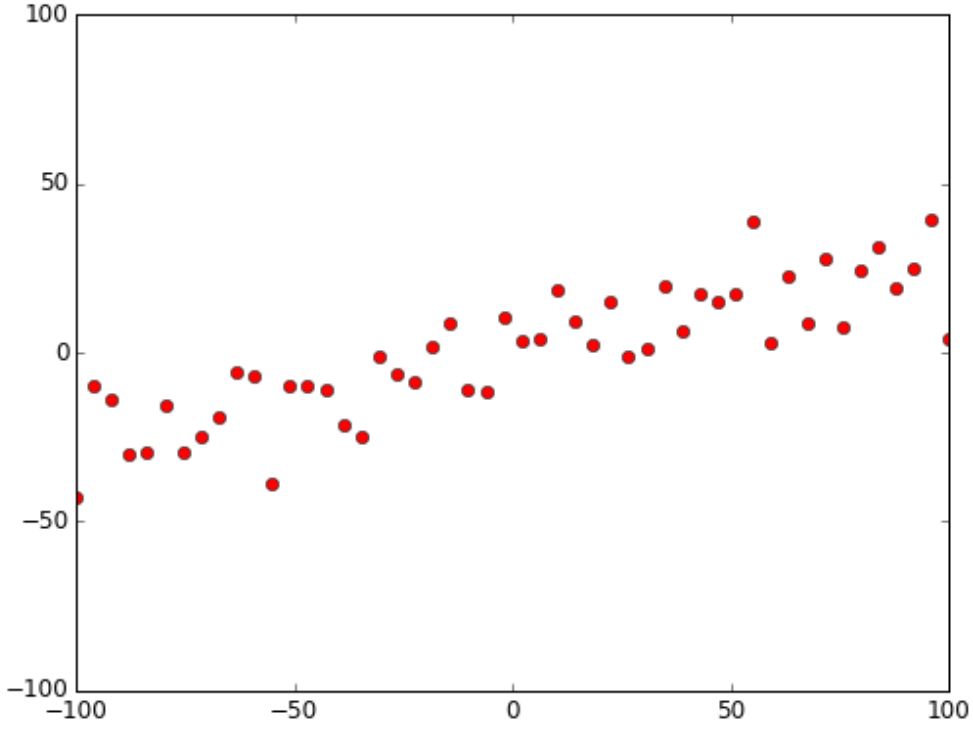
M697: Analysis and Machine Learning
Fall 2017

Nestor Guillen

Warmup

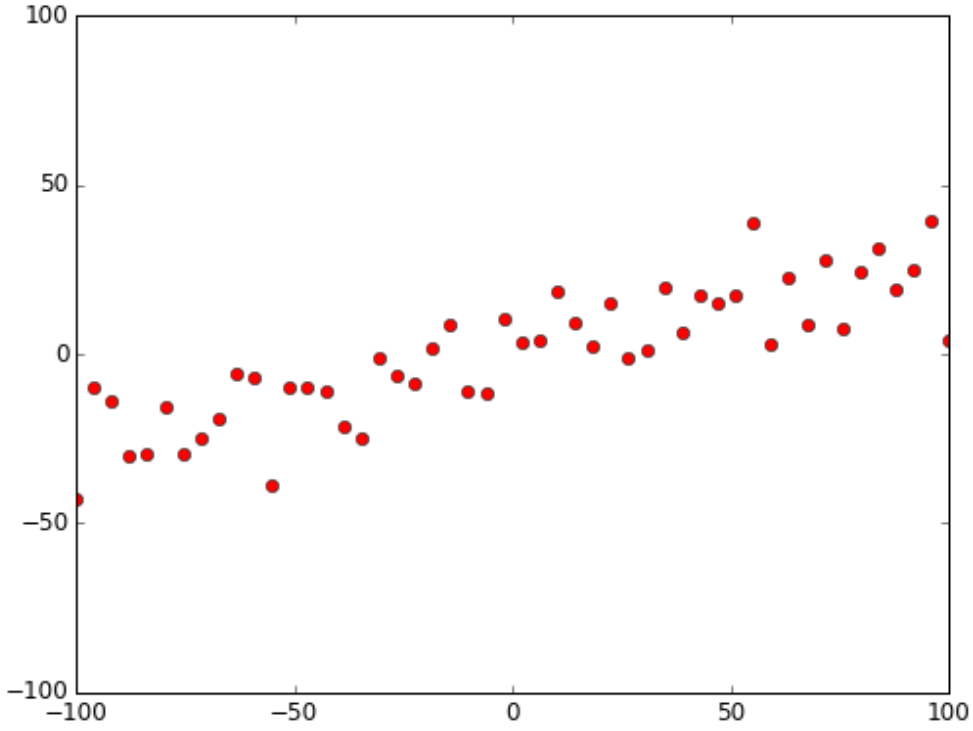


What are we looking at? This is a plot of (x_i, y_i), where

$$y_i = \beta x_i + \beta_0 + 10\varepsilon_i, \quad i = 1, \ldots, 50.$$

$$x_i = \{50 \text{ pts in } [-100, 100]\}, \; \varepsilon_i = \{50 \text{ i.i.d. Normal}(0, 1)\}$$

Warmup



Problem: You are given $\{(x_i, y_i)\}_{i=1}^{50}$, and know it has the structure above, **but** you don't know values of β_0 and β_1. What is your best guess for β and β_1?

