Working with Data (6.419x)

What tools can we use to extract, extrapolate, and explain patterns in data? In this class we'll introduce modern answers to this question. We assume experience with probability and programming so that we may leave implicit both the articulation of measure-theory technicalities and the implementation of pseudocode.

These notes divide into six parts: a head-first tour of the kinds of analysis we'll do — skim this part without trying to understand each step; then two foundational sections on the statistics of high-dimensional data; then three sections adapting this theory to common kinds of data.

Table of Contents

a prologue in examples

contaminated faucets

VISUALIZATION — The mineral kryptonite contaminates water faucets in a community. We know the latitude-longitude coordinates of 10^6 faucets (drawn from city plumbing records). We also know which of 10^4 randomly selected faucets are contaminated. We want to predict which unlabeled faucets are contaminated. To start, let's randomly partition our overall dataset into portions of relative sizes $(0.8, 0.1, 0.1)$; we'll call these our *training*, *validation*, and *testing* sets. As we develop, we swear never to look at the testing set, so that we don't fool ourselves into thinking we found a pattern; we'll later talk much more about this danger of *overfitting*.

Let's take a look at the training set to see what patterns we might exploit. We already have a mixture model of strong priors (implicitly) in our heads, and looking at the training set helps us focus on one of the mixture components:

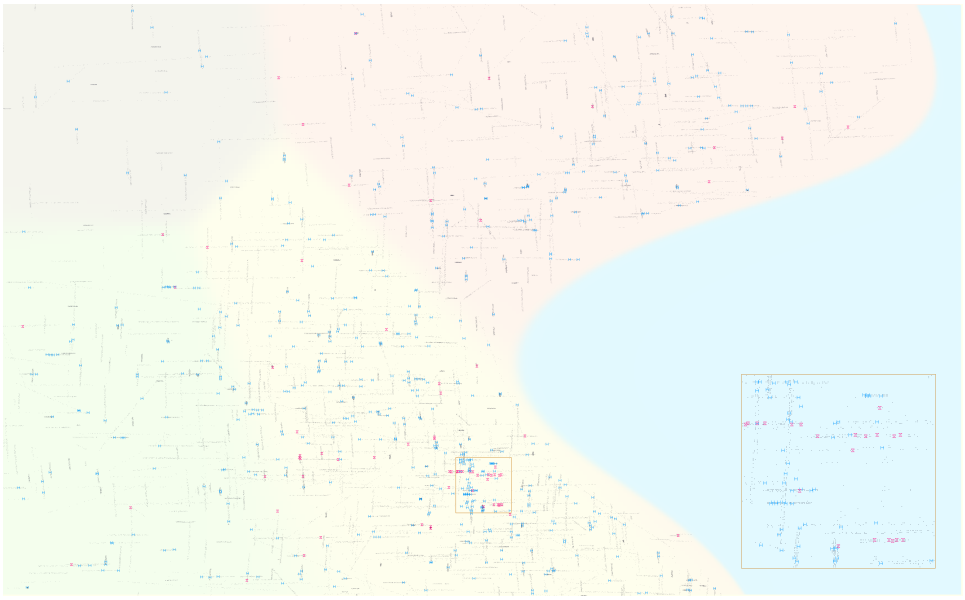

Figure 1: In light yellow and blue are land and sea: 6.4km × 10.4km around the city of Zembla. Gray dots: untested faucets. Red bowties: faucets tested positive. Blue I-beams: faucets tested negative. The small yellow box surrounds a neighborhood that enjoyed more thorough testing; the large yellow box in the lower right magnifies that extra-tested region. TODO: number of gray dots, among labels: training set vs hidden

Cool! We see that the faucets (gray dots) are strongly concentrated along north-south and east-west line segments mostly 0.2km to 0.5km in length. Do those segments reflect pipes aligned underground along roads? Or perhaps they arose as an artifact, if these coordinates were inferred from city records of population density and of utility bills organized by street address? Ideally we'd ask the city to clarify. But let's say they ignored our emails; then we'll do an analysis agnostic to the exact interpretation of these segments.

There seems to be some large-scale spatial dependence: contamination seems most prevalent in the bottom third of the map; it seems least prevalent in the middle third. Inspecting the zoomed section, we notice some small-scale spatial dependence, too: it seems plausible that faucet labels are highly correlated along each segment.

