# Lecture 6: Logistic Regression

## CS 194-10, Fall 2011

Laurent El Ghaoui

EECS Department
UC Berkeley

September 13, 2011

# Outline

# Outline

# Classification task

*Data* :

- $X = [x_1, \ldots, x_m]$: a $n \times m$ matrix of data points in $\mathbf{R}^n$.
- $y \in \{-1, 1\}^m$: $m$-vector of corresponding labels.

*Classification task* : design a linear classification rule of the form

$$\hat{y} = \mathbf{sign}(w^T x + b),$$

where $w \in \mathbf{R}^n$, $b \in \mathbf{R}$ are to be found.

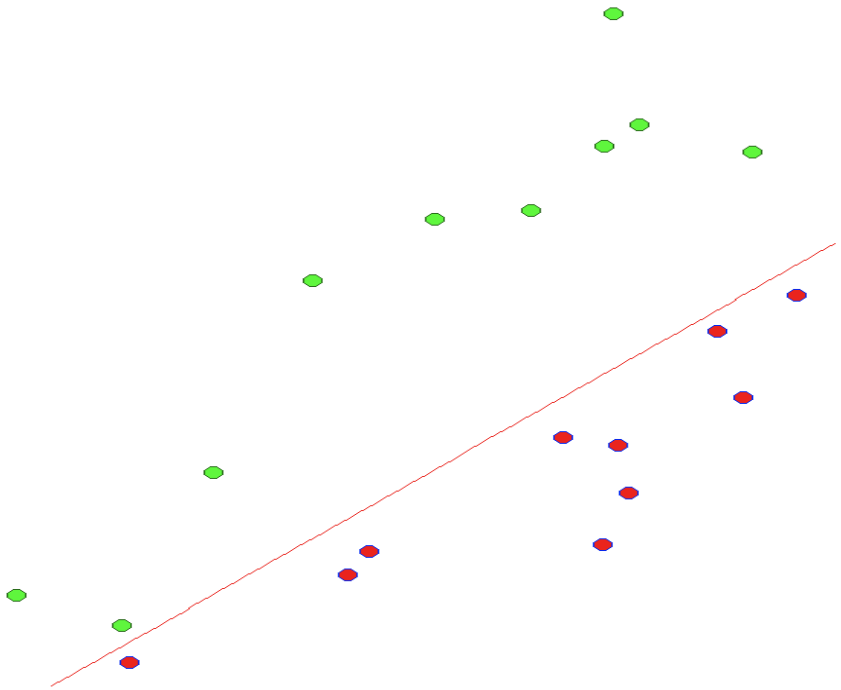
*Main solution idea* : formulate the task of finding $w, b$ as a "loss function" minimization problem.

# Separable data

Separability condition

$$y_i(w^T x_i + b) \geq 0, \quad i = 1, \ldots, m.$$

Ensures that negative (resp. positive) class is contained in half-space $w^T x + b \leq 0$ (resp. $w^T x + b \geq 0$).



# 0/1 loss function minimization

When data is not strictly separable, we seek to minimize the *number of errors* , which is the number of indices $i$ for which $y_i(w^T x_i + b) < 0$:

$$\min_{w,b} \sum_{i=1}^{m} L_{0/1}(y_i(w^T x_i + b))$$

where $L_{0/1}$ is the 0/1 loss function

$$L_{0/1}(z) = \begin{cases} 1 & \text{if } z < 0, \\ 0 & \text{otherwise.} \end{cases}$$