Warmup

First Approach: Least Squared Error

Among all affine functions, find $\hat{f}$ minimizing

$$\mathcal{J}(\hat{f}) = \sum_{i=1}^{N} |\hat{f}(x_i) - y_i|^2$$

Equivalently: find $\hat{\beta}$ and $\hat{\beta}_0$ which minimize

$$\mathcal{J}(\hat{\beta}, \hat{\beta}_0) = \sum_{i=1}^{N} |\hat{\beta} x_i + \hat{\beta}_0 - y_i|^2$$

Warmup

Second Approach: Roughness Penalization

Fix $\lambda > 0$. Among all functions, find $\hat{f}$ minimizing

$$\mathcal{J}(\hat{f}) = \sum_{i=1}^{N} |\hat{f}(x_i) - y_i|^2 + \lambda \int_{-10}^{10} |\hat{f}''(x)|^2 \, dx$$

Warmup

Each approach involved searching within a class of functions $\mathcal{F}$ for an element $\hat{f}$ minimizing a functional such as

$$\sum_{i=1}^{N} |\hat{f}(x_i) - y_i|^2$$

or

$$\sum_{i=1}^{N} |\hat{f}(x_i) - y_i|^2 + \lambda \int_{-10}^{10} (\hat{f}''(x))^2 \, dx$$

Warmup

These are **variational problems**, that is, we must determine

$$\min_{f \in \mathcal{F}} \mathcal{J}(f) \quad \text{and} \quad \operatorname*{argmin}_{\mathcal{F}} \mathcal{J},$$

and study the properties of the minimizers $\hat{f}$.

Warmup

Why these functionals $\mathcal{J}$?

What if instead of the least sum of squares, we had minimized

$$\mathcal{J}(\hat{f}) = \sum_{i=1}^{N} |\hat{f}(x_i) - y_i|^p, \quad p \geq 1 \text{ ??}$$

or

$$\mathcal{J}(\hat{f}) = \sum_{i=1}^{N} |\hat{f}(x_i) - y_i|^p + \int_{-10}^{10} F(\hat{f}''(x))\, dx \text{ ??}$$

Warmup

One more thing!

The data y_i are **random variables**.

Accordingly, $\mathcal{J}$ and thus its minimizer(s) $\hat{f}$ are also random.

In the first case, this means $\hat{\beta}$ and $\hat{\beta}_0$ are random variables.

Which leads to the question: are $\hat{\beta}$ and $\hat{\beta}_0$ good estimators for β and β_0?

Demystifying ML

A lie –a reductionistic and useful one

All of the methods in machine learning boil down to solving some variational problem.

(but that's not what machine learning is about of course, that would be like saying classical physics is all about finding critical points of Lagrangians . . .but there is some value in being aware classical physics shares this unifying framework)

That being said.
Determining which classes $\mathcal{F}$ and objectives $\mathcal{J}$ are most convenient for a given situation is a challenging task, and then there is the issue of which algorithms are most effective in calculating the minimizer(s) $\hat{f}$, and then of assessing how good $\hat{f}$ is at making predictions. . .

Setup

What this course aims to cover

1. The basics of statistical inference, the fundamental issues, and basic techniques (particularly linear methods).
2. Real analysis topics of relevance to statistical inference: e.g. the manifold ways of measuring the difference between two functions or measures (norms), the geometric and topological properties of these norms.
3. The calculus of variations, a branch of mathematics that deals with the properties of minimizers of functionals arising in physics, geometry, and optimization.
4. ML methods intrinsically related to nonlinear problems arising in the calculus of variations and vice-versa (manifold learning, spectral clustering, optimal transport).

Setup

What this course is **NOT**

1. A good substitute for an introduction to machine learning or statistical inference.
2. A good substitute for a measure theory course, or a statistics course, or a probability course, or a PDE course.
3. A course where the instructor will display a great command of programming skills.
4. A course with very difficult programming assignments.
5. ~~A course with difficult math problem sets.~~ I mean define **difficult**?
6. A course with a lot of mathematical theory that turns out to be of little or no relevance to the practice of ML, predictive modeling, and statistics (well, **hopefully**).

Setup

An aspirational plan for the semester

Part I

1. Linear methods for regression.
2. Linear methods for classification.
3. Unsupervised learning (emphasis on clustering).
4. Selecting & judging learning methods ("Model assessment").

Setup

An aspirational plan for the semester

Part II

1. Some functional analysis + calculus of variations.
2. Basics of geometric measure theory and Γ-convergence.
3. Spectral theory + geometry (graphs & continuum).
4. Manifolds and manifold learning.
5. Optimal Transport and the Monge-Kantorovich distance.

Setup

Prerequisites

Minimal requirements:
Undergraduate real analysis (basics of metric spaces, integration), basic probability (distributions, random variables, law of large numbers), and a strong background in calculus and linear algebra.
Basic programming skills, ideally Python or C++, but other languages are ok (R or Matlab should be fine too). I will be using Python.

Ideal requirements:
Familiarity with one or more of the following topics: measure theory, differentiable manifolds, linear programming, spectral graph theory, calculus of variations, functional analysis, and partial differential equations.

