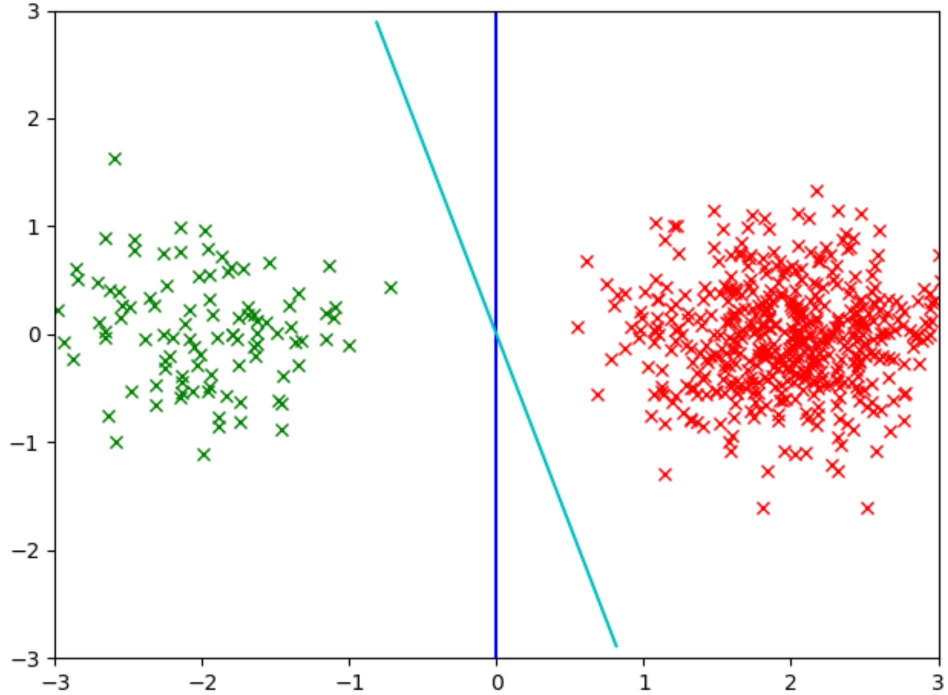
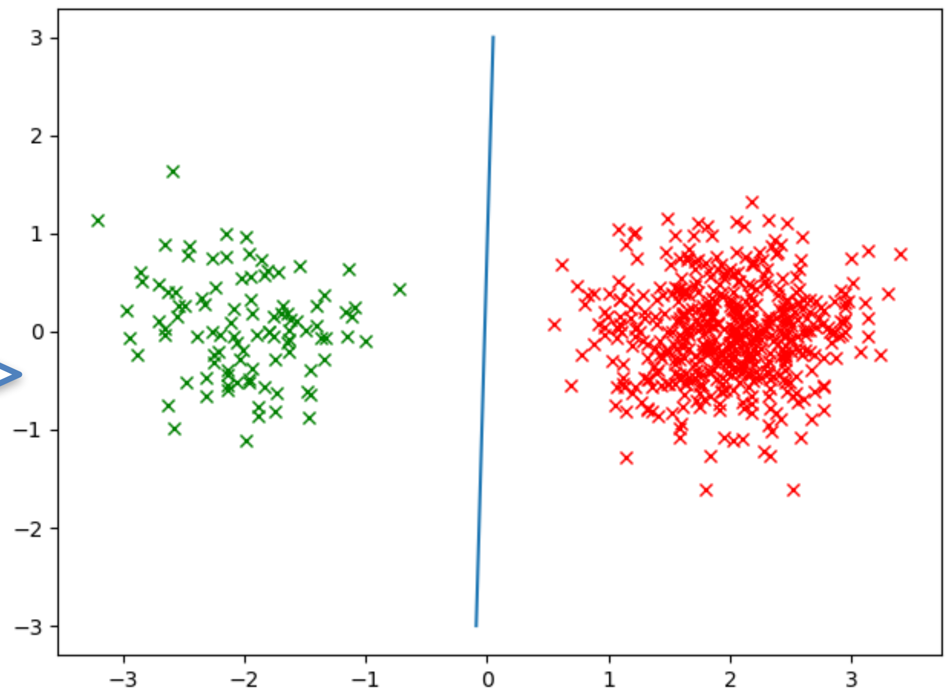
Maximizing the Margin

Pascal Fua
IC-CVLab

Logistic Regression is Better than the Perceptron



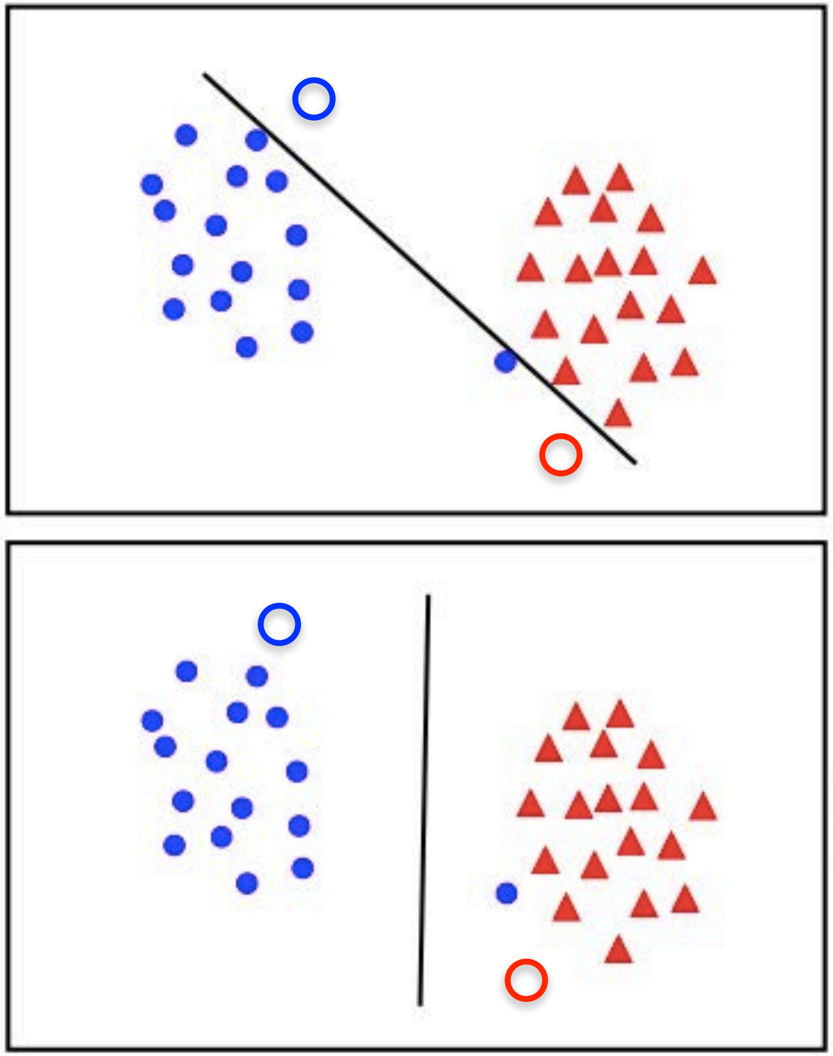
Perceptron



Logistic

But ….

