b) \le 0, \quad i = 1, \cdots, N$$

- The Lagrangian is

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + \sum_{i=1}^{N}\alpha_i\{1 - y^i(\mathbf{w}\cdot\mathbf{x}^i + b)\}, \text{ subject to } \alpha_i \ge 0$$

- We want to solve $\max_{\alpha} \min_{w,b} \mathcal{L}(w, b, \alpha) \quad s.t. \quad \alpha_i \ge 0$

- Setting the gradient of $\mathcal{L}$ w.r.t. **w** and b to zero, we have

$$\mathbf{w} - \sum_{i=1}^{N}\alpha_i y^i \mathbf{x}^i = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N}\alpha_i y^i \mathbf{x}^i$$

$$\sum_{i=1}^{N}\alpha_i y^i = 0$$

Dual Problem

If we substitute $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i$ to $\mathcal{L}$, we have

$$
\begin{aligned}
L(\boldsymbol{\alpha}) &= \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_{i=1}^{N} \alpha_i \{y^i(\mathbf{w}\cdot\mathbf{x}^i + b) - 1\} \\
&= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j < \mathbf{x}^i \cdot \mathbf{x}^j > - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j < \mathbf{x}^i \cdot \mathbf{x}^j > - b\sum_{i=1}^{N} \alpha_i y^i + \sum_{i=1}^{N} \alpha_i \\
&= \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j < \mathbf{x}^i \cdot \mathbf{x}^j >
\end{aligned}
$$

- Note that $\sum_{i=1}^{N} \alpha_i y^i = 0$
- This is a function of α_i only

Dual Problem

- The new objective function is in terms of α_i only
- It is known as the dual problem: if we know all α_i, we know **w**
- The original problem is known as the primal problem
- The objective function of the dual problem needs to be maximized!
- The dual problem is therefore:

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j < \mathbf{x}^i \cdot \mathbf{x}^j >$$

$$\text{subject to} \quad \alpha_i \geq 0, i = 1, \ldots, n, \qquad \sum_{i=1}^{N} \alpha_i y^i = 0$$

Properties of α_i when we introduce the Lagrange multipliers

The result when we differentiate the original Lagrangian w.r.t. b

- Note that there is only one constraint as against N in the original formulation
- Less number of variables

Dual Problem

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y^i y^j < \mathbf{x}^i \cdot \mathbf{x}^j >$$

$$\text{subject to} \quad \alpha_i \geq 0, i = 1, \ldots, n, \qquad \sum_{i=1}^{N} \alpha_i y^i = 0$$

- This is also quadratic programming (QP) problem
  - A global maximum of α_i can always be found
- **w** can be recovered by $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i$
- b can also be recovered as well (wait for a bit)

Characteristics of the Solution

- Many of the α_i are zero
  - **w** is a linear combination of only a small number of data points
- In fact, optimization theory requires that the solution to satisfy the following KKT conditions:

$$\alpha_i \geq 0, i = 1, \ldots, n,$$

$$y^i \left(\sum_{j=1}^{N} \alpha_i y^j < \mathbf{x}^j \cdot \mathbf{x}^i > + \mathrm{b}\right) \geq 1$$

Functional margin ≥ 1

$$\alpha_i \{ y^i \left(\sum_{j=1}^{N} \alpha_i y^j < \mathbf{x}^j \cdot \mathbf{x}^i > + \mathrm{b}\right) - 1 \} = 0$$

α_i is nonzero only when functional margin = 1

- $\mathbf{x}_i$ with non-zero α_i are called support vectors (SV)
  - The decision boundary is determined only by the SV
  - Let t_j (j=1, ..., s) be the indices of the s support vectors. We can write

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y^{t_j} \mathbf{x}^{t_j}$$

Solve for b

- Note that we know that for support vectors the functional margin = 1
- We can use this information to solve for b
- We can use any support vector to achieve this

$$y^i \left(\sum_{j=1}^{s} \alpha_{t_j} y^{t_j} < \mathbf{x}^{t_j} \cdot \mathbf{x}^i > + \mathrm{b}\right) = 1$$

