Curse of Dimensionality and Statistical Models

MATH 697 AM:ST

Tuesday, September 12th

Warmup

Parameter inference

Let $Y_1, \ldots, Y_N$ be distributed via some law f_y depending on y.

An estimator for y is **any** random variable of the form

$$\hat{Y} = g(Y_1, \ldots, Y_N)$$

The Bias and Mean Squared Error (MSE) of $\hat{Y}$ are defined as

$$\text{Bias}(\hat{Y}) := \mathbb{E}[\hat{Y}] - y, \quad \text{MSE}(\hat{Y}) := \mathbb{E}[|\hat{Y} - y|^2]$$

Warmup

Parameter inference

An estimator with $\text{Bias}(\hat{Y}) = 0$ is called unbiased.

Observe that

$$\text{MSE}(\hat{Y}) = \text{Var}(\hat{Y}) + (\text{Bias}(\hat{Y}))^2$$

When using the MSE to judge an estimator, one sees there is a balance between Bias and Variance. Being unbiased is often a desirable property, however, some estimators could be biased and have such a small Variance that their MSE is smaller in practice, and are thus preferred.

Warmup

The most basic estimators

Fix a sequence of i.i.d. variables $X_1, X_2, \ldots$

The following is the (obvious) estimator for the **mean**

$$M_N := \frac{1}{N} \sum_{i=1}^{N} X_i$$

and the two estimators below are used for the **variance**

$$\bar{S}_N^2 := \frac{1}{N} \sum_{i=1}^{N} (X_i - M_N)^2$$

$$\hat{S}_N^2 := \frac{1}{N-1} \sum_{i=1}^{N} (X_i - M_N)^2$$

Warmup

A consequence of the Strong Law of Large Numbers

Denote the common mean and variance of the X_i by μ and σ^2. Then, for all large N we have with probability ≈ 1

$$M_N \approx \mu$$
$$\bar{S}_N \approx \sigma^2$$
$$\hat{S}_N \approx \sigma^2$$

It is thus that if $x_1, x_2, \ldots$ are a particular sample of the sequence $X_1, X_2, \ldots$, then we may guess μ and σ^2 via

$$\frac{1}{N}\sum_{i=1}^{N} x_i, \quad \frac{1}{N}\sum_{i=1}^{N}(x_i - m_N)^2, \ldots$$

Warmup –from last time

Expected (Squared) Prediction Error (EPE)

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

and the conditional EPE at x_0

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

Warmup –from last time

Looking for the best linear predictor, we sought to minimize EPE within $\hat{f}$ of the form $x \cdot \beta$, leading to the equation

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

In other words, for $i = 1, \ldots, p$, we have

$$\sum_{j=1}^{p} \mathbb{E}[X_i X_j]\beta_j = \mathbb{E}[Y X_i]$$

Warmup –from last time

Let $(x_1, y_1), \ldots, (x_N, y_N)$.
Then, the Strong Law of Large Numbers says that

$$\mathbb{E}[X_i X_j] \approx \frac{1}{N} \sum_{l=1}^{N} x_{li} x_{lj}$$

$$\mathbb{E}[Y X_i] \approx \frac{1}{N} \sum_{l=1}^{N} y_l x_{li}$$

Then,

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

with the $\approx$ becoming $=$ in the limit $N \to \infty$.

Warning: A source of confusion

A few things to always keep in mind when studying the statistical properties of a predictive model:

- What is a random variable and what isn't?
- What are you sampling from?
- What are you taking the expectation with respect to?

Curse of Dimensionality

In view of all that we have said in the foregoing sections, the many obstacles we appear to have surmounted, what casts the pall over our victory celebration?

It is the curse of dimensionality, a malediction that has plagued the scientist from earliest days.

Richard Bellman in *Adaptive Control Processes*, p. 94, 1961.

Curse of Dimensionality

It is more than a matter of inconvenience. If $u = u(t)$ is specified over $0 \leq t \leq 1$ by its values at the points $k(.01)$, $k = 0, 1, 2, \ldots, 99$, we must tabulate 100 values. If $u = u(s, t)$ is specified over $0 \leq s, t \leq 1$ by a grid of similar type in s and t, we now require $100 \times 100 = 10^4$ values. Similarly, a function of three values defined over the same grid requires 10^6 values.

Since current machines have fast memories capable of storing only 3×10^4 values, and contemplated machines over the next ten years may go up only to 10^6 values, we see that multidimensional variational problems cannot be solved *routinely* because of the memory requirements.