Setup

Some notation for this semester

Euclidean Space and linear algebra
As usual, $\mathbb{R}$ = the set of real numbers

$$\mathbb{R}^p = \{(x, \ldots, x_p) \mid x_i \in \mathbb{R}\}$$

Elements of $\mathbb{R}^p$ will be denoted simply by x, **hopefully without causing too much confusion**. Matrices, that is, linear operators $\mathbb{R}^p \mapsto \mathbb{R}^q$ will be denoted by a capital letter. We will also use the following notation

$$Mx, \quad x \cdot y, \quad Mx \cdot y, \quad (Mx, y)$$

Setup

Some notation for this semester

A **Probability Space** is three things (Ω, Σ, μ):

A set Ω, a σ-algebra Σ, and a measure μ with $\mu(\Omega) = 1$.

Setup

Some notation for this semester

Conditional Probability:

$$\mathbb{P}(A \mid B) := \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

It is said A is independent from B if

$$\mathbb{P}(A \mid B) = \mathbb{P}(A)$$

That is, if

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B).$$

Setup

Some notation for this semester

Random Variables:
A random variable is a function

$$\Omega \mapsto X$$

if X is the real numbers, we say we have a real random variable. Likewise, we can have a random vector or random matrix, and so on.
Most of the time, the random variables one deals with are real-valued.

Setup

Some notation for this semester

Random variables, independence of random variables.

What we mean by "sampling from a distribution", theoretically, and practically.

Setup

Some notation for this semester

The Law of Large numbers

Let $X_1, X_2, \ldots$ be a sequence of i.i.d. real random variables. Suppose also that the second moment of the X_i is finite.

Then, if μ denotes their common mean, we have

$$\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} X_i = \mu \text{ with probability 1}$$

Setup

Python

If you decide to use Python (which I would advise for this course), you should acquaint yourself with the following libraries:

- matplotlib
- scipy
- numpy
- scikit-learn

Congratulations, you've found an easter egg. To prove you've scrolled this far down on the slides on the first two weeks of classes, send me an email with the subject line "Elementary, dear Data!"

All right, now we can finally start!

The basic problems

Problem 1: *Supervised Learning*
There is an **unknown** function

$$f : X \mapsto Y$$

between two sets X and Y, the values of which are given to us for some large but finite set $\{x_1, x_2, \ldots, x_N\} \subset X$.

Then, based on the data $f(x_i) = y_i$ for $i = 1, 2, \ldots, n$, and an **input** $x \in X$, one seeks a rule to estimate or guess the $f(x)$.

The basic problems

Problem 2: *Unsupervised learning*
You are given two set of points $\{x_1, \ldots, x_N\} \subset \mathbb{R}^p$.

You are told these points were sampled from some distribution μ, and you must determine as much about μ as possible.

e.g. is μ a normal? is the support of the distribution connected? is the distribution supported along a curve?

The basic problems

These two problems, as stated, are (technically) hopeless.

In practice, there is way more structure that often makes these problems well posed. A few examples:

- We know that $f : X \mapsto Y$ lies or is well approximated in a **special class of functions** (linear functions, quadratic functions, spectrally constrained functions, and so on).
- The observed pairs (x_i, y_i) may be random variables distributed via some **probability distribution**. In this case one may choose $f(x)$ so as to minimize the expected value of $|f(x) - y|^2$ or $|f(x) - y|$.
- The nature of the problem is such that f is reasonably represented via a **multi-layer neural net**.

The basic problems

Let us see, for instance, how searching within **linear functions** makes the first problem better posed.

Least Squares

We are given *training data*

$$(x_i, y_i) \;\; i = 1, 2, \ldots, N$$

where $X = \mathbb{R}^p$ and $Y = \mathbb{R}$.

Attempt #1 (Linear Fit): find $\beta \in \mathbb{R}^p$ and $\beta_0 \in \mathbb{R}$ such that

$$x_i \cdot \beta + \beta_0 = y_i, \;\; i = 1, \ldots, N.$$

Then, given $x \in \mathbb{R}^p$, we let

$$\hat{f}(x) = x \cdot \beta + \beta_0$$

Least Squares

We are given training data

$$(x_i, y_i) \;\; i = 1, 2, \ldots, N$$

where $X = \mathbb{R}^p$ and $Y = \mathbb{R}$.

Attempt #2 (Least Squares): find $\beta \in \mathbb{R}^p$ and $\beta_0 \in \mathbb{R}$ such that

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} |x_i \cdot \beta + \beta_0 - y_i|^2$$

is minimized. Then, given $x \in \mathbb{R}^p$, we let

$$\hat{f}(x) = x \cdot \beta + \beta_0$$

Least Squares

Let $\mathbf{X}$ the linear transformation defined by

$$\mathbf{X}\beta := (x_1 \cdot \beta, \ldots, x_N \cdot \beta)$$

and let y denote the vector $(y_1, \ldots, y_N)$.

Then, RSS(β) is simply the squared length of the vector

$$\mathbf{X}\beta - y$$

In particular, RSS(β) is a convex function of β.