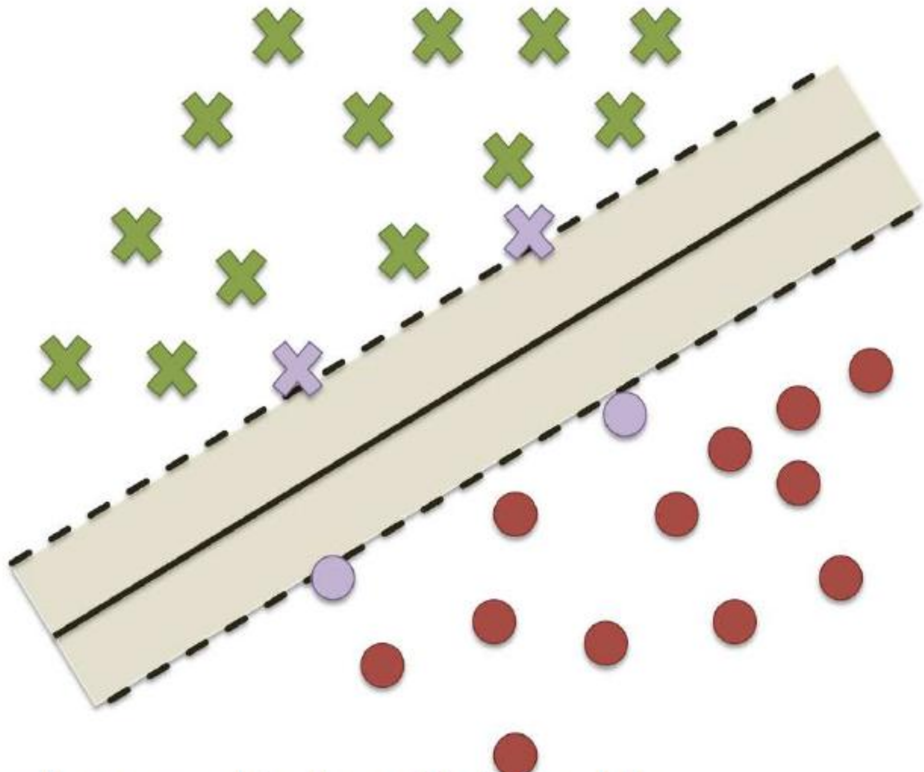
- A numerically more stable solution is to use all support vectors (details in the book)

Classifying new examples

- For classifying with a new input **z**
  - Compute $\mathbf{w}^T\mathbf{x}+b=\sum_{j=1}^{s}\alpha_{t_j}y^{t_j}<\mathbf{x}^{t_j}\cdot\mathbf{x}>+b$ and classify **z** as positive if the sum is positive, and negative otherwise
  - Note: **w** need not be formed explicitly, rather we can classify z by taking a weighted sum of the inner products with the support vectors
    (useful when we generalize from inner product to kernel functions later)

Support vectors

Only points, $\mathbf{x}_i$, that lie on the supporting hyperplanes have $\alpha_i > 0$. These are called the support vectors. *Complexity of the solution only depends on the number of* support vectors.