This does not mean that we cannot attack them. It merely means that we must employ some more sophisticated techniques. We shall discuss some of

Richard Bellman in *Adaptive Control Processes*, p. 94, 1961.

Curse of Dimensionality

It is worth looking at k-NN and least squares in high dimensions.

Curse of Dimensionality

O fortuna

First, a quick observation regarding sampling.

The p-dimensional cube $Q_\ell(0) := [-\ell/2, \ell/2]^p$ has volume

$$\ell^p$$

while the cube $Q_{\alpha\ell}(0)$ for say $\alpha \in (0, 1)$ has volume

$$\alpha^p \ell^p$$

Curse of Dimensionality

Sors immanis...

The percentage of points of $Q_\ell(0)$ which lie in $Q_{\alpha\ell}(0)$ is

$$\alpha^p$$

and, for fixed α, this becomes very small when p is large.

Curse of Dimensionality

. . . et inanis

In high dimensions, the mass of a cube concentrates near its boundary

One cannot escape this fate by resorting to some of the most well known norms, it is true of the Euclidean ball too, for instance. It is clearly a consequence of the fact that the volume scales like length to the power p.

Curse of Dimensionality

k-NN

This hurts k-NN in large dimensions.

Let us apply for instance 1-NN to y and x whose functional relation is deterministic, and known, say

$$y = f(x) = e^{-5|x|^2} \quad x \in \mathbb{R}^p,\ p \geq 10$$

A computer computation tells you that

$$f(x) \leq 0.3 \text{ if } |x| \geq 0.5$$

Curse of Dimensionality

k-NN

Let us draw, at random, uniformly in $Q_1(0)$, points $X_1, \ldots, X_N$. Then, 1-NN estimate for y for $x = 0$ is

$$\hat{y} = e^{-5|X_{k^*}|^2}$$

k^* being so X_{k^*} has smallest norm among $X_1, \ldots, X_N$.

Question: How large N needs to be for $\hat{y}$ to have a reasonable chance (say ≥ 0.5) of being closer to 1 than to 0?

Answer: One needs to choose $N \geq C^p$ for a certain $C > 1$.

Curse of Dimensionality

k-NN

The moral of the story is that the training set we need for 1-NN in some instances must have a number of elements which grows exponentially with the dimension of the problem.

Curse of Dimensionality

Linear model with noise

Not all is lost, however, certain high dimensional problems have been tackled in a computationally practical manner under further assumptions, i.e. linearity.

Let us take the method of least squares as an illustration of this.

Curse of Dimensionality

Linear model with noise

Consider three random variables $X \in \mathbb{R}^p$, $Y, \varepsilon \in \mathbb{R}$, with

$$Y = X \cdot \beta + \varepsilon$$

Assume X and ε are independent.

Curse of Dimensionality

Linear model with noise

Important: In what follows, we are going to study the statistics of **all** i.i.d. samples of size N for this model.

So, fix N, and take $(X_1, Y_1, \varepsilon_1), \ldots, (X_N, Y_N, \varepsilon)$ i.i.d. distributed as X, Y, and ε, and independent from the original variables X, Y, and ε, which will be used as our test variables.

In particular, for each i we have $Y_i = X_i \cdot \beta + \varepsilon_i$, these are not merely real numbers and vectors, but are random variables!

Curse of Dimensionality

Linear model with noise

Now, from the $X_1, \ldots, X_N$ we construct a $N \times p$ **random matrix**, defined as follows: for $v \in \mathbb{R}^p$, let

$$\mathbf{X}v := (X_1 \cdot v, \ldots, X_N \cdot v) \in \mathbb{R}^N$$

Moreover, we construct two random vectors in $\mathbb{R}^N$,

$$\mathbf{y} = (Y_1, \ldots, Y_N)$$
$$\bar{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_N)$$

Observe that the relationship between X_i, Y_i, and ε_i becomes

$$\mathbf{y} = \mathbf{X}v + \bar{\varepsilon}$$

Curse of Dimensionality

Linear model with noise

In summary, we have:
A **random** $N \times p$ matrix $\mathbf{X}$ and vectors $\mathbf{y}, \bar{\varepsilon} \in \mathbb{R}^N$ with

$$\mathbf{y} = \mathbf{X}v + \bar{\varepsilon}$$

where the rows of $\mathbf{X}$ are made out of $X_1, X_2, \ldots$

We now bring in the the least squares solution, which is given by

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

Observe: this $\hat{\beta}$ is a random vector, and a function of the X_i, Y_i, and ε_i. We intend to use it as an **estimator** for β.

