Working with Data (6.419x)

What tools might we use to extract, extrapolate, and explain patterns in data? We assume enough familiarity with probability that we may leave technical measure-theory hypotheses implicit. We assume enough familiarity with programming that we may leave the task of implementing pseudocode as an exercise.

These notes divide into six sections: two essential sections on the statistics of high-dimensional data, then three sections illustrating algorithmic themes in three common domains, then an appendix.

statistics

bayesian abduction

AN EXAMPLE — An example. Say we have data on which water faucets in a community are contaminated. From local records, we have the coordinates of 10^6 faucets. And from random testing, we've got labels of contamination level (in $[0, 1]$) for 10^4 of those faucets. We want to predict which unlabeled faucets are contaminated. To start off with, 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. Throughout our development, we swear never to look at or touch 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*.

OK, 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 components.

...

BAYESIAN FORMALISM — We're confronted with a 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:

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

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.

More 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 $p_{\mathsf{o},\mathsf{h}}$ 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 overall expected cost $\mathbb{E}_{\mathsf{h},\mathsf{o}}\, c(\pi(\mathsf{a}); \mathsf{h})$ small.

Below are three examples. In each case, we're designing a robotic vacuum cleaner, $\mathcal{H}$ is the set of possible floor plans, and $\mathcal{O}$ consists of 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; their distinctive challenge of balancing precision with accuracy.

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

I am wiser [than he] … for … he fancies he knows something … whereas I … do not fancy I do.
— socrates

UNIFORM PRIORS — 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:

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

Intuitively,

P-HACKING —

HIDDEN ASSUMPTIONS —

(multiple) hypothesis testing

The theory of probabilities is at bottom nothing but common sense reduced to calculus; it enables us to appreciate with exactness [what we] feel with a sort of instinct …
— pierre simon laplace

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

covariance, correlation, least squares

... [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

gradient descent

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

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 —

high dimensions

what is it like to live in high dimensions?

WEIRD BALLS —

CONCENTRATION —

SPARSITY, RANK, SUBMANIFOLDS —

VISUALIZATION —

classification and clustering

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

graphical models

tool of the trade: boosted trees

networks

time series

gaussian processes

appendix

probability notation

We've tried to use

sans serif for the names of random variables,

italics for the values they may take, and

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

For example, we write $p_{\mathsf{y}|\mathsf{h}}(y|h)$ 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 $p_{\hat{\mathsf{h}}|\mathsf{h}}(h|h)$ indicates the probability that the random variables $\hat{\mathsf{h}}$ and h agree in value given that h takes a value $h \in \mathcal{H}$.

ensembles

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