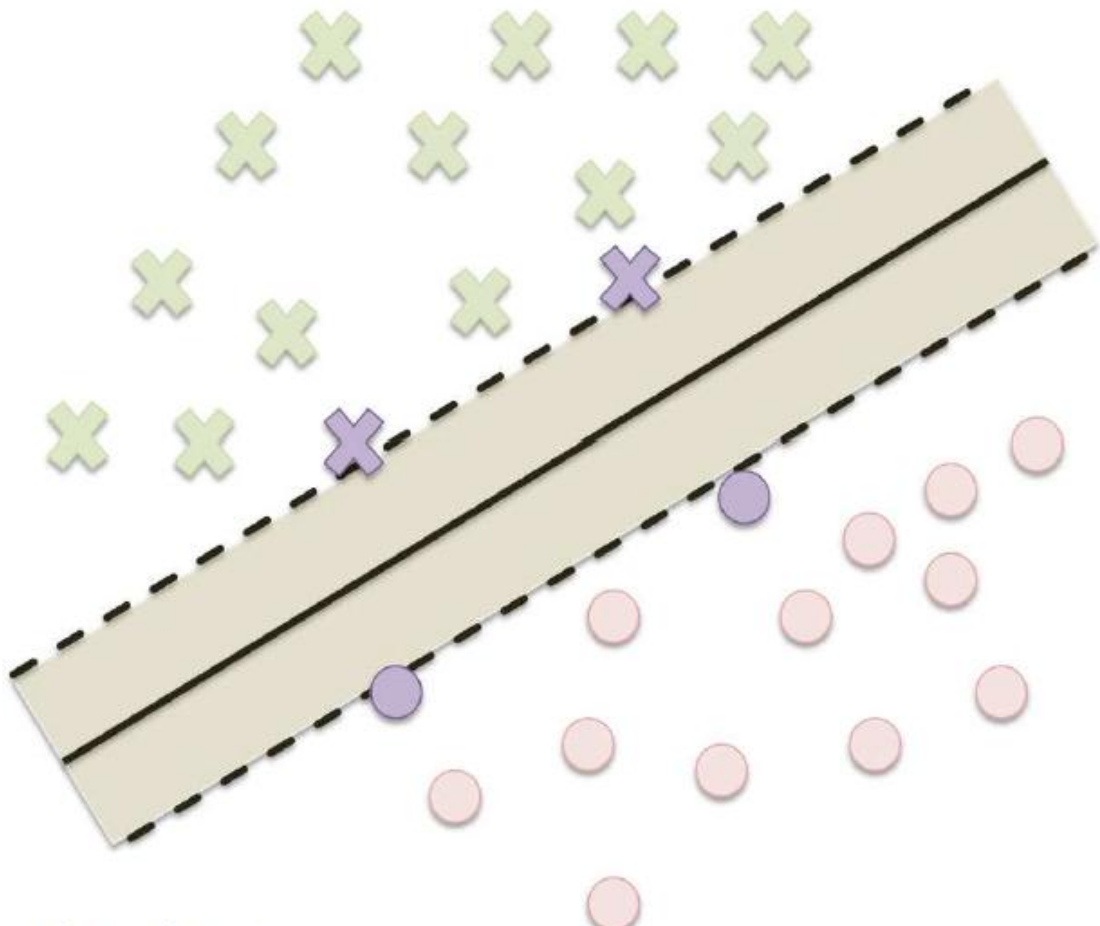


Recall that **w** is a linear combination of training data

$$\mathbf{w} = \sum_{i=1}^{N} y_i \, \alpha_i \, \mathbf{x}_i = \sum_{\text{support vectors}} y_i \, \alpha_i \, \mathbf{x}_i$$

Support Vectors

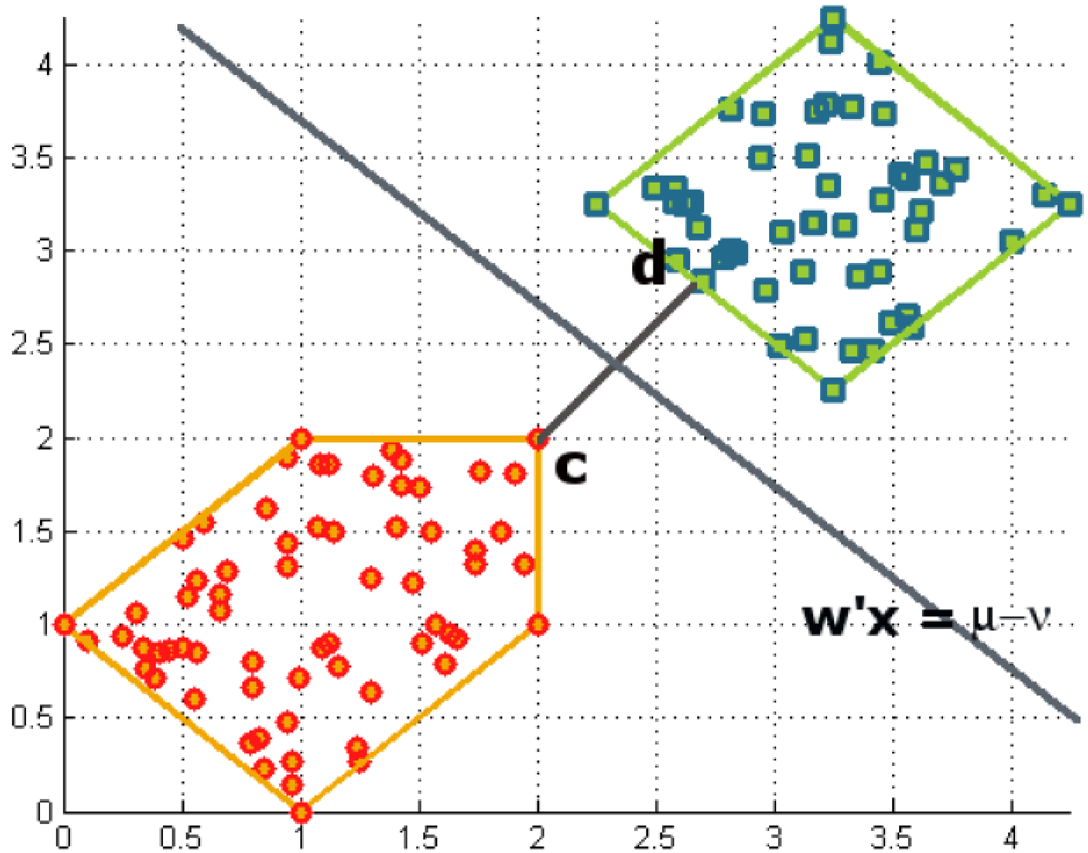
Learned model will not change if we delete all the data



