Review 566 notes
Wednesday, December 4, 2019 12:50 PM

Start with hypothesis testing for a few minutes for final project.

This class takes a strong optimization perspective.
- Ultimately, our goal is to learn some

$$y \approx f(x)$$

- We take a parametric approach

$$y \approx f_w(x) = f(x; w) = f(x, w)$$

with parameter(s) w.
- What assumptions have we already made?
  inductive bias

Example

$$f(x; w) = w_0 + w_1 x + \tfrac{1}{2} w_2 x^2$$

- we have a model of how we believe the world works
  but we don't know the specifics for our given problem.
- The above is the 2nd order kinematics eqn. relating position, velocity, acceleration, and time
  $y = x_t$ position at time t.
  $w_0 = x_0$ initial position
  $w_1 = v_0$ initial velocity
  $w_2 = a$ constant acceleration
- we assume newtonian physics and constant acc. and retrieve this function class
  but what function should we use? eg what values of $w = [w_0, w_1, w_2]$?
  - could relax assumptions with

$$f(x; w) = w_0 + w_1 x + \tfrac{1}{2} w_2 x^2 + \varepsilon$$

where ε is random and non-sytematic (i.i.d) maybe it captures "jerk" and "jounce" etc. and measurement error.

- Function Class

Example

```
class Kinematic:
    def __init__(self, w0, w1, w2):
        :

    def predict(time):
        vel = self.w0 + self.w1 * time + 1/2 self.w2 * time^2
        return vel
```

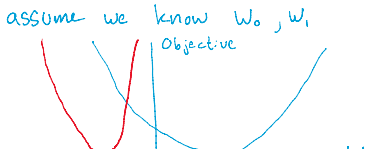
- we can have infinitely many of these
- uniquely determined by w_0, w_1, w_2

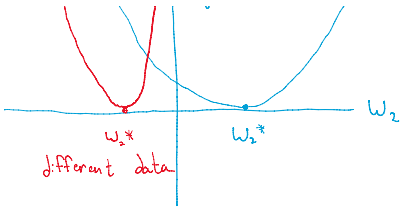
$$\text{Kinematic}(1,2,3) == \text{kinematic}(1,2,3)$$

- Optimization: how do we find the best instance of our function class?
  - what does "best" mean?
    - objective function

Example: $\| f(x; w_2) - y \|^2$

assume we know w_0, w_1



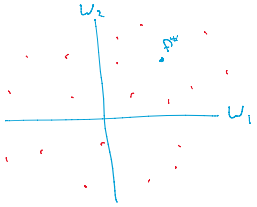


Question:
should shape change with different data?
is it always parabolic for this choice of f and objective?

- So how do we find the best:

$$f^* = \arg\min_f \mathcal{L}(f(x), y) = \arg\min_w \mathcal{L}(f(x; w), y)$$

- guess and check (random search / genetic algs etc.)



- gradient descent

$$f = f - \alpha \nabla_f \mathcal{L}$$

note this means we need to define what it means to add / multiply functions.
because f is uniquely defined by w, this is easy.

- MLE / MAP
  - MLE: $P(D|f)$ (if the world worked according to f, what are chances of seeing this (x,y) pair?)
  - MAP: $P(f|D) = \frac{P(D|f)P(f)}{P(D)}$

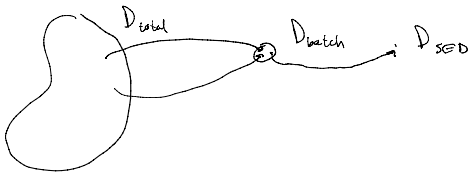
Example
if my $x_0 = 0$ and velocity = $2\ ^{km}/_{hr}$
acceleration = $0\ ^{km}/_{hr^2}$

MLE: what is probability that I'm at position 100 at t=1?

MAP: given I'm at position 100 at t=1, what is the probability the above model is correct?