Curse of Dimensionality

Linear model with noise

Let's compute the Mean Squared Error associated to $\hat{\beta}$

$$\mathrm{MSE}(f_{\hat{\beta}}, x_0) = \mathbb{E}[|Y - \hat{\beta} \cdot x_0|^2 \mid X = x_0]$$

By independence of X and ε, we have

$$\mathrm{MSE}(f_{\hat{\beta}}, x_0) = \mathbb{E}[|(\beta - \hat{\beta}) \cdot x_0|^2 \mid X = x_0] + \mathbb{E}[\varepsilon^2]$$

Curse of Dimensionality

Linear model with noise

Here is where we put to use our random variable set up! Recalling that,

$$\mathbf{y} = \mathbf{X}\beta + \bar{\varepsilon} \in \mathbb{R}^N$$

where $\bar{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_N) \in \mathbb{R}^N$, we have that

$$\begin{aligned}
\hat{\beta} &= (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{y} \\
&= (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{X}\beta + (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\bar{\varepsilon} \\
&= \beta + (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\bar{\varepsilon}
\end{aligned}$$

Curse of Dimensionality

Linear model with noise

So, we have

$$\hat{\beta} = \beta + (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\bar{\varepsilon}$$

From the independence of X_1, X_2 from $\varepsilon_1, \varepsilon_2$, it follows that

$$\mathbb{E}[(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\bar{\varepsilon}] = \mathbb{E}[(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t]\mathbb{E}[\bar{\varepsilon}] = 0$$

Therefore $\mathbb{E}[\hat{\beta}] = \beta$, i.e. $\hat{\beta}$ is an unbiased estimator for β.

Curse of Dimensionality

Linear model with noise

Back to the MSE, using our new found formula we have

$$\begin{aligned}&\mathbb{E}[|(\beta - \hat{\beta}) \cdot x_0|^2 \mid X = x_0] \\ &= \mathbb{E}[|((\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\bar{\varepsilon}, x_0)|^2 \mid X = x_0]\end{aligned}$$

and therefore,

$$\mathrm{MSE}(\hat{\beta}, x_0) = \mathbb{E}[|((\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\bar{\varepsilon}, x_0)|^2 \mid X = x_0] + \sigma^2$$

Curse of Dimensionality

Linear model with noise

Following the last identity with some meticulous calculations, one can see that

$$\mathrm{MSE}(\hat{\beta}, x_0) = \mathbb{E}[(\mathbf{X}^t\mathbf{X})^{-1}x_0, x_0) \mid X = x_0] + \sigma^2$$

Further, with a bit more work, it can be shown that in fact

$$\mathrm{MSE}(\hat{\beta}) = \frac{p}{N}\sigma^2 + \sigma^2$$

Voilá! The growth in p it's **only linear in the dimension**, and the number of samples N only needs to be linear in p to make MSE reasonably small.

Curse of Dimensionality

Linear model with noise

Voilá! The growth in p it's **only linear in the dimension**, and the number of samples N only needs to be linear in p to control the MSE.

Thus, with extra structure, the curse of dimensionality was circumvented.

A lot of ongoing research in ML entails developing ways of getting around the curse of dimensionality in situations where linearity is not a good approximation.

Statistical models

Having met with the curse of dimensionality and seen the type of problems lying ahead for our learning methods, let's discuss a few structures which, together with the formalism of the EPE, lead to many important algorithms.

We continue to think in terms of two random variables X and Y, for which we seek to understand any deterministic relation of the form $Y = f(X)$. We already saw the EPE leads to

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

Statistical models

Additive Noise Model

Let X and ε be independent random variables, with $X \in \mathbb{R}^p$, and $\varepsilon \in \mathbb{R}$ being such that $\mathbb{E}[\varepsilon] = 0$, $\mathrm{Var}(\varepsilon) = \sigma_\varepsilon^2$.

We assume there is some f (the exact form of which is unknown), and consider the random variable Y given by

$$Y = f(X) + \varepsilon$$

Problem: Knowing this structure, and knowing as little as possible about f, can we estimate f by sampling X and Y?