Least Squares

Using the chain rule, it follows that

$$\nabla \mathrm{RSS}(\beta) = 2\mathbf{X}^t(\mathbf{X}\beta - y)$$

Therefore, the following equation yields the minimizer(s)

$$\mathbf{X}^t\mathbf{X}\beta = \mathbf{X}^t y$$

If the matrix $\mathbf{X}^t\mathbf{X}$ *happens to be invertible*, we have

$$\hat{\beta} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t y$$

In this way, the data $(\mathbf{X}, y)$ yields a best guess $\hat{\beta}$ (and thus a $\hat{f}$).

Least Squares

Now, if we are given a new input x, our best guess for $f(x)$ is

$$\hat{f}(x) = x \cdot \hat{\beta}.$$

This is still just a guess, and there are a number of ways of estimating how reliable it is.

Least Squares

Example: Classification

We are given a two finite set of points in the plane, Red and Blue, and whose union is denoted by $\{x_1, \ldots, x_N\}$.

Consider the function $f : S \mapsto \mathbb{R}$ given by

$$f(x_i) = \begin{cases} 1 & \text{if } x_i \in \text{Red} \\ -1 & \text{if } x_i \in \text{Blue} \end{cases}$$

Let us do a linear regression on $\{(x_i, y_i)\}_{i=1}^N$, where $y_i = f(x_i)$.

Least Squares

Example: Classification

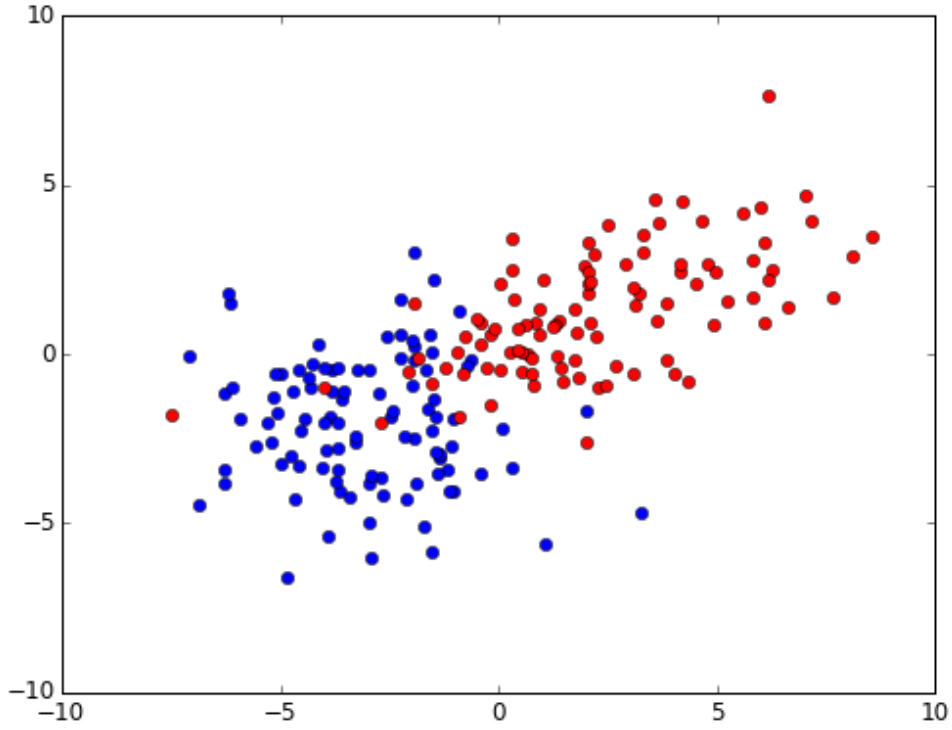
Then, given $x \in \mathbb{R}^2$, we proceed as follows

$$\hat{f}(x) \begin{cases} > 0 & \text{we classify } x \text{ as Red} \\ \leq 0 & \text{we classify } x \text{ as Blue} \end{cases}$$

This is an example of a **linear classification**.