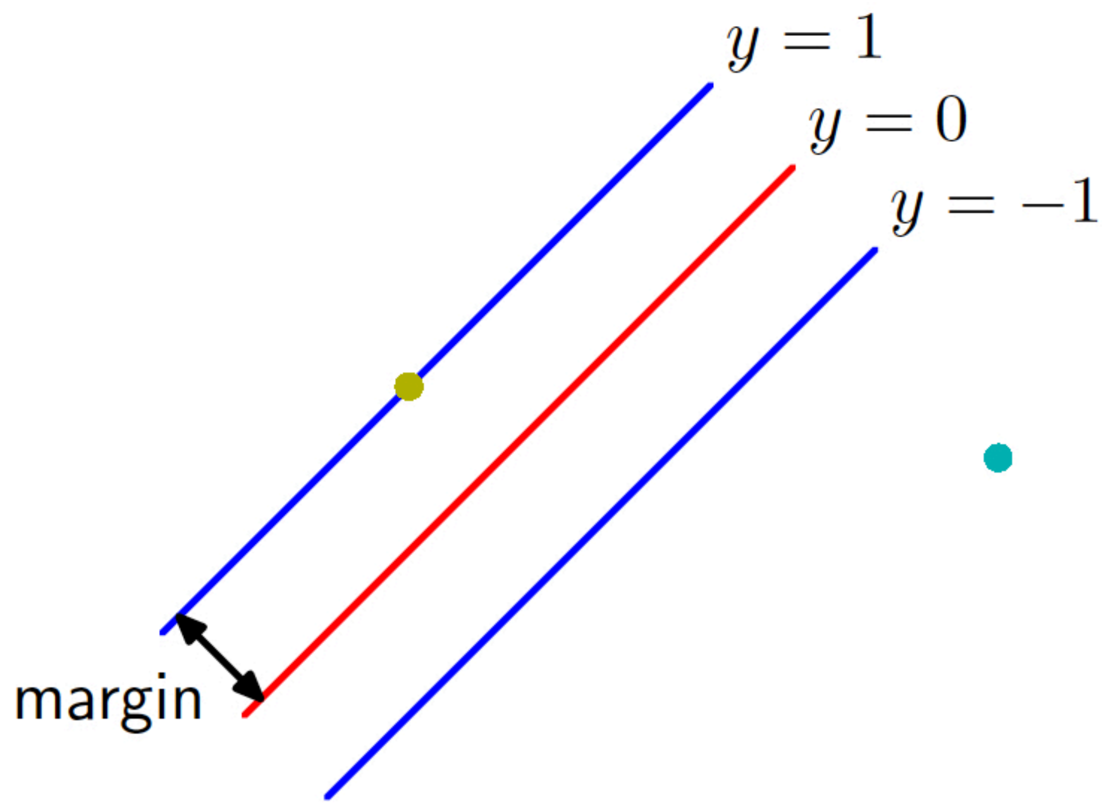
Outliers Can Cause Problems



- Logistic regression tries to minimize the error-rate at training time.
- Can result in poor classification rates at test time.

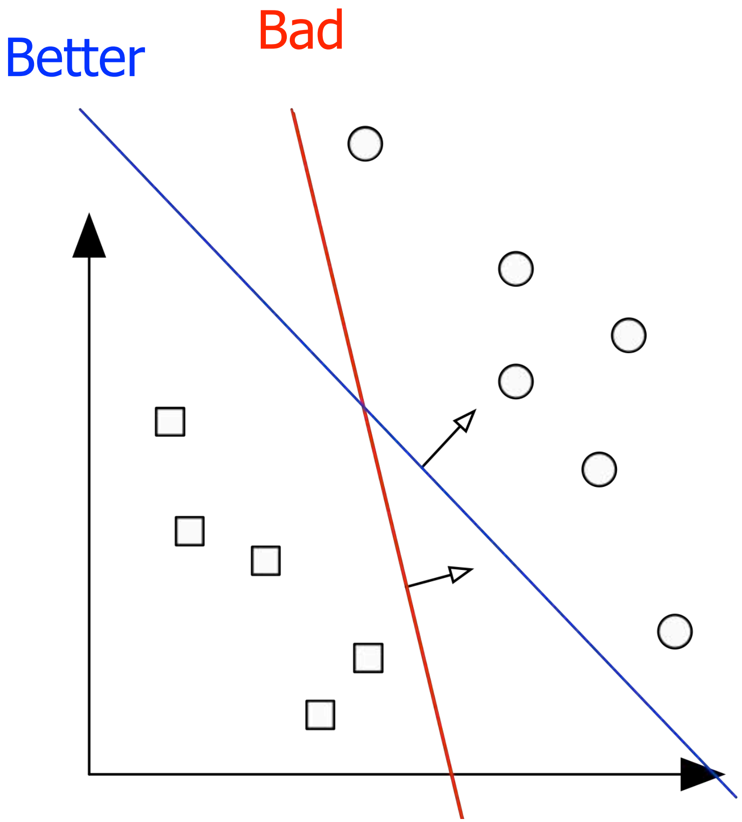
—> Sometimes, we should accept to misclassify a few training samples.

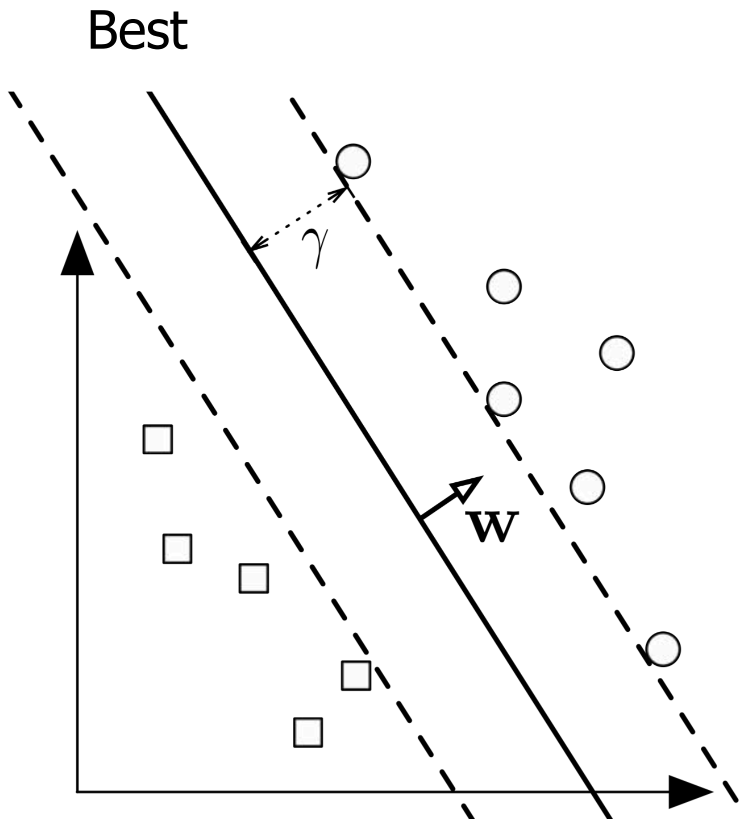
Margin



The orthogonal distance between the decision boundary and the nearest sample is called the **margin**.