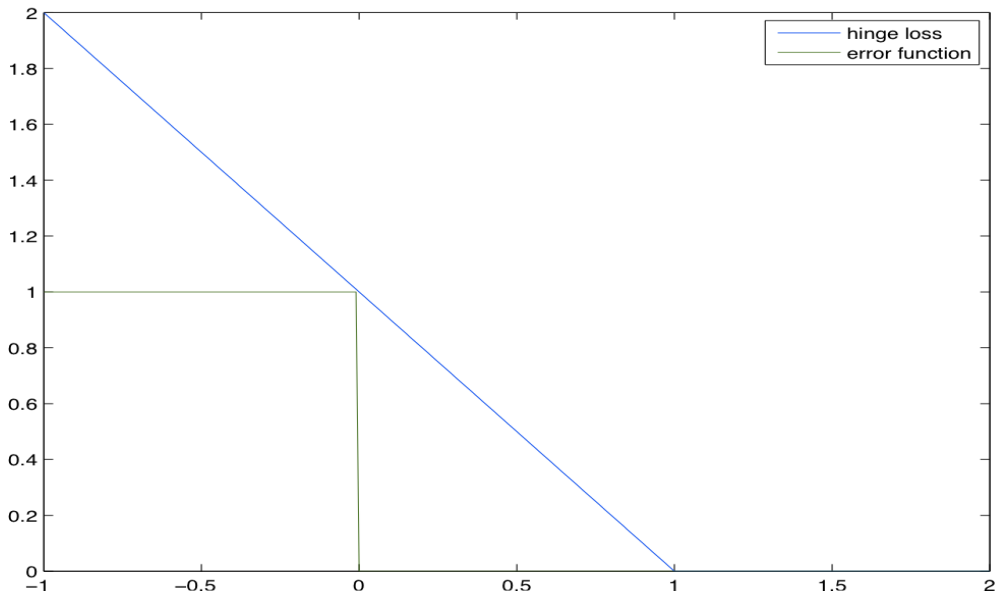
Above problem is very hard to solve exactly.

# From 0/1 loss to hinge loss

We approximate (from above) the 0/1 loss by the hinge loss:

$$H(z) = max(0, 1 - z).$$

This function is convex (its slope is always increasing).



# Support vector machine (SVM)

SVMs are based on hinge loss function minimization:

$$\min_{w,b} \sum_{i=1}^{m} \max(0, 1 - y_i(w^T x_i + b)) + \lambda \|w\|_2^2$$

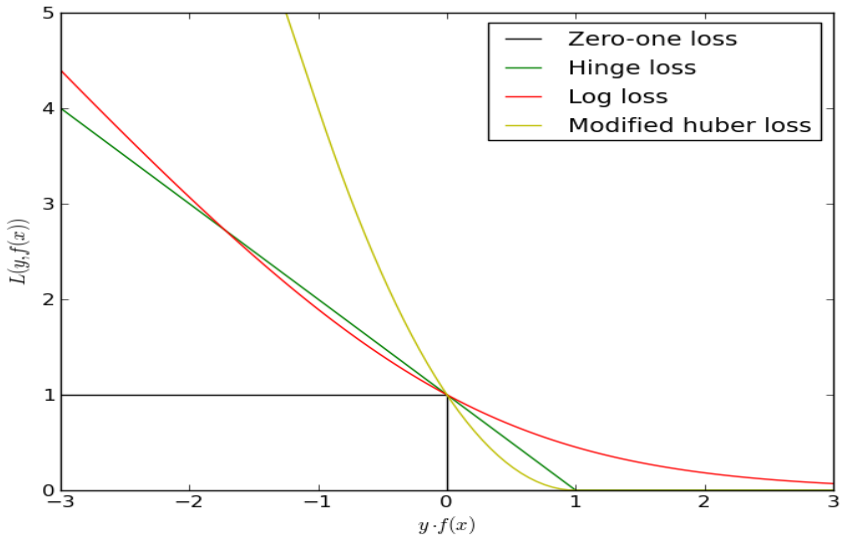
Above problem much easier to solve than with $0/1$ loss (see why later).

In lecture 5 we have seen the geometry of this approximation.

# Outline

# Basic idea

Work with a smooth (differentiable) approximation to the 0/1 loss function.




There are many possible choices, this particular one having a nice *probabilistic* interpretation.

# Logistic model

We model the probability of a label $Y$ to be equal $y \in \{-1, 1\}$, given a data point $x \in \mathbf{R}^n$, as:

$$P(Y = y \mid x) = \frac{1}{1 + \exp(-y(w^T x + b))}.$$

This amounts to modeling the *log-odds ratio* as a linear function of $X$:

$$\log \frac{P(Y = 1 \mid x)}{P(Y = -1 \mid x)} = w^T x + b.$$

- The decision boundary $P(Y = 1 \mid x) = P(Y = -1 \mid x)$ is the hyperplane with equation $w^T x + b = 0$.
- The region $P(Y = 1 \mid x) \geq P(Y = -1 \mid x)$ (*i.e.*, $w^T x + b \geq 0$) corresponds to points with predicted label $\hat{y} = +1$.