Least Squares

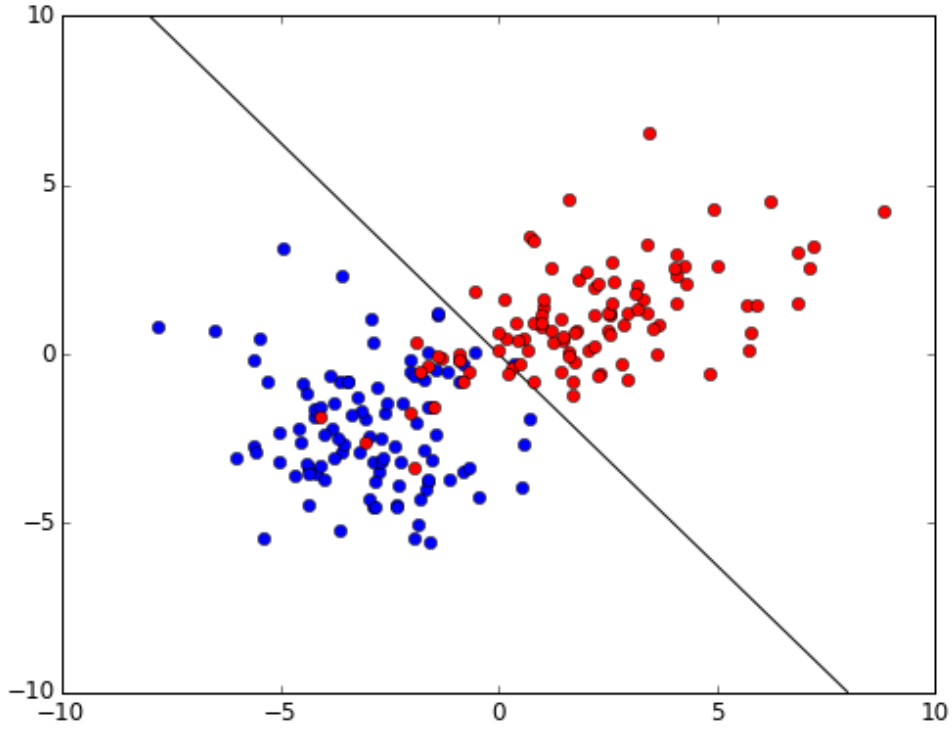
Example: Classification



I mean, what could **go wrong**?

Least Squares

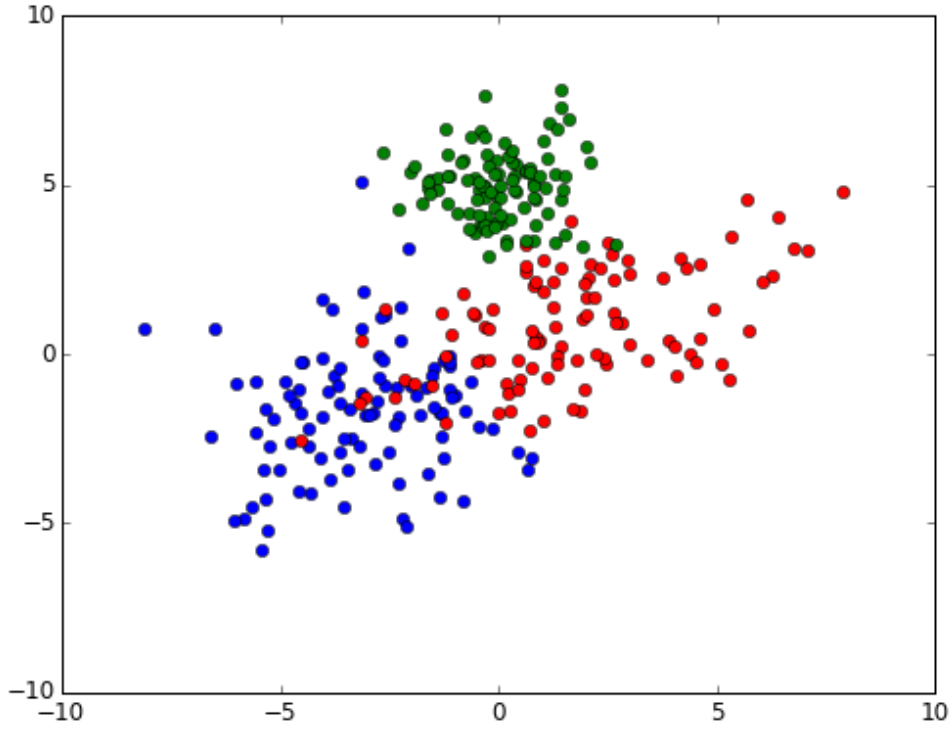
Example: Classification



I mean, what could **go wrong**?

Least Squares

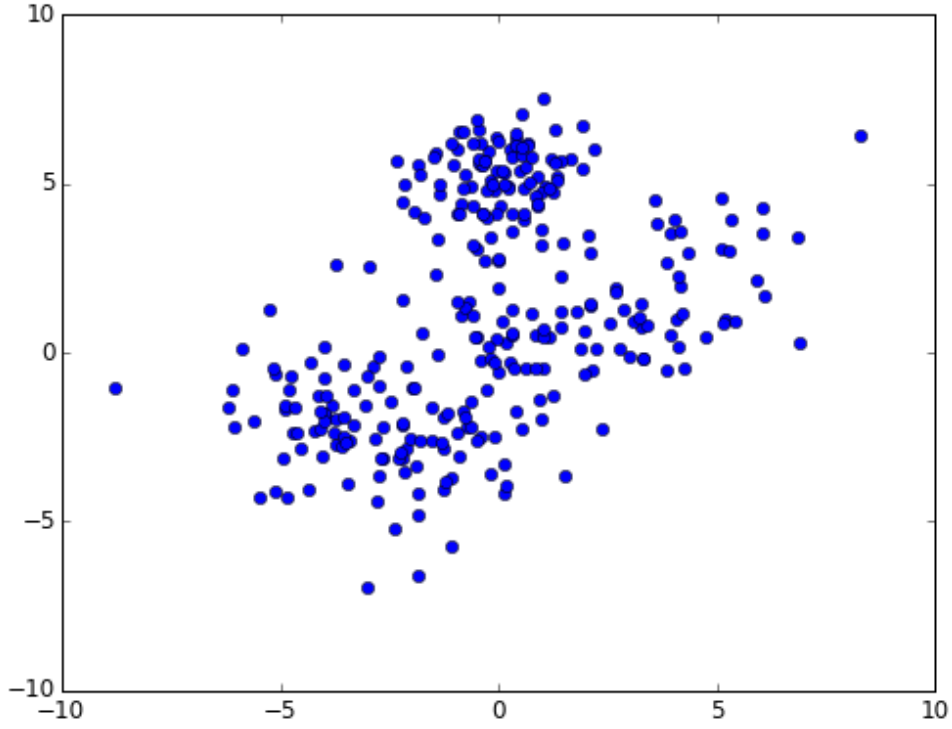
Example: Classification



I mean, what could **go wrong**?