Maximizing the Margin

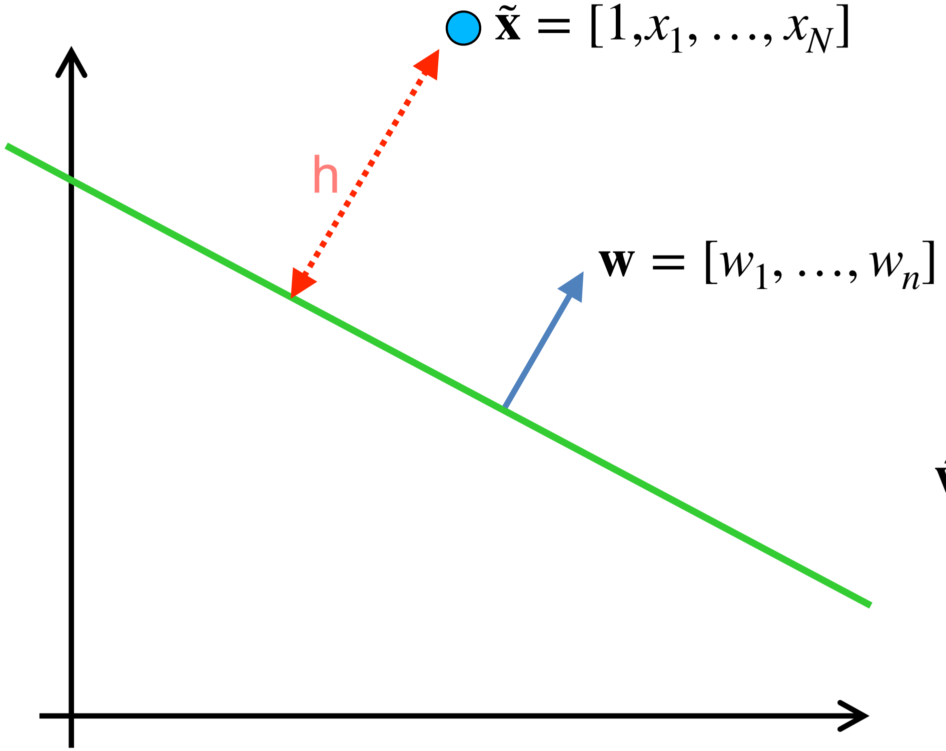




- The larger the margin, the better!
- The logistic regression does not guarantee the largest.

How do we maximize it?

Reminder: Signed Distance



h=0: Point is on the decision boundary.
h>0: Point on one side.
h<0: Point on the other side.

$\tilde{\mathbf{w}} = [w_0, w_1, \ldots, w_n]$ with $\sum_{i=1}^{N} w_i^2 = 1$

Hyperplane: $\mathbf{x} \in R^N, \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}} = 0$, with $\tilde{\mathbf{x}} = [1 \,|\, \mathbf{x}]$.

Signed distance: $\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}$, with $\tilde{\mathbf{w}} = [w_0 \,|\, \mathbf{w}]$ and $||\mathbf{w}|| = 1$.

Binary Classification in N Dimensions

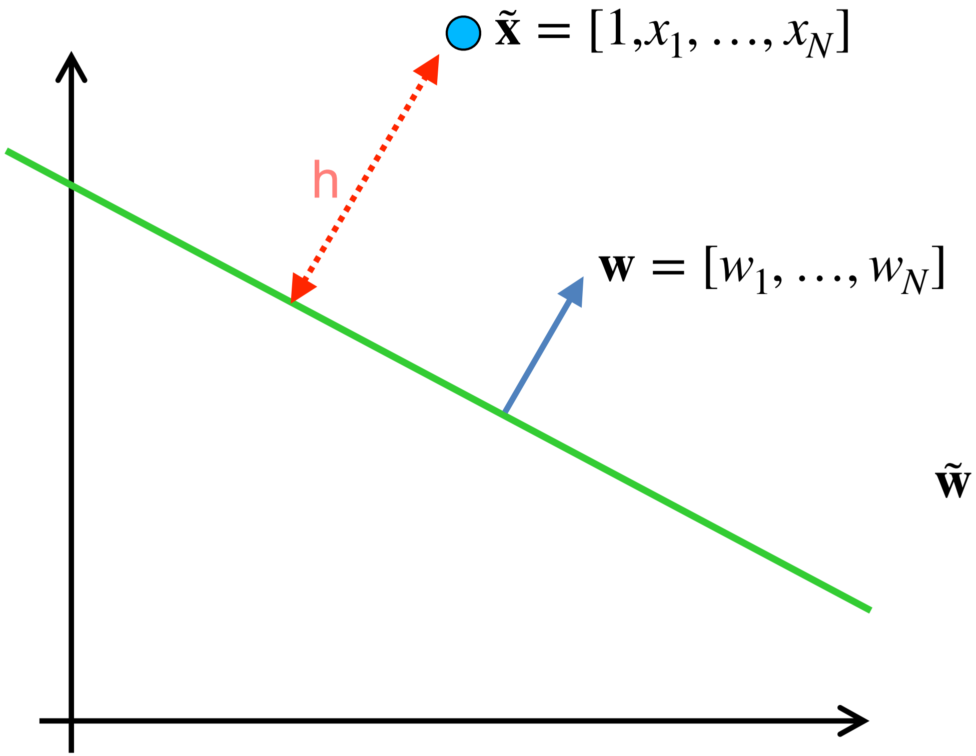
Hyperplane: $\mathbf{x} \in R^N, \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}} = 0$, with $\tilde{\mathbf{x}} = [1 \,|\, \mathbf{x}]$.

Signed distance: $\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}$, with $\tilde{\mathbf{w}} = [w_0 \,|\, \mathbf{w}]$ and $||\mathbf{w}|| = 1$.

Problem statement: Find $\tilde{\mathbf{w}}$ such that

- for all or most positive samples $\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}} > 0$,
- for all or most negative samples $\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}} < 0$.