# Maximum-likelihood





The likelihood function is

$$l(w,b) = \prod_{i=1}^{m} \frac{1}{1+e^{-y_i(w^T x_i + b)}}.$$

Now maximize the log-likelihood:

$$\max_{w,b} \; L(w,b) := -\sum_{i=1}^{m} \log(1+e^{-y_i(w^T x_i + b)})$$

In practice, we may consider adding a regularization term

$$\max_{w,b} \; L(w,b) + \lambda r(w),$$

with $r(w) = \|w\|_2^2$ or $r(x) = \|w\|_1$.

# Outline

# Learning problems seen so far

Least-squares linear regression, SVMs, and logistic regression problems can all be expressed as minimization problems:

$$\min_{w} \; f(w, b)$$

with

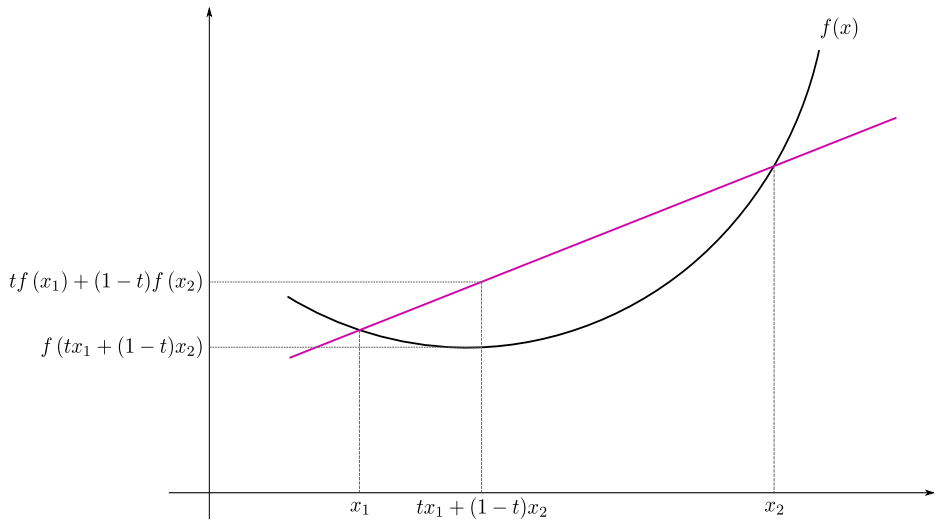
- $f(w, b) = \|X^T w - y\|_2^2$ (square loss);
- $f(w, b) = \sum_{i=1}^{m} \max(0, 1 - y_i(w^T x_i + b))$ (hinge loss);
- $f(w, b) = -\sum_{i=1}^{m} \log(1 + \exp(-y_i(w^T x_i + b)))$ (logistic loss).

The regularized version involves an additional penalty term.

How can we solve these problems?

# Convexity

The square, hinge, and logistic functions share the property of being *convex* . These functions have “bowl-shaped” graphs.




*Formal definition* : $f$ is convex if the chord joining any two points is always above the graph.

- If $f$ is differentiable, this is equivalent to the fact that the derivative function is increasing.
- If $f$ is twice differentiable, this is the same as $f''(t) \geq 0$ for every $t$.

# How to prove convexity

- A function is convex if it can be written as a maximum of linear functions. (You may need an infinite number of them.)
- If $f$ is a function of one variable, and is convex, then for every $x \in \mathbf{R}^n$, $(w, b) \to f(w^T x + b)$ also is.
- The sum of convex functions is convex.