Statistical models

Additive Noise Model

Note that for the additive noise model we have

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

If f does not oscillate nor change too rapidly, k-NN could be quite adept at recovering $f(x)$ from training data $\mathcal{T} = \{(x_i, y_i)\}$.

(From the curse of dimensionality, the larger the dimension, the more rapid changes in f affect the computations)

Statistical models

Restricted function classes

One way to impose further structure on our problem is by restricting function classes where one minimizes the EPE.

We have already done this in two cases: linear regression and k-NN. The former was too restrictive, the latter too unstable.

Statistical models

Restricted function classes

In between these extremes we have classes of functions which, while still restricted, go far beyond complexity when compared to linear functions.

Often, a vector θ is used to parametrize the space of functions under consideration. In such a case, we shall write

$$f(x, \theta) = f_\theta(x).$$

Then minimizing $\mathrm{EPE}(f_\theta)$ becomes a (possibly nonlinear, or even non-convex) minimization problem.

Statistical models

Restricted function classes

Example: The sigmoid function with slope β,

$$h_\beta(x) = \sigma(x \cdot \beta) = \frac{1}{1 + e^{-x \cdot \beta}}, \quad \text{for } \beta \in \mathbb{R}^p$$

here, σ denotes the well known sigmoid function

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

Structured regression

Restricted function classes

Example: One may use linear basis expansions, which means that one first preselects and fixes a family of functions

$$h_1(x), \ldots, h_M(x) : \mathbb{R}^p \mapsto \mathbb{R}$$

Then, for $\theta \in \mathbb{R}^M$, we set

$$f_\theta(x) := \sum_{m=1}^{M} \theta_m h_m(x)$$

If the $h_m(x) = x_m$ are the components of the vector, from minimizing $\mathrm{EPE}(f_\theta)$ we get back the least squares method.

Structured regression

Basis functions

Example: A single layer, feed forward, neural net*

$$f_\theta(x) := \sum_{m=1}^{M} \beta_m \sigma(\alpha_m \cdot x + b_m)$$

here the parameters allows both to change the slope and location of the sigmoids by choosing the α_m's and b_m's, and their amplitude, by changing the β_m's.

Accordingly, we have $\theta = (\beta_1, \alpha_1, b_1, \ldots, \beta_M, \alpha_M, b_M)$.

*also known as *basis pursuit*.

Next class

1. More on EPE(f_θ)
2. Other functionals besides squared error.
3. Further restricted function classes (often equivalents to the ones seen today): kernel methods and roughness penalization.
4. Model selection and more on bias versus variance.

Reminders: 1) Don't forget the first problem set is **due** Thursday of the coming week, and the second problem set is due the Thursday after that.
2) Check your email today for a questionnaire –you must also use it to tell me who will be your project partner.

Statistical Models and Model Selection

MATH 697 AM:ST

Thursday, September 14th

Warm up

Today, let us start by discussing, briefly, two simple ideas

- Maximum likelihood as an statistical estimation method.
- How least squares changes when one has a **weighted sum**.

Warm up I

Maximum Likelihood

Let $Y_1, \ldots, Y_N$ be random variables with joint distribution

$$\mathrm{p}(y_1, \ldots, y_N; \theta)$$

with a parameter vector $\theta = (\theta_1, \ldots, \theta_M)$, for some M.

Example:
The Y_i are i.i.d. Normals in $\mathbb{R}^p$ with $\mathrm{Cov}(Y_i) = \sigma^2 \mathrm{I}$ and mean θ

$$\mathrm{p}(y_1, \ldots, y_N; \theta) = \prod_{i=1}^{N} \frac{1}{(2\pi\sigma^2)^{\frac{p}{2}}} e^{-\frac{|y_i - \theta|^2}{2\sigma^2}}$$

Warm up I

Maximum Likelihood

Given a sample $y_1, y_2, \ldots, y_N$, choose $\hat{\theta}$ so as to maximize their probability, that is, take

$$\hat{\theta} := \operatorname*{argmax}_{\theta} \mathrm{p}(y_1, \ldots, y_N; \theta)$$

This $\hat{\theta}$ is somestimes called the **maximum likelihood** estimator. It corresponds to the idea that if $y_1, \ldots, y_N$ was observed, then it is a good guess to choose the θ that made this observation most likely.

This intuition is further justified by the same idea behind the Strong Law of Large Numbers: if N is large, then the above estimation should be more accurate.