Reformulating the Signed Distance Again



h=0: Point is on the decision boundary.
h>0: Point on one side.
h<0: Point on the other side.

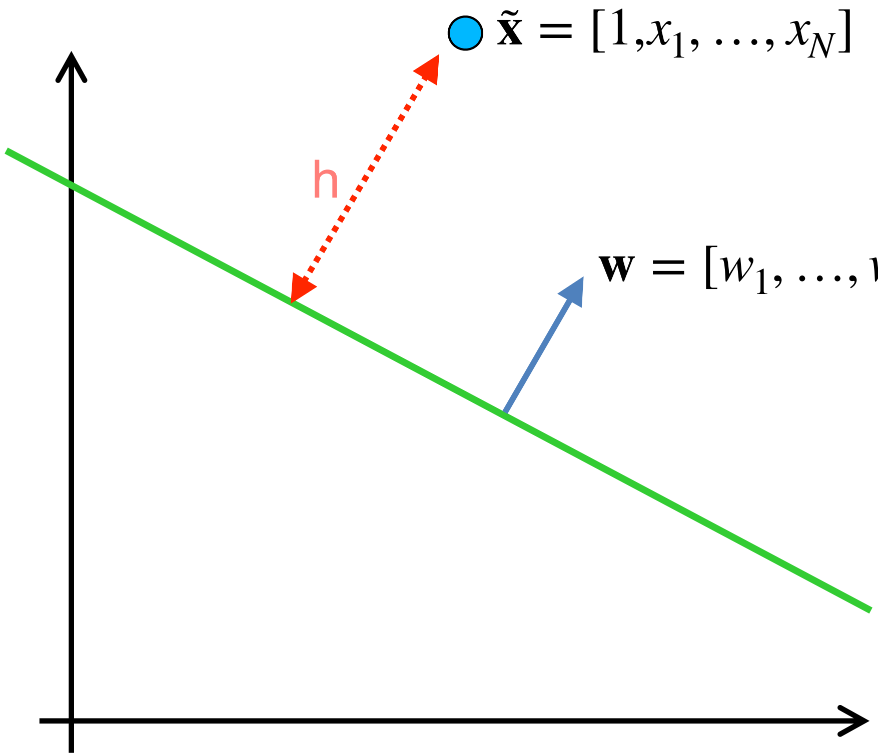
$\tilde{\mathbf{w}} = [w_0, w_1, \ldots, w_N]$ with $\cancel{\sum_{i=1}^{N} w_i^2 = 1}$

Hyperplane: $\mathbf{x} \in R^N, \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}} = 0$, with $\tilde{\mathbf{x}} = [1 \,|\, \mathbf{x}]$.

Signed distance: $\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}$, with $\tilde{\mathbf{w}} = [1 \,|\, \mathbf{w}]$ and $||\mathbf{w}|| = 1$.

EPFL

Reformulated Signed Distance



h=0: Point is on the decision boundary.
h>0: Point on one side.
h<0: Point on the other side.

$$\tilde{\mathbf{w}} = [w_0 | \mathbf{w}] \in R^{N+1}$$

$$\tilde{\mathbf{w}}' = \frac{\tilde{\mathbf{w}}}{||\mathbf{w}||} = [\frac{w_0}{||\mathbf{w}||} | \frac{\mathbf{w}}{||\mathbf{w}||}]$$

Hyperplane: $\mathbf{x} \in R^N, \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}} = 0$, with $\tilde{\mathbf{x}} = [1 | \mathbf{x}]$.

Signed distance: $\tilde{\mathbf{w}}' \cdot \tilde{\mathbf{x}} = \boxed{\frac{\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}}{||\mathbf{w}||}, \forall \tilde{\mathbf{w}} \in R^{N+1}.}$

Maximum Margin Classifier

- Given a training set $\{(\mathbf{x}_n, t_n)_{1 \leq n \leq N}\}$ with $t_n \in \{-1,1\}$ and solution such that all the points are correctly classified, we have

$$\forall n, \quad t_n(\tilde{\mathbf{w}}_n \cdot \tilde{\mathbf{x}}_n) > = 0 \, .$$

- We can write the **unsigned** distance to the decision boundary as

$$d_n = t_n \frac{(\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_n)}{||\mathbf{w}||}$$

—> A maximum margin classifier aims to maximize this distance for the point closest to the boundary, that, is maximize the minimum such distance.

$$\tilde{\mathbf{w}}^* = \mathrm{argmax}_{\tilde{\mathbf{w}}} \min_n \left(\frac{t_n \cdot (\tilde{\mathbf{w}} \cdot \mathbf{x}_n)}{\|\mathbf{w}\|} \right)$$

Maximum Margin Classifier

$$\tilde{\mathbf{w}}^* = \text{argmax}_{\tilde{\mathbf{w}}} \min_n \left(\frac{t_n \cdot (\tilde{\mathbf{w}} \cdot \mathbf{x}_n)}{||\mathbf{w}||} \right)$$

- Unfortunately, this is a difficult optimization problem to solve.
- We will convert it into an equivalent, but easier to solve, problem.

Maximum Margin Classifier

- The signed distance is invariant to a scaling of $\tilde{\mathbf{w}}$:

$$\tilde{\mathbf{w}} \rightarrow \lambda\tilde{\mathbf{w}} : d_n = t_n \frac{(\lambda\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_n)}{||\lambda\mathbf{w}||} = t_n \frac{(\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_n)}{||\mathbf{w}||} .$$

- We can choose λ so that for the point m closest to the boundary, we have

$$t_m \cdot (\tilde{\mathbf{w}} \cdot \mathbf{x}_m) = 1 .$$

- For all points we therefore have

$$t_n \cdot (\tilde{\mathbf{w}} \cdot \mathbf{x}_n) \geq 1 ,$$

and the equality holds for at least one point.

Linear Support Vector Machine

$$\forall n, \quad t_n(\tilde{\mathbf{w}} \cdot \mathbf{x}_n) \geq 1$$
$$\exists n \quad t_n(\tilde{\mathbf{w}} \cdot \mathbf{x}_n) = 1$$
$$\Rightarrow min_n d_n = min_n \frac{t_n(\tilde{\mathbf{w}} \cdot \mathbf{x}_n)}{||\mathbf{w}||} = \frac{1}{||\mathbf{w}||}$$

- To maximize the margin, we only need to maximize $1/||\mathbf{w}||$.
- This is equivalent to minimizing $\frac{1}{2}||\mathbf{w}||^2$.
- We can find max margin classifier as