$P(f)$ = I'm pretty sure my velocity is $5\ ^{km}/_{hr}$ ish

- Random Variables
  - $P(D)$ what is this?



If I ran my car long enough, I could collect all data about position vs. time
or if I ran many exactly the same cars, etc.

Instead we get small batch of D.

When we deploy we reintroduce ourselves back to big D.

- Gradients are random
  - $\nabla_w(D)\mathcal{L}$ is a function of D

Example $x \sim N(0,1)$ is random

$x+2$	is	random
$2x$	is	random
x^2	is	random
$\mathbb{E}[x]$	is	not

- how do I know ∇ is random?
  sample many data points ↑ and compute ∇ for each
  from D_{total}

- Is $\nabla(D_{SGD})$ random? yes
- Is $\nabla(D_{total})$ random? no
- Is $\nabla(D_{batch})$ random? yes according to D_{total}
  no according to D_{batch}

- Bias / Variance

Bias: $\mathbb{E}\left[\hat{x} - x\right]$

- we want $\nabla(D_{total})$, but don't have access
  - is $\nabla(D_{SGD})$ a biased estimate of $\nabla(D_{total})$?

$$\mathbb{E}\left[\nabla(D_{SGD}) - \nabla(D_{total})\right] \stackrel{?}{=} 0$$

$$\mathbb{E}\left[\nabla(D_{total})\right] = \nabla(D_{total}) \checkmark$$

$$\mathbb{E}\left[\nabla(D_S)\right] = \frac{1}{n}\sum_{i=0}^{n}\nabla(D_i) = \frac{1}{|D|}\sum_{d\in D}\nabla(d)$$

$$= \nabla(D_{batch})$$

  - is $\mathbb{E}\left[\nabla(D_{batch}) - \nabla(D_{total})\right] = 0$?

hopefully!

- what about variance?

$$\left.\begin{aligned}\mathrm{Var}(\nabla(D_S)) &= \mathbb{E}\left[\nabla(D_S)^2\right] - \mathbb{E}\left[\nabla(D_S)\right]^2 \\ &= \mathbb{E}\left[\nabla(D_S)^2\right] - \nabla(D_b)^2\end{aligned}\right\}\ \text{element-wise}$$

can also consider $\mathrm{Cov}(\nabla,\nabla)$. considering on Var for simplicity

- So what?
  - SURE: what if we use 2 estimates of $\nabla(D_b)$ to have even lower variance

$$\nabla(D_{SURE}) = \nabla(d_1) - \left(\nabla(d_2) - \mathbb{E}\left[\nabla(d_2)\right]\right)$$

$$= \nabla(d_1) - \left[\nabla(d_2) - \nabla(D_b)\right]$$

why not just use $\nabla(D_b)$ which is lowest variance?
maybe we can approximate it cheaply or compute it infrequently.

- stepsizes:

$$f = f - \alpha \nabla(d)$$

$$\alpha = \frac{1}{\text{var}(\nabla(d))} \quad \text{"divide out variance"}$$

recall variance is second centered moment. That means

$$\mathbb{E}\left[\nabla(d)^2\right] - \nabla(D_0)^2 \quad \text{if we drop "center" } \nabla(D_0)^2$$

then we get

$$\mathbb{E}\left[\nabla(d)^2\right] \approx \sum \nabla(d)^2 \approx (1-\beta)\, v + \beta\, \nabla(d)^2 \quad \leftarrow \text{rms prop !}$$

- what if model is non-linear?

$$y = w_0 + w_1 v_3 e^{-\frac{x}{v_3}} + c w_3 \left[1 - e^{-\frac{x}{v_3}}\right] + \tfrac{1}{2} w_2 x^2$$

(kinematics + linear drag) with minor simplifications

$$\mathcal{L} = \| f(x) - y \|^2$$

$$\nabla_w (f(x)-y)^2 = 2\left(f(x)-y\right) \nabla_w f(x)$$

for w_0, w_1, w_2 easy
for w_3 need product rule