Least Squares

Example: Classification



I mean, what could **go wrong**?

Many modes clusters may make it hard to have a single linear classifier

Last but not least, a preview

(of the second half of the semester)

First: Calculus of variations and elliptic equations

Consider the functional

$$\mathcal{J}(u) := \frac{1}{2}\int_D |\nabla u|^2 \, dx + \int_D f(x)u(x) \, dx$$

Minimizers of this problem solve Poisson's equation

$$\Delta u = f \text{ in } D$$

Last but not least, a preview

(of the second half of the semester)

First: Calculus of variations and elliptic equations

What if instead we now had

$$\mathcal{J}(u) := \lambda \int_D |\nabla u|^2 \, dx + \int_D (f(x) - u(x))^2 \, dx$$

Minimizers of this problem instead solve

$$\lambda \Delta u + u = f \text{ in } D$$

Last but not least, a preview

(of the second half of the semester)

Second: The Perimeter and Plateau's problem

The perimeter of a set $E \subset \mathbb{R}^d$ is defined as

$$\mathrm{Per}(E) = \int |\nabla \chi_E| \, dx$$

where

$$\int_D |\nabla \chi_E| \, dx = \sup \left\{ \int_E \mathrm{div}(g(x)) \, dx \;\; | \;\; g : D \mapsto \mathbb{R}^d \; \|g\|_\infty \leq 1 \right\}$$

It was first Ennio De Giorgi who figured this was an accurate, and useful way of expressing the surface area of a set.

Last but not least, a preview

(of the second half of the semester)

Second: The Perimeter and Plateau's problem

A useful functional when analyzing images is

$$\lambda \mathrm{Per}(E) + \int |f(x) - \chi_E|^2 \, dx$$

Here $f : [0,1] \times [0,1] \mapsto \mathbb{R}$ represents a gray scale image.

Problems involving the minimization of $\mathrm{Per}(E)$ lead to *minimal surfaces*, and their study have led to the development of the calculus of variations, differential geometry, and measure theory.

All right, that's all for the first class.

Don't worry, a whole semester of Data awaits us!.

M 697
Today:
Least squares, k-NN, and some probability

Nestor Guillen

Least Squares

We considered the situation where we were given data

$$\{x_1, \ldots, x_N\} \subset \mathbb{R}^p, \quad \{y_1, \ldots, y_N\} \subset \mathbb{R},$$

and wanted to minimize, over $\beta \in \mathbb{R}^p$ and $\beta_0 \in \mathbb{R}$,

$$\mathrm{RSS}(\beta, \beta_0) = \sum_{i=1}^{N} |x_i \cdot \beta + \beta_0 - y_i|^2.$$

Least Squares

We may augment every $x_i \in \mathbb{R}^p$ with a dummy 0-th coordinate

$$\tilde{x}_i = (1, x_{i1}, x_{i2}, \ldots, x_{ip})$$

so that $x_i \cdot \beta + \beta_0 = \tilde{x}_i \cdot \tilde{\beta}$, where $\tilde{\beta} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_p)$.

Thus, it suffices to think of the problem of minimizing

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} |x_i \cdot \beta - y_i|^2.$$

Least Squares

From the data, one may define a linear map / $N \times p$ matrix

$$\mathbf{X} : \mathbb{R}^p \mapsto \mathbb{R}^N$$

given by $\mathbf{X}\beta := (x_1 \cdot \beta, \ldots, x_N \cdot \beta)$, as well as a vector $y \in \mathbb{R}^N$

$$y = (y_1, \ldots, y_N).$$

$$\text{minimize } \mathrm{RSS}(\beta) = \|\mathbf{X}\beta - y\|_2^2 \text{ over } \beta \in \mathbb{R}^p.$$

Least Squares

$$\nabla \mathrm{RSS}(\beta) = \mathbf{X}^t (\mathbf{X}\beta - y)$$

A vector β achieves the minimum if and only if it solves

$$\mathbf{X}^t (\mathbf{X}\beta - y) = 0$$

equivalently, if it solves

$$(\mathbf{X}^t\mathbf{X})\beta = \mathbf{X}^t y$$

Least Squares

A quick linear algebra refresher

What does the transpose do? If M is a $n \times m$ matrix, then

$$(M^t)_{ij} = M_{ji}$$

is a $m \times n$ matrix: the rows of one are the columns of the other.