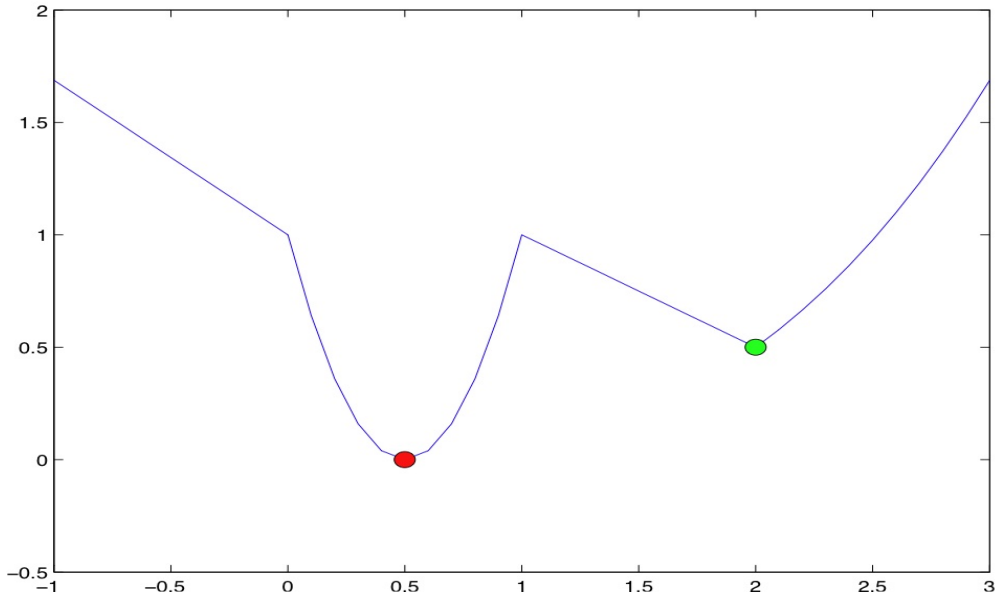
*Example* : logistic loss

$$l(z) = log(1 + e^{-z}) = \max_{0 \le v \le 1} -zv + v \log v + (1 - v) \log(1 - v).$$

# Non-convex minimization: hard



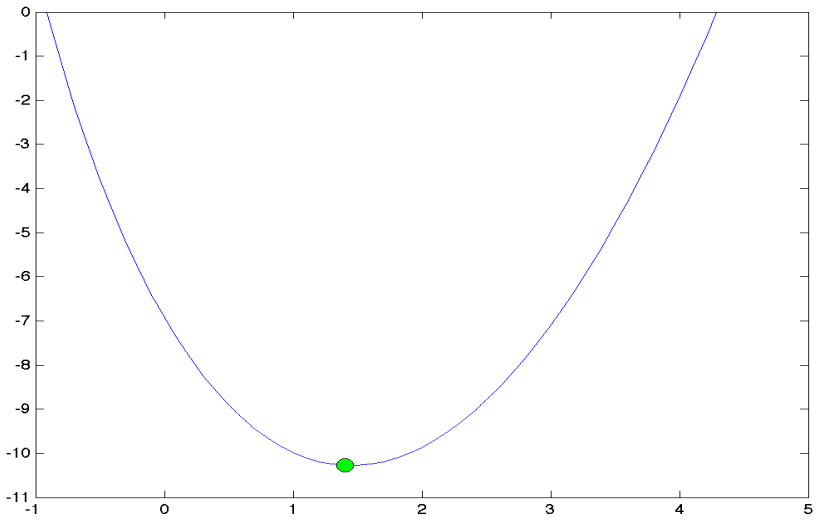





Optimizing non-convex functions (such as in $0/1$ loss minimization) is usually very hard. Algorithms may be trapped in so-called "local" minima (in green), which do not correspond to the true minimal value of the objective function (in red).

# Convex minimization: easy

Convexity ensures that there are no local minima. It is generally easy to minimize convex functions numerically via specialized algorithms. The algorithms can be adapted to cases when the function is convex but not differentiable (such as the hinge loss).




This is why the convexity properties of square, hinge and logistic loss functions are computationally attractive.

# One-dimensional case

To minimize a one-dimensional convex function, we can use bisection.

- We start with an interval that is guaranteed to contain a minimizer.
- At each step, depending on the slope of the function at the middle of the interval, we shrink the interval by choosing either the left- or right-sided interval.

Convexity ensures that this procedure converges (very fast).

# Coordinate descent

Many specialized algorithms exist for the regularized problems

$$\min_{w,b} \; L(w,b) + \lambda r(w),$$

where $r(w) = \|w\|_2^2$ or $r(x) = \|w\|_1$.

One of the most efficient involves *coordinate descent* :

- Fix all the variables except for one.
- Minimize the resulting one-dimensional convex function by bisection.
- Now proceed to minimizing w. r. t. the next variable.

For SVMs, the actual procedure involves taking two variables at a time.

Problems with millions of features and data points can be solved that way.

# In case you need to try

For moderate-size convex problems, try free matlab toolbox CVX (not Chevron!), at http://cvxr.com/cvx/.

For convex learning problems, look at libraries such as WEKA (http://www.cs.waikato.ac.nz/ml/weka/).