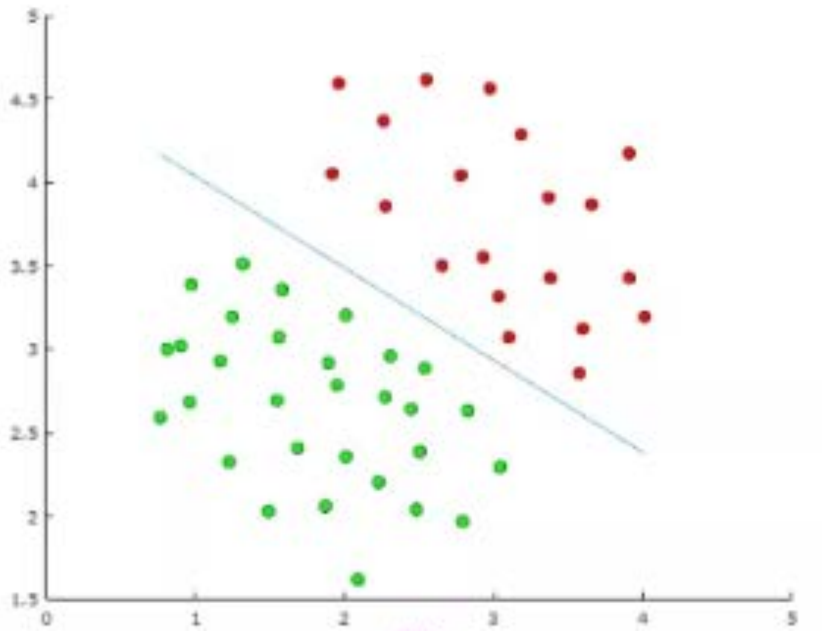
$$\mathbf{w}^* = argmin_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 \text{subject to } \forall n, \quad t_n \cdot (\tilde{\mathbf{w}} \cdot \mathbf{x}_n) \geq 1$$

- This is a quadratic program, which is a **convex** problem.

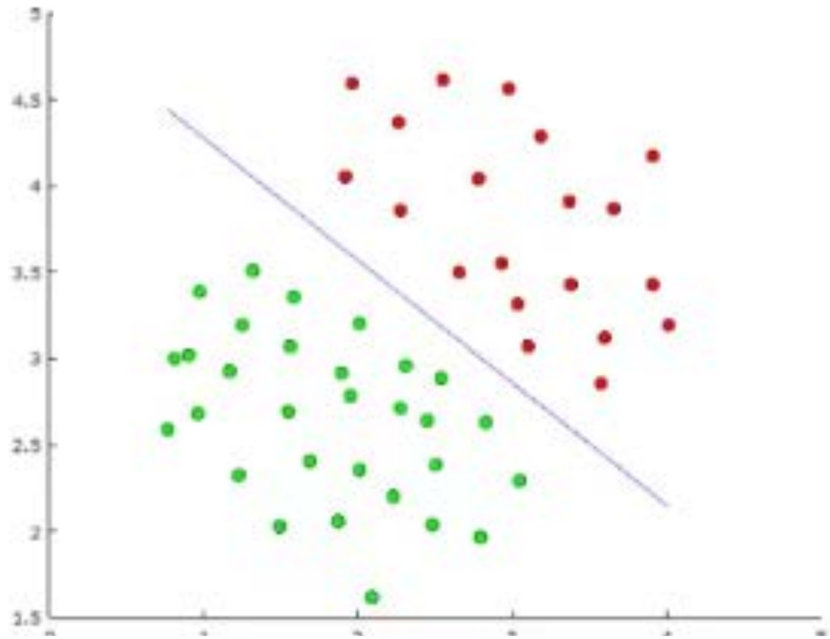
—> It can be solved to optimality.

EPFL

LR vs Linear SVM



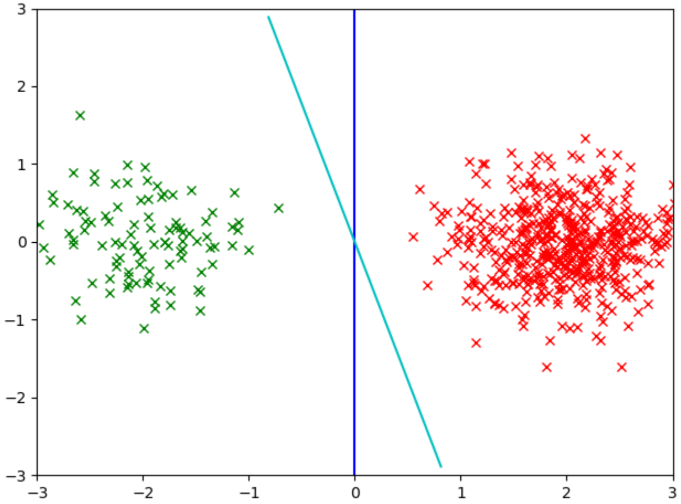
Logistic regression



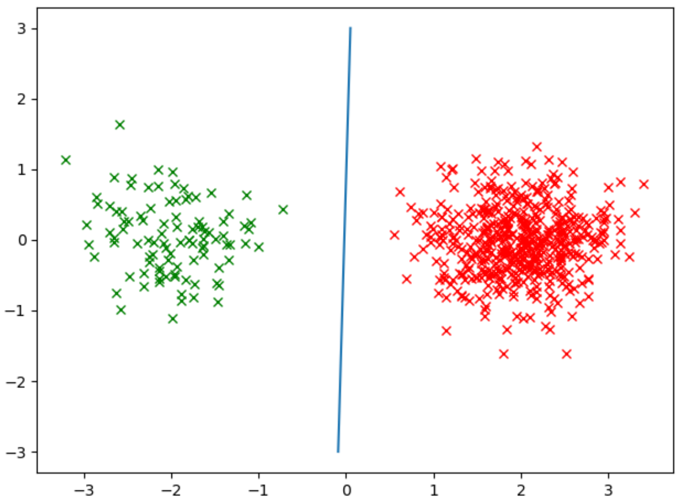
Linear SVM

- The LR decision boundary can come close to some of the training examples.
- The SVM tries to prevent that.

