Maximizing the Margin

Pascal Fua
IC-CVLab
Logistic Regression is Better than the Perceptron

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.

—> Must sometime accept to misclassify a few training samples.
The orthogonal distance between the decision boundary and the nearest sample is called the **margin**.
• The larger the margin, the better!
• The logistic regression does not guarantee a large one.

How do we maximize it?
Hyperplane: $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{w} \cdot \mathbf{x} = 0$, with $\mathbf{x} = [1 \mid \mathbf{x}]$.

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

$h=0$: Point is on the decision boundary. 
$h>0$: Point on one side. 
$h<0$: Point on the other side.
Binary Classification in N Dimensions

Hyperplane: \( x \in R^N, \tilde{w} \cdot \tilde{x} = 0 \), with \( \tilde{x} = [1 \mid x] \).

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

Problem statement: Find \( \tilde{w} \) such that
- for all or most positive samples \( \tilde{w} \cdot \tilde{x} > 0 \),
- for all or most negative samples \( \tilde{w} \cdot \tilde{x} < 0 \).
Reformulating the Signed Distance Again

\[ \mathbf{w} = [w_1, \ldots, w_n] \]

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

\[ \tilde{\mathbf{x}} = [1, x_1, \ldots, x_N] \]

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

Hyperplane: \( \mathbf{x} \in \mathbb{R}^N, \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}} = 0 \), with \( \tilde{\mathbf{x}} = [1 \mid \mathbf{x}] \).

Signed distance: \( \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}} \), with \( \tilde{\mathbf{w}} = [1 \mid \mathbf{w}] \) and \( ||\mathbf{w}|| = 1 \).
Reformulated Signed Distance

$w = [w_1, \ldots, w_n]$

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

$\tilde{x} = [1, x_1, \ldots, x_N]$

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

$\tilde{w}' = \frac{\tilde{w}}{||w||} = \left[ \frac{w_0}{||w||} \right.$

$\left. \frac{w}{||w||} \right]$

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

Signed distance: $\tilde{w}' \cdot \tilde{x} = \frac{\tilde{w} \cdot \tilde{x}}{||w||}$, $\forall \tilde{w} \in \mathbb{R}^{N+1}$. 
We are going to use this to find a classifier whose decision boundary is as far as possible from all the points.
Maximum Margin Classifier

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

\[
\forall n, \quad t_n (\tilde{w}_n \cdot \tilde{x}_n) \geq 0 .
\]

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

\[
d_n = t_n \frac{(\tilde{w} \cdot \tilde{x}_n)}{\|w\|}
\]

\( \text{-->} \) A maximum margin classifier aims to maximize this distance for the point closest to the boundary, that, is maximize the minimum such distance.

\[
\tilde{w}^* = \text{argmax}_{\tilde{w}} \min_n \left( \frac{t_n \cdot (\tilde{w} \cdot x_n)}{\|w\|} \right)
\]
Maximum Margin Classifier

\[ \tilde{w}^* = \arg\max_w \min_n \left( \frac{t_n \cdot (\tilde{w} \cdot x_n)}{\|w\|} \right) \]

- 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 $\tilde{\mathbf{w}}$:

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

• We can choose $\lambda$ 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.
To maximize the margin, we only need to maximize $t_n(\tilde{\mathbf{w}} \cdot \mathbf{x}_n)$.

This is equivalent to minimizing $||\mathbf{w}||$.

We can find the max margin classifier as

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

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

$\Rightarrow \min_n d_n = \min_n \frac{t_n(\tilde{\mathbf{w}} \cdot \mathbf{x}_n)}{||\mathbf{w}||} = \frac{1}{||\mathbf{w}||}$
LR vs 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

Are we done yet?

No!

Logistic Regression

Linear SVM
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

\[
\forall n, \quad t_n (\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_n) \geq 1.
\]

• We can write the unsigned distance to the decision boundary as

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

\( \rightarrow \) A maximum margin classifier aims to maximize this distance for the point closest to the boundary, that is, maximize the minimum such distance.

\[
\tilde{\mathbf{w}}^* = \arg\max_{\tilde{w}} \min_n \left( \frac{t_n \cdot (\tilde{\mathbf{w}} \cdot \mathbf{x}_n)}{||\mathbf{w}||} \right)
\]

Rarely achievable in practice.
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

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

cannot be satisfied.

• We must allow some of the constraints to be violated, but as few as possible.
Slack Variables

- We introduce an additional slack variable $\xi_n$ 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

\[ w^* = \min_w \frac{1}{2} ||w||^2 \]

subject to \( \forall n, \quad t_n \cdot (\tilde{w} \cdot x_n) \geq 1 - \xi_n \) 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

\[ w^* = \min_{(w, \{\xi_n\})} \frac{1}{2} ||w||^2 + C \sum_{n=1}^{N} \xi_n, \]

subject to \( \forall n, \quad t_n \cdot (\tilde{w} \cdot x_n) \geq 1 - \xi_n \) and \( \xi_n \geq 0. \)

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

http://www.cristiandima.com/basics-of-support-vector-machines/
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.
Optimal vs Best

- 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