Let's develop a simple model. Each faucet f belongs to a member $s(f) \in S$ of the set of segments. Each faucet f has scaled coordinates $\vec{x}(f) \in [-1,+1]^2$, and likewise for $\vec{x}(s)$ for the midpoint of segment s.

$$p_s \sim \text{Beta}(q_{\vec{x}(s)}, 1 - q_{\vec{x}(s)}) \qquad \ell_f \sim \text{Bern}(p_{s(f)})$$

Here, q is some unknown function that tells us for each location $\vec{x}$ how likely a faucet near that location is contaminated. We imagine q as smoothly varying and as sampled from some translation-invariant distribution over functions. It's thus natural to model q as generated by a fourier series with random coefficients indexed by pairs $\vec{k}$ of integers:

$$c_{\vec{k}} \sim \text{Gauss}(0, \nu_{\vec{k}}) + i\text{Gauss}(0, \nu_{\vec{k}}) \qquad q_{\vec{x}} = \sigma\left[\mathfrak{Re} \sum_k c_{\vec{k}} \cdot \exp(\pi i \vec{k} \cdot \vec{x})\right]$$

A natural choice for the variances $\nu_{\vec{k}}$ is a power law like $\nu_{\vec{k}} = 1/(1+|k|)$; that's on the boundary of the energy diverging. This induces a brownian-motion-like Gaussian Process.

FEATURIZATION AND FITTING —

MODEL SELECTION —

PREDICTION —

OVERFITTING —

pairing flavors

VISUALIZATION —

MODELING AND FITTING —

REGULARIZATION —

EXPLANATION —

DOMAIN ADAPTATION —

ancient tablets

VISUALIZATION —

TRANSCRIPTION —

CLEANING AND IMPUTATION —

MODELING AND FITTING —

GENERATION —

statistics

bayesian inference

So little of what could happen does happen.
— salvador dali

CONCEPTUAL FRAMEWORK — We're confronted with an observation or dataset o that comes from some unknown underlying pattern h. We know how each possible value h for h induces a distribution on o and we have a prior sense of which hs are probable. Bayes' law helps us update this sense to account for the dataset by relating two functions of h:

$$\underbrace{p_{h|o}(h|o)}_{\text{posterior}} \propto \underbrace{p_{o|h}(o|h)}_{\text{likelihood}} \cdot \underbrace{p_h(h)}_{\text{prior}}$$

Bayes' law underlies our analyses throughout these notes. Like Newton's $F = ma$, Bayes is by itself inert: to make predictions we'd have to specify our situation's forces or likelihoods. Continuing the metaphor, we will rarely solve our equations exactly; we'll instead make approximations good enough to build bridges and swingsets. Still, no one denies that $F = ma$ orients us usefully in the world of physics. So it is with the law of Bayes.

Formally, we posit a set $\mathcal{H}$ of *hypotheses*, a set $\mathcal{O}$ of possible *observations*, and a set $\mathcal{A}$ of permitted *actions*. We assume as given a joint probability measure $p_{o,h}$ on $\mathcal{O} \times \mathcal{H}$ and a *cost function* $c : \mathcal{A} \times \mathcal{H} \to \mathbb{R}$. That cost function says how much it hurts to take the action $a \in \mathcal{A}$ when the truth is $h \in \mathcal{H}$. Our primary aim is to construct a map $\pi : \mathcal{O} \to \mathcal{A}$ that makes the expected cost $\mathbb{E}_{h,o}\, c(\pi(a); h)$ small.

Below are three examples. In each case, we're designing a robotic vacuum cleaner: $\mathcal{H}$ contains possible floor plans; $\mathcal{O}$, possible readings from the robot's sensors. The examples differ in how they define and interpret $\mathcal{A}$ and c.

A. $\mathcal{A}$ consists of probability distributions over $\mathcal{H}$. We regard $\pi(o)$ as giving a posterior distribution on $\mathcal{H}$ upon observation o. Our cost $c(a; h)$ measures the surprise of someone who believes a upon learning that h is true. Such *inference problems*, being in a precise sense universal, pose huge computational difficulties; we thus often collapse distributions to points, giving rise to the distinctive challenge of balancing estimation error with structural error.

B. $\mathcal{A}$ consists of latitude-longitude pairs, interpreted as a guessed location of the robot's charging station. The cost $c(a; h)$ measures how distant our guess is from the truth. Such *estimation problems* abound in science and engineering; they pose the distinctive challenge of balancing sensitivity-to-misleading-outliers against sensitivity-to-informative-datapoints.