Warm up I

Maximum Likelihood

If the Y_i are independent, what we are maximizing is a product

$$\mathrm{p}(y_1, \ldots, y_N; \theta) = \prod_{i=1}^{N} \mathrm{p}(y_i; \theta)$$

This leads naturally to the idea of **log-probability**, since

$$\begin{aligned} &\log(\mathrm{p}(y_1, \ldots, y_N; \theta)) \\ &= \sum_{i=1}^{N} \log(\mathrm{p}(y_i; \theta)) \end{aligned}$$

Warm up I

Maximum Likelihood

Therefore, it is clear in this case that maximizing

$$\prod_{i=1}^{N} \mathrm{p}(y_i; \theta)$$

is exactly the same as maximizing the sum

$$\sum_{i=1}^{N} \log(\mathrm{p}(y_i; \theta))$$

Warm Up I

Maximum Likelihood

Example: Following our previous example, for given a sample $y_1, \ldots, y_N$ we seek $\hat{\theta}$ which maximizes

$$\sum_{i=1}^{N} \log(\mathrm{p}(y_i; \theta))$$

$$= -\frac{Np}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{N} |y_i - \theta|^2$$

Warm Up I

Maximum Likelihood

Example: (continued)
But, this is the same as minimizing

$$\frac{1}{2\sigma^2}\sum_{i=1}^{N}|y_i-\theta|^2$$

It follows that $\hat{\theta}$ is given by averaging the y_i

$$\hat{\theta}=\frac{1}{N}\sum_{i=1}^{N}y_i$$

Warm Up II

Weighted Least Squares

We are given the following:

$$\text{Data } (x_1, y_1), \ldots, (x_N, y_N)$$
$$\text{“weights” } K_1, \ldots, K_N \text{ all positive numbers.}$$

Then, let us see how different is it to minimize the functional

$$J(\beta) = \sum_{i=1}^{N} K_i |y_i - x_i \cdot \beta|^2$$

which yields back the usual least squares if $K_1 = \ldots = K_N = 1$.

Warm Up II

Weighted Least Squares

What is different?

Warm Up II

Weighted Least Squares

What is different?

As usual for least squares, let us define $\mathbf{y}$ and $\mathbf{X}$ by

$$\mathbf{y} = (y_1, \ldots, y_N), \quad \mathbf{X}\beta = (x_1 \cdot \beta, \ldots, x_N \cdot \beta),$$

and let $\mathbf{K}$ be the $N \times N$ diagonal matrix whose ii-th entry is K_i.

With this notation, it turns out that

$$J(\beta) = (\mathbf{K}(\mathbf{X}\beta - \mathbf{y}), (\mathbf{X}\beta - \mathbf{y}))$$

Warm Up II

Weighted Least Squares

A **responsible** use of both linear algebra & the chain rule yields

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

Thus, the minimizer(s) β solve

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

and, provided we can invert $\mathbf{X}^t\mathbf{K}\mathbf{X}$, the minimizer is unique, and given by

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

Warm Up II

Weighted Least Squares

It is rather satisfactory to see that the formula for $\hat{\beta}$

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

Arises from a minor modification of the standard formula

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

Exercise: Can one easily check the invertibility of $(\mathbf{X}^t\mathbf{K}\mathbf{X})^{-1}$?

Statistical models

Additive Models and the Likelihood Functional

We continue to study the additive model: three random variables X, Y and ε satisfy the relation

$$Y = f(X) + \varepsilon$$

for some unknown f. X and ε are assumed independent, and

$$\mathbb{E}[\varepsilon] = 0.$$

Our purpose is, as always, is to estimate f at some x from a given sample of N observations (x_i, y_i), where $y_i = f(x_i) + \varepsilon_i$.

Statistical models

Additive Models and the Likelihood Functional

As we have seen, it follows from our assumptions that

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

$$(\text{and } \mathrm{p}_{Y|X}(y) \text{ is } \mathcal{N}(f(X), \sigma^2) \ )$$

This is more or less where we were last time, before we started our discussion of some of families of restricted functions.

Statistical models

Additive Models and the Likelihood Functional

Thus, picking up from last time, we restrict ourselves to a (paremetrized) family of functions, labeled by parameter θ.

This does not mean the unknown f belongs to this family, nor that we expect it to be. We make no assumption on whether the function f, which we seek to learn, is among the $f_\theta \ldots$

$\ldots$we merely contend ourselves with finding the "best" approximation to f among the f_θ.

Of course,"best approximation" is up to us to define!