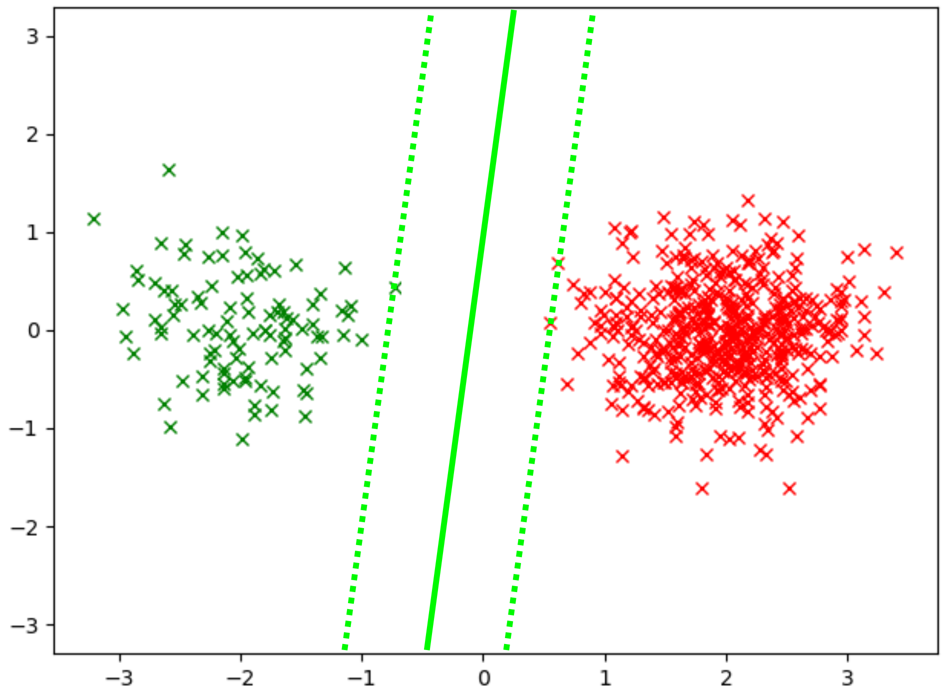
From Perceptron and LR to Linear SVM



Perceptron



Logistic Regression



Linear SVM

Are we done yet?

No!

Maximum Margin Classifier

Rarely achievable in practice.

- Given a training set $\{(\mathbf{x}_n, t_n)_{1 \leq n \leq N}\}$ with $t_n \in \{-1,1\}$ and solution such that all the points are correctly classified, we have

$$\boxed{\forall n, \quad t_n(\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_n) >= 1 \,.}$$

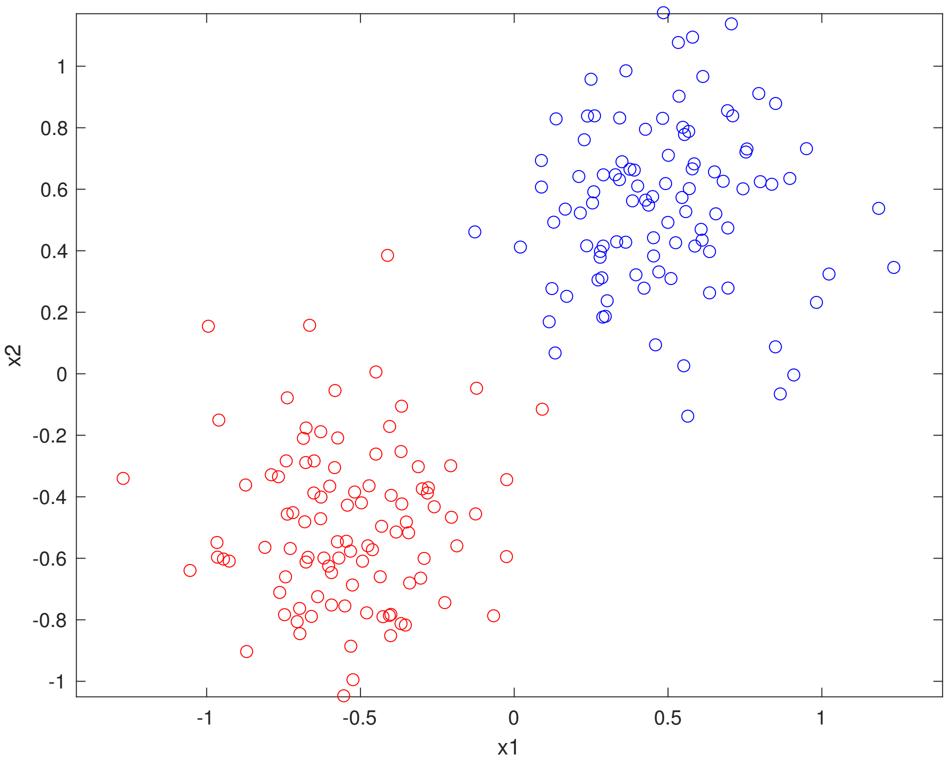
- We can write the **unsigned** distance to the decision boundary as

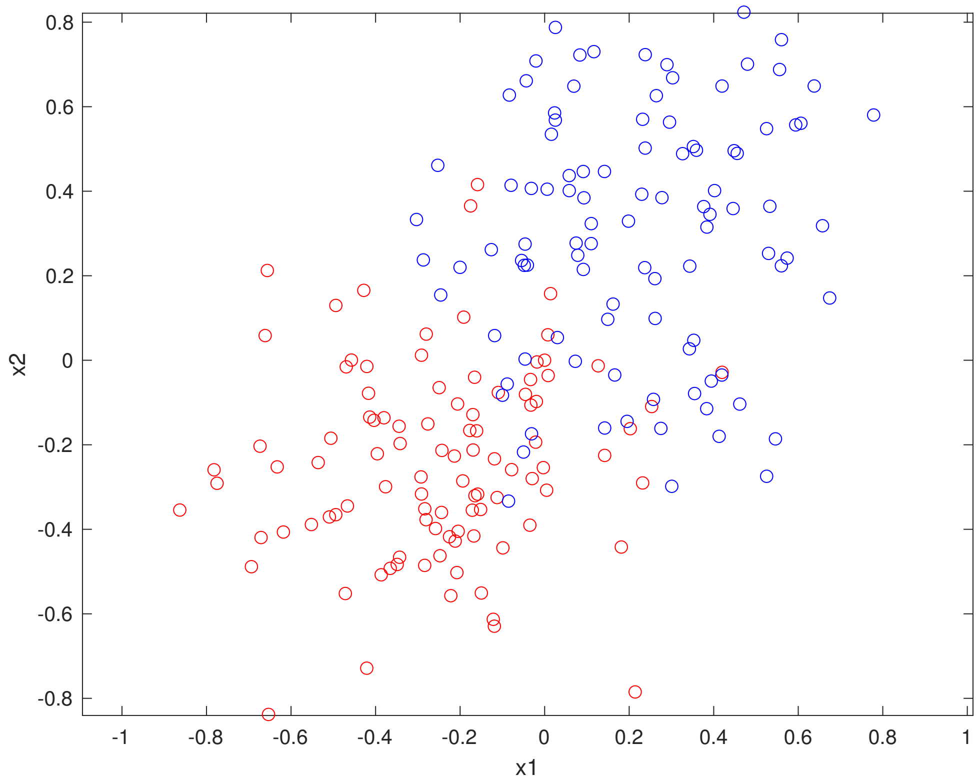
$$d_n = t_n \frac{(\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_n)}{||\mathbf{w}||}$$

—> A maximum margin classifier aims to maximize this distance for the point closest to the boundary, that, is maximize the minimum such distance.

$$\tilde{\mathbf{w}}^* = \mathrm{argmax}_{\tilde{\mathbf{w}}} \min_n \left(\frac{t_n \cdot (\tilde{\mathbf{w}} \cdot \mathbf{x}_n)}{\|\mathbf{w}\|} \right)$$

Overlapping Classes





The data rarely looks like this. It generally looks like that.