C. $\mathcal{A}$ consists of instructions we may send to the motors, instructions that induce motion through our partially-known room. The cost $c(a; h)$ incentivizes

motion into dusty spaces and penalizes bumping into walls. We often compose such *decision problems* sequentially; this gives rise to the distinctive challenge of balancing exploration with exploitation.

FREQUENTISM AND CHOICE OF PRIOR — Our engineering culture prizes not just *utility* but also *confidence*, since strong guarantees on our designs allow composition of our work into larger systems: equality, unlike similarity, is transitive. For example, we'd often prefer a 99% guarantee of adequate performance over a 90% guarantee of ideal performance. This asymmetry explains our pessimistic obsession with worst-case bounds over best-case bounds, cost functions over fitness functions, and simple models with moderate-but-estimatable errors over rich models with unknownable-but-often-small errors.

The *frequentist* or *distribution-free* style of statistics continues this risk-averse tradition. In the fullest instance of this style, we do inference as if the true unknown prior on $\mathcal{H}$ is chosen adversarially. That is, we try to find π that makes the following error small:

$$\max_{p_h} \mathbb{E}_{h\sim p_h(\cdot)} \mathbb{E}_{o\sim p_o(\cdot|h)}\, c(\pi(o); h)$$

Intuitively,

P-HACKING —

HIDDEN ASSUMPTIONS —

(multiple) hypothesis testing

Let's now consider the case where $\mathcal{H}$ is a small and finite. We

covariance, correlation, least squares

The key to success is failure.
— michael j. jordan

gradient descent

In what follows, `init` returns a point in $\mathcal{P}$, `rate` is a small positive real number, `time` is a large natural number, and `loss` is a function : $\mathcal{P} \to \mathbb{R}$ amenable to differentiation. An important hidden input to gradient descent is the choice of transpose function; this function converts row vectors to column vectors and thus biases learning just as a choice of svm kernel does.

```
def gd(init, rate, time, loss):
  θ = init()
  for t in range(time):
    θ -= rate * (∇loss(θ))^transpose
  return θ
```

SMOOTHNESS —

DEEP LEARNING —

OPTIMIZERS —

INITIALIZATION —

EXPECTATION MAXIMIZATION —

high dimensions

what is it like to live in high dimensions?

WEIRD BALLS —

CONCENTRATION —

SPARSITY, RANK, SUBMANIFOLDS —

VISUALIZATION —

classification and clustering

graphical models

tool of the trade: boosted trees

networks

time series

gaussian processes

appendix

notation and math refresher

PROBABILITY AND MATRIX NOTATION — We've tried to use

sans serif for the names of random variables,

italics for the values they may take, and

$\mathcal{CURLY\ CAPS}$ for sets of such values.

For example, we write $\mathsf{p}_{\mathsf{y}|\mathsf{h}}(\mathsf{y}|\mathsf{h})$ for the probability that the random variable y takes the value y conditioned on the event that the random variable h takes the value h. Likewise, our notation $\mathsf{p}_{\hat{\mathsf{h}}|\mathsf{h}}(\mathsf{h}|\mathsf{h})$ indicates the probability that the random variables $\hat{\mathsf{h}}$ and h agree in value given that h takes a value $\mathsf{h} \in \mathcal{H}$.

PROBABILITY —

LINEAR ALGEBRA —

CALCULUS —

CONCENTRATION —

python programming refresher

SETUP —

STATE AND CONTROL FLOW —

INPUT/OUTPUT —

It is written that animals are divided into (a) those belonging to the emperor; (b) embalmed ones; (c) trained ones; (d) suckling pigs; (e) mermaids; (f) fabled ones; (g) stray dogs; (h) those included in this classification; (i) those that tremble as if they were mad; (j) innumerable ones; (k) those drawn with a very fine camel hair brush; (l) et cetera; (m) those that have just broken the vase; and (n) those that from afar look like flies.
— jorge luis borges

... [to treat] complicated systems in simple ways[,] probability ... implements two principles[:] [the approximation of parts as independent] and [the abstraction of aspects as their averages].
— michael i. jordan

Doing ensembles and shows is one thing, but being able to front a feature is totally different. ... there's something about ... a feature that's unique.
— michael b. jordan

NUMPY —

beyond the i.i.d. hypothesis

OUT-OF-DISTRIBUTION TESTS —

DEPENDENT SAMPLES —

CAUSALITY —