An equivalent and very useful definition is given by the identity

$$(Mx) \cdot y = x \cdot (M^t y) \quad \forall\, x \in \mathbb{R}^m,\ y \in \mathbb{R}^n.$$

Least Squares

A quick linear algebra refresher

This is why, if we think of M as

$$M : \mathbb{R}^m \mapsto \mathbb{R}^n$$

then M^T corresponds to a map in the opposite direction

$$M^t : \mathbb{R}^n \mapsto \mathbb{R}^m$$

Least Squares
$\mathbf{X}^t\mathbf{X}$

Then, back to our equation

$$(\mathbf{X}^t\mathbf{X})\beta = \mathbf{X}^t y.$$

We see that $\mathbf{X}^t\mathbf{X} : \mathbb{R}^p \mapsto \mathbb{R}^p$, is this matrix invertible?.

$$\text{Observe that } (\mathbf{X}^t X)\beta = 0 \Rightarrow (\mathbf{X}^t X)\beta \cdot \beta = 0.$$

in this case, however,

$$0 = (\mathbf{X}^t X)\beta \cdot \beta = \mathbf{X}^t(X\beta) \cdot \beta = \mathbf{X}\beta \cdot \mathbf{X}\beta = |\mathbf{X}\beta|^2$$

Least Squares
$\mathbf{X}^t\mathbf{X}$

In other words $(\mathbf{X}^t\mathbf{X})\beta = 0 \Leftrightarrow \mathbf{X}\beta = 0$.

Now, what does $\mathbf{X}\beta = 0$ mean? Well, this is the same as

$$x_1 \cdot \beta = \ x_2 \cdot \beta = \ \ldots = x_N \cdot \beta = 0$$

That is, β is orthogonal to all of the vectors $\{x_1, \ldots, x_N\}$.

Conclusion:

$\mathbf{X}^t\mathbf{X}$ is invertible $\Leftrightarrow \{x_1, \ldots, x_N\}$ spans $\mathbb{R}^p$.

Least Squares
$\mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t$

Let us then, consider only cases where $\{x_1, \ldots, x_N\}$ spans $\mathbb{R}^p$.

Then, the least squares solution $\hat{\beta}$ is unique and given by

$$\hat{\beta} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t y$$

and, the respective best linear fit for y, is the vector

$$\hat{y} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t y$$

Least Squares
$\mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t$

Note: The $N \times N$ matrix,

$$\mathbf{P} = \mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t$$

is the **orthogonal projection** onto the image of the map $\mathbf{X}$.

This means that

$$\mathbf{P}^t = \mathbf{P}, \;\; \mathbf{P}^2 = \mathbf{P}, \;\; \mathrm{Im}(\mathbf{P}) = \mathrm{Im}(\mathbf{X})$$

Least Squares

A warning about $(\mathbf{X}^t\mathbf{X})^{-1}$

An important issue in practice is when $\mathbf{X}^t\mathbf{X}$ is close to being singular, which may mean that

$$(\mathbf{X}^t\mathbf{X})^{-1}$$

is too large, creating all kind of computational issues.

Question: What needs to happen for $\mathbf{X}^t\mathbf{X}$ to be close to being non invertible? how can those situations be avoided?

Nearest Neighbor

Let us move on to a different method: k-Nearest Neighbors

k-Nearest Neighbor

As before, we start from a training data set

$$\{x_1, \ldots, x_N\} \subset \mathbb{R}^p, \quad \{y_1, \ldots, y_N\} \subset \mathbb{R}$$

and assuming this corresponds to $y = f(x)$, we estimate f.

Fix $k \in \mathbb{N}$, then the k-Nearest Neighbor algorithm returns

$$\hat{f}(x) = \sum_{x_i \in N_k(x)} y_i$$

where

$$N_k(x) = \{x_i \mid x_i \text{ is among the } k \text{ data points closest to } x\}$$

k-Nearest Neighbor

Voronoi Diagrams and 1-NN

The case $k = 1$, corresponds to simply taking

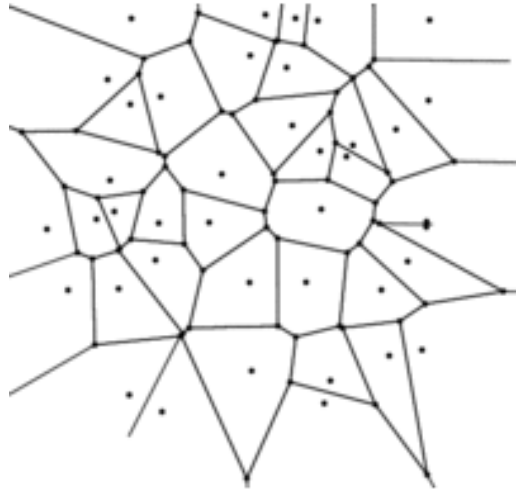
$$\hat{f}(x) = f(x_i), \quad \text{where } x_i \text{ is closest to } x.$$

Note that in any case, $\hat{f}$ is a **piecewise constant** function.

The "level sets" of $\hat{f}$ are given by the **Voronoi Diagram** of the set $\{x_1, \ldots, x_n\}$

k-Nearest Neighbor

Voronoi Diagrams and 1-NN



k-Nearest Neighbors

$$N_k(x) = \{x_i \mid x_i \text{ is among the } k \text{ data points closest to } x\}$$

Question: What if there are points x_i and x_j equidistant to x?

k-Nearest Neighbors

Back to the classification example

Assume the data came from a **classification problem** with 2 classes

$$\hat{f}(x) := \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

Since $\hat{f}(x)$ is classified in class 1 or 2 according to whether $\hat{f} > 0.5$ or $\hat{f} \leq 0.5$, the rule k-NN provides is the following

$$\text{Over a half of } N_k(x) \text{ lies in class 1} \Rightarrow x \in \text{ Class 1}$$
$$\text{At most half of } N_k(x) \text{ lies in class 2} \Rightarrow x \in \text{ Class 2}$$

k-Nearest Neighbors

Important:
Specially for high dimensions, determining $N_k(x)$ (even for $k = 1$) may entail a prohibitive computation time!