—> Must account for the fact that not all training samples can be correctly classified!

Relaxing the Constraints

- The original problem

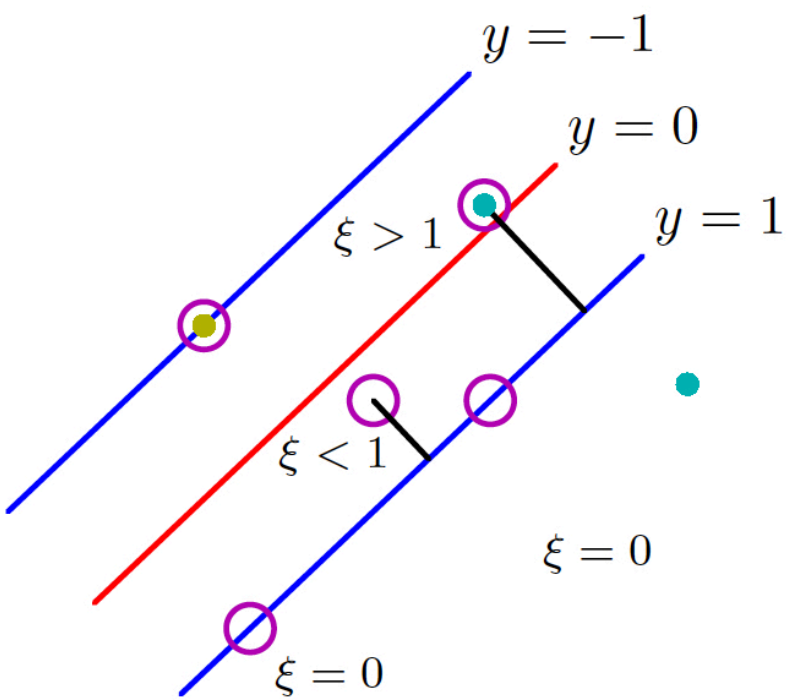
$$\mathbf{w}^* = argmin_{\mathbf{w}} \frac{1}{2} ||\mathbf{w}||^2 \text{subject to} \forall n, \quad t_n \cdot (\tilde{\mathbf{w}} \cdot \mathbf{x}_n) \geq 1,$$

  cannot be satisfied.

- We must allow some of the constraints to violated, but as few as possible.

Slack Variables

- We introduce an additional slack variable ξ_n for each sample.
- We rewrite the constraints as $t_n \cdot (\tilde{\mathbf{w}} \cdot \mathbf{x}_n) \geq 1 - \xi_n$.
- $\xi_i \geq 0$ weakens the original constraints.



- If $0 < \xi_n \leq 1$, sample n lies inside the margin, but is still correctly classified
- If $\xi_n \geq 1$, then sample i is misclassified

Naive Formulation

$$\mathbf{w}^* = argmin_{\mathbf{w}} \frac{1}{2} ||\mathbf{w}||^2$$

$$\text{subject to } \forall n, \quad t_n \cdot (\tilde{\mathbf{w}} \cdot \mathbf{x}_n) \geq 1 - \xi_n \text{ and } \xi_n \geq 0$$

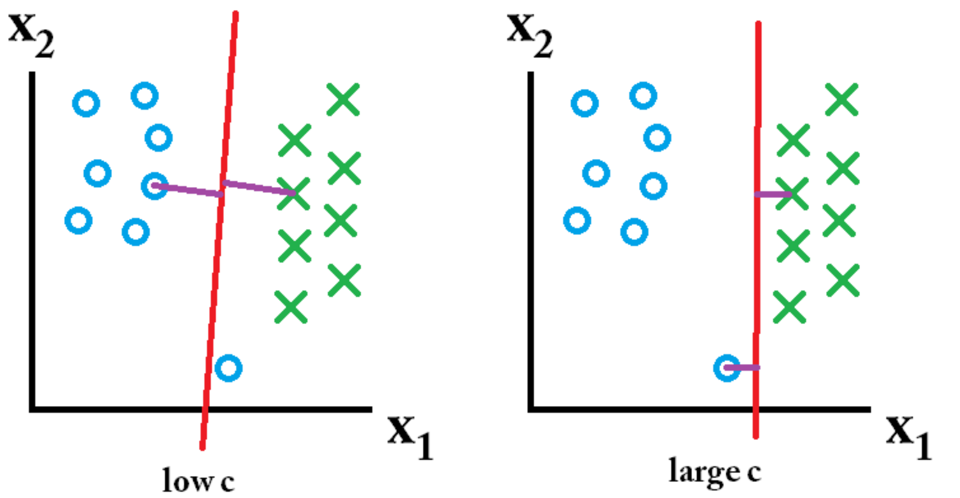
- This would simply allow the model to violate all the original constraints at no cost.
- This would result in a useless classifier.

Improved Formulation

$$\mathbf{w}^* = argmin_{(\mathbf{w},\{\xi_\mathbf{n}\})} \frac{1}{2} ||\mathbf{w}||^2 + C \sum_{n=1}^{N} \xi_n,$$

$$\text{subject to } \forall n, \quad t_n \cdot (\tilde{\mathbf{w}} \cdot \mathbf{x}_n) \geq 1 - \xi_n \text{ and } \xi_n \geq 0.$$

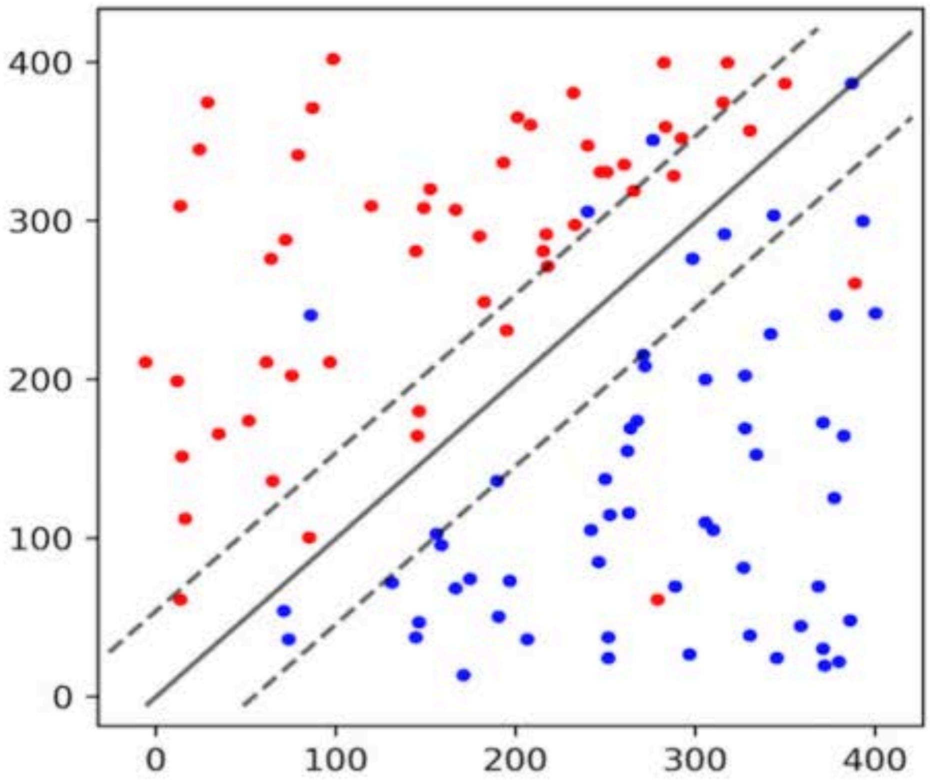
- C is constant that controls how costly constraint violations are.
- The problem is still convex.