Statistical models

Additive Models and the Likelihood Functional

Remember the Bayesian classifier: If Y is a categorical variable,

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

A similar idea yields a criterium to choose a best fit $\hat{f}$ for given training data, where the best fit $\hat{f}$ is sought within a predetermined class of paremetrized functions f_θ.

Statistical models

Additive Models and the Likelihood Functional

Motivated by this, we consider a predictor for the additive model $Y_\theta = f_\theta(X) + \varepsilon$ where we also have

f_θ a family of fuctions parametrized by θ

$x_1, \ldots, x_N$ a fixed training set $\subset \mathbb{R}^p$

One possible choice for "best "f_θ" is to choose θ which maximizes the Likelihood of the observed data...

Statistical models

Additive Models and the Likelihood Functional

. . .which is done as follows: given a sample $y_1, \ldots, y_N$, we set

$$\hat{\theta} := \operatorname*{argmax}_{\theta} \sum_{i=1}^{N} \log(\mathrm{p}_{Y_\theta | X = x_i}(y_i))$$

and then let, accordingly, $\hat{f}(x) = f_{\hat{\theta}}(x)$.

Statistical models

Additive Models and the Likelihood Functional

Example:
Assume that $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. Then, for each y_i, we have

$$\log(\mathrm{p}_{Y_\theta|X=x}(y_i)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{|y-f_\theta(x)|^2}{2\sigma^2}}$$

It follows that the log-probability is

$$-\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}|y_i - f_\theta(x_i)|^2$$

Statistical models

Additive Models and the Likelihood Functional

We conclude that for the additive model with Gaussian noise:

$$\text{Maximum likelihood} \equiv \text{Least Squares}$$

Statistical models

Additive Models and the Likelihood Functional

We conclude that for the additive model with Gaussian noise:

$$\text{Maximum likelihood} \equiv \text{Least Squares}$$

What other objective functionals may arise this way?

Statistical models

Additive Models and the Likelihood Functional

Let $H : \mathbb{R} \mapsto \mathbb{R}$ be such that

$$\int_{\mathbb{R}} e^{-H(x)}\, dx < \infty, \quad \int_{\mathbb{R}} e^{-H(x)} x\, dx = 0.$$

e.g. if $H(x) = H(-x)$ and $H(x) \geq \alpha_0 |x| - \beta_0$ with $\alpha_0 > 0$.

Then, maximum likelihood leads amounts to minimzing

$$\hat{\theta} = \operatorname*{argmin}_{\theta} \sum_{i=1}^{N} H(y_i - f_\theta(x_i))$$

that is, provided ε has a Gibbs distribution $Z_H^{-1} e^{-H(x)}$.

Statistical models

Other Objective functionals?

Of course, it is less obvious if other objective functionals of interest arise in the same way

$$\frac{1}{N}\sum_{i=1}^{N}|x_i\cdot\beta-y_i|^2+\lambda\sum_{i=1}^{N}\beta_i^2$$

$$\frac{1}{N}\sum_{i=1}^{N}|x_i\cdot\beta-y_i|^2+\lambda\sum_{i=1}^{N}|\beta_i|$$

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

This question we will not explore, but we will be studying these functionals, and will be returning to maximum likelihood.

Structured regression

A way to get around the curse of dimensionality

1. Roughness penalty
2. Kernel methods
3. Basis functions

The third item above we discussed last class, where we looked at restricted families of the form

$$f_\theta(x) = \sum_{i=1}^{M} \theta_i h_i(x)$$

As we are about to see, these three methods are closely connected

Structured regression

Penalizing lack of regularity

Let us revisit roughness penalization, where to N data points (x_i, y_i) we associate the functional

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

Structured regression

Penalizing lack of regularity

We let $\hat{f}$ be the minimizer of $\mathcal{J}(f)$, where

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

Theorem: There exists a function $K(x, y)$ such that

$$\hat{f}(x) = \frac{1}{N}\sum_{i=1}^{N} K(x, x_i) y_i \;\; \forall \, x.$$

See B. Silverman, Annals of Statistics, 1984.

Structured regression

Penalizing lack of regularity

This theorem, follows in turn, from another basic fact about the minimizer, $\hat{f}$: it may be written as

$$\hat{f}(x) = \sum_{i=1}^{M} \theta_i h_i(x)$$

where the θ_i can be computed explicitly from the data (x_i, y_i), and the h_i is a certainly family of piece wise polynomial functions known as **cubic splines** and which depend only on the x_i and not the y_i.