(we'll discuss this further later today and in next class)

Least Squares Versus k-Nearest Neighbor

What approach is best here?
Least Squares, or k-nearest neighbor?

The answer depends on how the data was generated.

For example:

If the data arose from **two** multinomial Gaussians, least squares.
If the data arose from **many** highly localized Gaussians, k-NN.

Expected Prediction Error

Statistical Modeling can help us navigate and judge various methods to analyze data

Expected Prediction Error

Model
We have two random variables X and Y taking values in $\mathbb{R}^d$ and $\mathbb{R}$, respectively.

We seek $f : \mathbb{R}^d \mapsto \mathbb{R}$ which intends to get to a deterministic relationship between X and Y.

Statistical Criterium:
Choose f so that it minimizes the

$$\mathbb{E}[|f(X) - Y|^2]$$

this is known as the Expected Prediction Error (EPE).

Expected Prediction Error

Recall the total expectation formula

$$\mathbb{E}[|f(X) - Y|^2] = \mathbb{E}[\mathbb{E}[|f(X) - Y|^2 \mid X]]$$

In other words,

$$\mathbb{E}[|f(X) - Y|^2] = \mathbb{E}[g(X)],$$

where

$$g(x) = \mathbb{E}[|f(X) - Y|^2 \mid X = x]$$

Expected Squared Prediction Error

This leads to the best answer, which is given by

$$\hat{f}(x) = \mathbb{E}[Y \mid X = x]$$

This is often known as the statistical **regression function**.

Expected Squared Prediction Error

What about a measure other than mean square error?

If one instead decides to minimize not the square, but the absolute value

$$\mathbb{E}[|X - Y|]$$

Then, arguing as above, one is led to

$$\hat{f}(x) = \text{median}[Y \mid X = x]$$

Expected Squared Prediction Error

What about a measure other than mean square error?

Most generally, one could consider $L : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$, a **loss function** or error function. Then, we want to minimize

$$\mathbb{E}[L(Y, f(X))]$$

and, the respective regression function would be defined as

$$\hat{f}(x) := \underset{g \in \mathbb{R}}{\operatorname{argmin}} \, \mathbb{E}[L(Y, g) \mid X = x]$$

Expected Squared Prediction Error

What about a measure other than mean square error?

In the context of classification, a popular loss function is the **zero-one** function.

The **Bayes classifier**, is the associated regression function

$$\hat{f}(x) := \underset{g}{\operatorname{argmax}}\, \mathbb{P}[Y = g \mid X = x]$$

Expected Squared Prediction Error

The return of Least Squares and k-NN

How may we estimate $\mathbb{E}[Y \mid X = x]$?

As it turns out, both Least Squares and k-NN are effectively approximating this conditional expectation –under various regimes and assumptions, that is.

Expected Squared Prediction Error

k-NN.

Fix k, and let

$$\hat{f}(x) := \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

Under rather mild assumptions*, one can show that as the sample $N \to \infty$, we have

$$\hat{f}(x) \to \mathbb{E}[Y \mid X = x]$$

(*Recall the Law of Large Numbers!)

Expected Squared Prediction Error

Least Squares.

Going back to

$$\mathrm{EPE}(f) := \mathbb{E}[|f(X) - Y|^2]$$

Assume that the minimizer f is linear, or well approximated by a linear function. Then, it makes sense of find the minimizer of $\mathrm{EPE}(f)$ within the class of linear functions.

Expected Squared Prediction Error

Least Squares.

In this case, let us write

$$f_\beta(x) = \beta \cdot x$$

and seek the minimizer of

$$\beta \mapsto \mathrm{EPE}(f_\beta), \quad \beta \in \mathbb{R}^d.$$

Expected Squared Prediction Error

Least Squares.

This is a convex functional in β, so the minimizer is given by

$$\nabla_\beta \mathrm{EPE}(f_\beta) = 0$$

That is

$$\mathbb{E}[(X \cdot \beta - Y)X] = 0$$

Which may be rewritten as

$$(\mathbb{E}[(X \otimes X])\beta = \mathbb{E}[YX]$$

Expected Squared Prediction Error

Least Squares is back!

If we average over the training data, then the solution of

$$(\mathbb{E}[(X \otimes X])\beta = \mathbb{E}[YX]$$

corresponds to least squares solution!

The moral of the story: Both k-NN and Least Squares correspond to the minimization of the Expected Squared Prediction Error, the former when we expected the regression function to be well approximated *locally* by constants, and the latter when minimizing within affine functions only.

The first week, in one slide

1. Learning problems: supervised and unsupervised.
2. Supervised learning: Regression (quantitive) and classification (categorical).
3. ML problems lead naturally to variational problems.
4. Least Squares yields a regression method is rigid.
5. k-Nearest Neighbors is a regression method which is flexibly, but less stable than least squares.
6. Learning methods/models involve parameters that quantify their complexity. One must decide what degree of variance, bias, and stability/rigidity are best for a given problem.
7. Given a statistical model for our data, minimization of EPE leads (under various regimes) to Least Squares or k-NN.