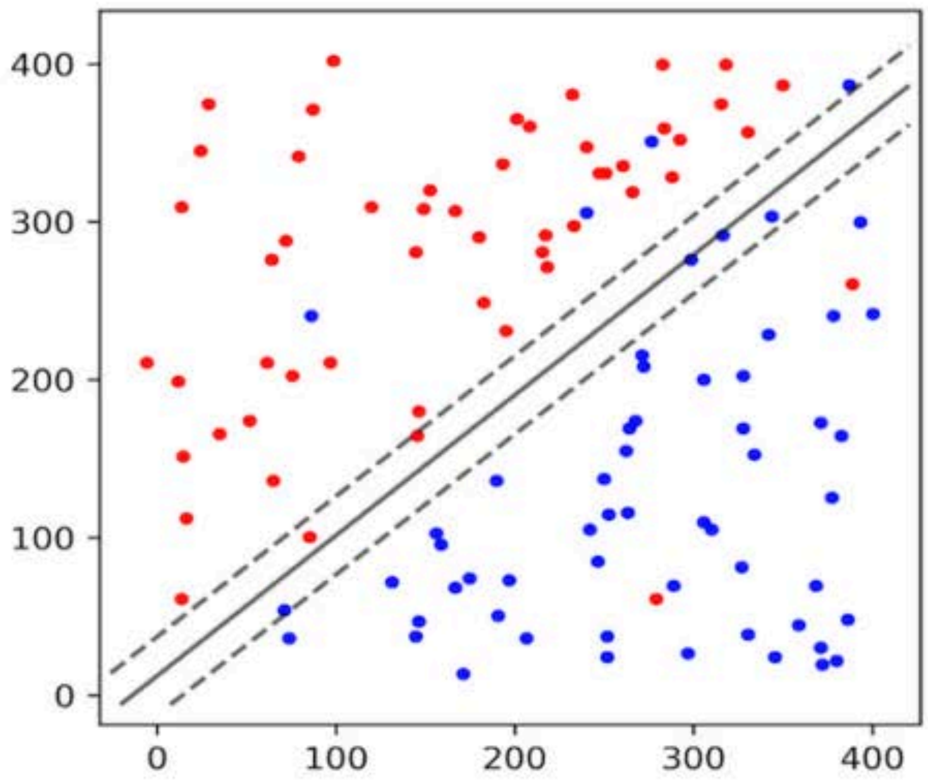


Large margin but potential misclassifications.

Smaller margin but fewer misclassifications.

Choosing the C Parameter



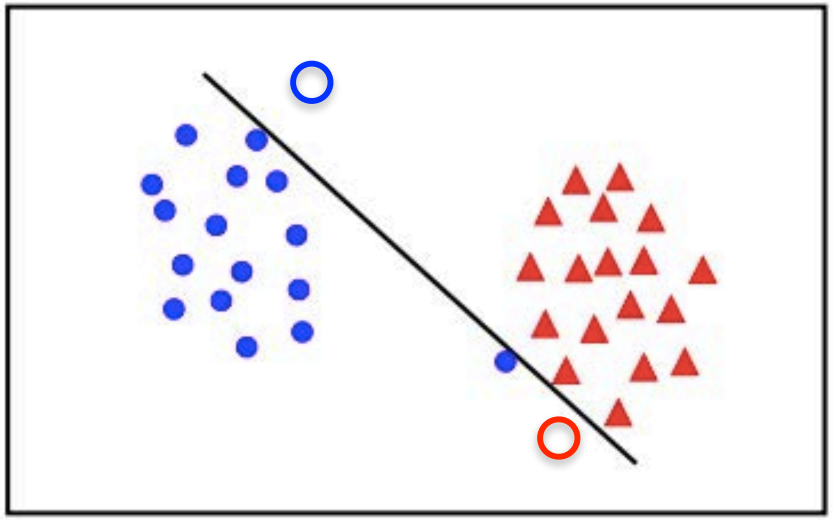


C=1:
- Large margin.
- Many training samples misclassified.

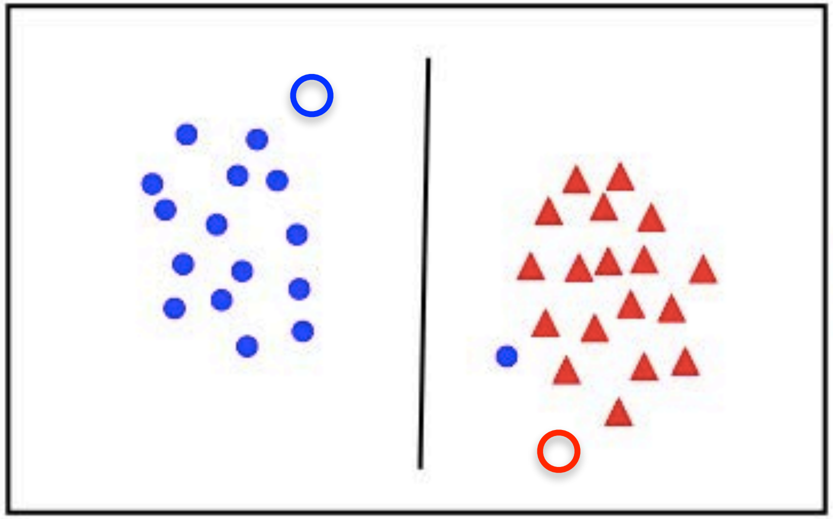
C=100:
- Small margin.
- Few training samples misclassified.

Which is best?
- It depends.
- Must use cross-validation, as we did for k-Means.

Linear SVM Trade Off



- The points can be linearly separated but the margin is still very small.
- At test time the two circles will be misclassified.



- The margin is much larger but one training example is misclassified.
- At test time the two circles will be classified correctly.

—> Tradeoff between the number of mistakes on the training data and the margin.

Support Vector Machines