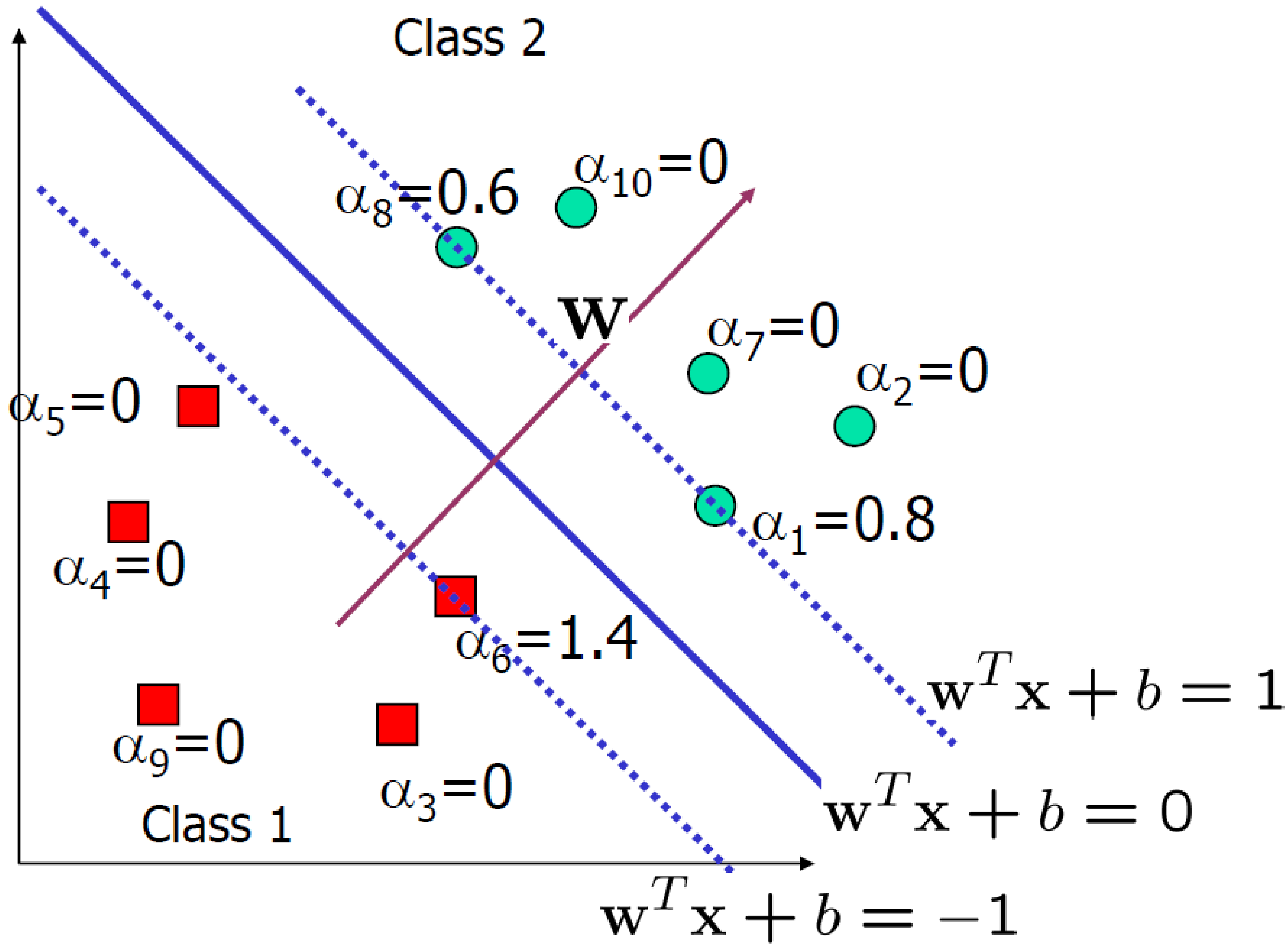
except the support vectors.

Geometric Perspective

Maximizing margin is equivalent to *maximizing distance between the two closet points on the convex hulls of the two sets*.



Geometric Perspective (2)



Summary

- We demonstrated that we prefer to have linear classifiers with large margin.

- We formulated the problem of finding the maximum margin linear classifier as a quadratic optimization problem

- This problem can be solved by solving its dual problem, and efficient QP algorithms are available.

- Problem solved?

- How about non-linear data? – Kernels

- How about noise? – Soft Margin SVMs