Margin violations and misclassifications

\[ f(x) = \mathbf{w}^\top \mathbf{x} + b \]

Penalize points according to their distance from the margin.
Loss functions

- SVM uses “hinge” loss $\max (0, 1 - yf(x))$
- in contrast to the 0-1 loss

$f(x) = w^T x + b$
"Soft" margin problem

The optimization problem becomes

\[ f(x) = w^\top x + b \]

\[
\min_{w \in \mathbb{R}^d} \|w\|^2 + C \sum_{i}^{N} \max(0, 1 - y_i f(x_i))
\]

- \( C \) is a regularization parameter:
  - small \( C \) allows constraints to be easily ignored \( \rightarrow \) large margin
  - large \( C \) makes constraints hard to ignore \( \rightarrow \) narrow margin
  - \( C = \infty \) enforces all constraints: hard margin

- Note, there is only one parameter, \( C \).
Loss function

\[
\min_{\mathbf{w} \subset \mathbb{R}^d} \|\mathbf{w}\|^2 + C \sum_{i}^{N} \max(0, 1 - y_i f(x_i))
\]

Points are in three categories:

1. \(y_i f(x_i) > 1\)
   - Point is outside margin.
   - No contribution to loss

2. \(y_i f(x_i) = 1\)
   - Point is on margin.
   - No contribution to loss.
   - As in hard margin case.

3. \(y_i f(x_i) < 1\)
   - Point violates margin constraint.
   - Contributes to loss
• data is linearly separable
• but only with a narrow margin
C = Infinity    hard margin
C = 10  soft margin
• Does this cost function have a unique solution?

• Does the solution depend on the starting point of an iterative optimization algorithm (such as gradient descent)?

If the cost function is **convex**, then a locally optimal point is globally optimal (provided the optimization is over a convex set, which it is in our case).
Convex functions

\[ D \text{ – a domain in } \mathbb{R}^n. \]

A convex function \( f : D \to \mathbb{R} \) is one that satisfies, for any \( x_0 \) and \( x_1 \) in \( D \):

\[ f((1 - \alpha)x_0 + \alpha x_1) \leq (1 - \alpha)f(x_0) + \alpha f(x_1). \]

Line joining \((x_0, f(x_0))\) and \((x_1, f(x_1))\) lies above the function graph.
Convex function examples

A non-negative sum of convex functions is convex
SVM

\[
\min_{w \in \mathbb{R}^d} \sum_{i=1}^{N} \max (0, 1 - y_i f(x_i)) + \|w\|^2 \quad \text{convex}
\]
Summary: Learning by optimizing cost functions

\[
\min_w \sum_{i=1}^N l(f(x_i, w), y_i) + \lambda R(w)
\]

- loss function
- regularization

• We have seen – “ridge” regression

  squared loss: \( (y_i - f(x_i, w))^2 \)  
  squared regularizer: \( \lambda \| w \|^2 \)

• Lasso regression

  squared loss: \( (y_i - f(x_i, w))^2 \)  
  lasso regularizer: \( \lambda \sum_j |w_j| \)

• SVM

  hinge loss: \( \max(0, 1 - y_i f(x_i, w)) \)  
  squared regularizer: \( \lambda \| w \|^2 \)
Logistic Regression Classifier
Overview

- Logistic regression is actually a classification method.

- LR introduces an extra non-linearity over a linear classifier, \( f(x) = w^T x + b \), by using a logistic (or sigmoid) function, \( \sigma() \).

- Here, choose binary classification to be represented by \( y_i \in \{0, 1\} \), rather than \( y_i \in \{1, -1\} \).

- The LR classifier is defined as

\[
\sigma(f'(x_i)) \begin{cases} 
\geq 0.5 & y_i = 1 \\
< 0.5 & y_i = 0
\end{cases}
\]

where \( \sigma(f(x)) = \frac{1}{1+e^{-f(x)}} \).
The logistic function or sigmoid function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- As $z$ goes from $-\infty$ to $\infty$, $\sigma(z)$ goes from 0 to 1, a “squashing function”.
- It has a “sigmoid” shape (i.e. S-like shape)
- $\sigma(0) = 0.5$, and if $z = w^\top x + b$ then $\left\| \frac{d\sigma(z)}{dx} \right\|_{z=0} = \frac{1}{4} \| w \|$
A sigmoid favours a larger margin of a step classifier. Though, need to control the gradient. How?
Learning

- Think of $\sigma(f(x))$ as the posterior probability that $y = 1$, i.e. $P(y = 1|x) = \sigma(f(x))$, where $f(x) = w^\top x + b$,

- Hence, if $\sigma(f(x)) > 0.5$ then class $y = 1$ is selected
Maximum Likelihood Estimation

Assume

\[ p(y = 1 | x; w) = \sigma(w^T x) \]
\[ p(y = 0 | x; w) = 1 - \sigma(w^T x) \]

write this more compactly as

\[ p(y | x; w) = \left( \sigma(w^T x) \right)^y \left( 1 - \sigma(w^T x) \right)^{1-y} \]

Then the likelihood (assuming data independence) is

\[ p(y | x; w) \sim \prod_{i}^{N} \left( \sigma(w^T x_i) \right)^{y_i} \left( 1 - \sigma(w^T x_i) \right)^{1-y_i} \]

and the negative log likelihood is

\[ L(w) = -\sum_{i}^{N} y_i \log \sigma(w^T x_i) + (1 - y_i) \log(1 - \sigma(w^T x_i)) \]

Note, this is the cross-entropy between the ground truth and predicted distributions.
Logistic Regression Loss function

Use notation \( y_i \in \{-1, 1\} \). Then for \( f(x) = w^\top x + b \)

\[
P(y = 1|x) = \sigma(f(x)) = \frac{1}{1 + e^{-f(x)}}
\]

\[
P(y = -1|x) = 1 - \sigma(f(x)) = \frac{1}{1 + e^{+f(x)}}
\]

So in both cases

\[
P(y_i|x_i) = \frac{1}{1 + e^{-y_if(x_i)}}
\]

Assuming independence, the likelihood is

\[
\prod_{i=1}^{N} \frac{1}{1 + e^{-y_if(x_i)}}
\]

and the negative log likelihood is

\[
= \sum_{i} \log \left(1 + e^{-y_if(x_i)}\right)
\]

which defines the loss function.
Comparison of SVM and LR cost functions

**SVM**

$$\min_{w \in \mathbb{R}^d} C \sum_{i}^{N} \max (0, 1 - y_i f(x_i)) + ||w||^2$$

**Logistic regression:**

$$\min_{w \in \mathbb{R}^d} \sum_{i}^{N} \log \left( 1 + e^{-y_i f(x_i)} \right) + \lambda ||w||^2$$

**Note:**

- both approximate 0-1 loss
- very similar asymptotic behaviour
- main difference is smoothness of LR, and non-zero outside SVM margin
- SVM gives **sparse** solution for $\alpha_i$
Next lecture

• **Beyond Binary Classification**
  • Multi-class and Multi-label
  • Using binary classifiers

• **Big Data**
  • retrieval and ranking
  • precision-recall curves

• **Nearest Neighbours (NN)**
  • Approximate NN, Locality Sensitive Hashing, Product Quantization

• **Intro to practical**