Structured regression

Penalizing lack of regularity

That this is so follows easily from the exercises in the first problem set. One of them corresponds to the following observation: if

$$g : [a, b] \mapsto \mathbb{R}$$

minimizes, among all g's with fixed values at a and b,

$$\int_a^b |g''(t)|^2 \, dt$$

Then, g must be a third order polynomial in $[a, b]$.

Structured regression

Penalizing lack of regularity

For the functional $\mathcal{J}$, we have that minimizers need to be piecewise second order polynomials, with the polynomial possibly changing from the interval (x_i, x_{i+1}) to (x_{i+1}, x_{i+2}) (assuming that we have labeled the x_i in increasing order).

A bit more work will show that this solution is also such that it's second order derivative is continouous across each x_i.

Structured regression

Kernels & Local Regression

Now, to motivate the kernel expression, let us introduce the ideal of a local average

$$\mathrm{RSS}(f_\theta, x_0) = \sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - f_\theta(x_i))^2$$

The idea is that $K_\lambda(x_0, x_i)$ is very small when $|x_0 - x_i|$ is sufficiently large, this being controlled by the parameter λ.

Thus, one does at **each** point x_0, a least squares fit adapted to the points close to x_0.

Structured regression

Kernels & Local Regression

Gaussian weights with variance λ

$$K_\lambda(x,y) = \frac{1}{(2\pi\lambda)^{\frac{p}{2}}} e^{-\frac{|x-y|^2}{\lambda}}$$

The Epanechnikov Quadratic kernel

$$K_\lambda(x,y) = D(\lambda^{-1}|x-y|),$$
$$\text{where } D(t) = \tfrac{3}{4}(1-t^2)\chi_{[-1,1]}(t)$$

Structured regression

Kernels & Local Regression

$$\mathrm{RSS}(f_\theta, x_0) = \sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - f_\theta(x_i))^2$$

Two classes of functions

$$\begin{aligned} f_\theta(x) &:= \theta && \text{(locally constant)}, \\ f_\theta(x) &:= \theta_0 + \theta_1 \cdot x && \text{(locally linear)} \end{aligned}$$

Structured regression

Kernels & Local Regression

In the first case, f_θ is constant, and

$$\mathrm{RSS}(f_\theta, x_0) = \sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - \theta)^2$$

This is easily seen to be a quadratic polynomial in θ, differentiating to obtain the minimum $\hat{\theta}$, one arrives at

$$\sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - \hat{\theta}) = 0$$

$$\Rightarrow \hat{\theta} = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$

This is known as the *Nadaraya-Watson* weighted average

$$\hat{\theta} = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$

Structured regression

Kernels & Local Regression

In the second case, we have, for $\theta = (\theta_0, \theta_1) \in \mathbb{R}^{p+1}$

$$\mathrm{RSS}(f_\theta, x_0) = \sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - \theta_0 - \theta_1 \cdot x_i)^2$$

This turns out to be a quadratic polynomial in θ_0 and the components of θ_1, computing the gradient as we did for least squares, we obtain this time the formula

$$(\hat{\theta}_0(x_0), \hat{\theta}_1(x_0)) = (\mathbf{X}^t\mathbf{K}(x_0)\mathbf{X})^{-1}\mathbf{X}^t\mathbf{K}(x_0)\mathbf{y}$$

The corresponding prediction $\hat{f}(x_0)$ is then given by

$$\hat{f}(x_0) = \sum_{i=1}^{N} \ell_i(x_0) y_i$$

where the coefficients $\ell_i(x_0)$ can be computed from x_0 and the components of the matrix $(\mathbf{X}^t \mathbf{K}(x_0) \mathbf{X})^{-1} \mathbf{X}^t \mathbf{K}(x_0)$.

The second week, in one slide

1. The Curse of Dimensionality means high dimensional problems are in general computationally intractable, and in particular hampers the usability of k-NN.
2. If further structure is available (e.g. linearity) the curse of dimensionality may be avoided.
3. When selecting a predictive model, one may judge it by looking at the *training error* and the *test error*. Of these two, the *training error* is by the better evaluation metric.
4. A biased algorithm is often preferable to an unbiased one: the tradeoff may lead to a reduced mean squared error.
5. There are many choices to restrict the function classes for regression: roughness penalization, basis functions, kernel methods, local regression...