Lecture 0

Monday, January 15, 2024 10:32 AM

Reading

Murphy §2

Bishop §1.2, §2

Some generic ML tasks:

Supervised learning

Given data $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$

(x_i: features, y_i: labels)

with data distribution $\mathbb{P}(x, y)$

Task: Given a new feature $\hat{x}$ from same dist, can we predict label $\hat{y}$?

Classification

Supervised learning problem where label ~~denotes~~ class ownership

ex
- from microscopy, infer elastic / plastic deform.
- from PIV data, are we in laminar, transition, or turbulent?

Unsupervised

Given unlabeled data $\mathcal{D} = \{x_i\}_{i=1}^N$

can we characterize the underlying distribution $p(\mathcal{D})$?

e.g. clustering

Generative modeling

Sample from $p(\mathcal{D})$ to generate realistic synthetic data

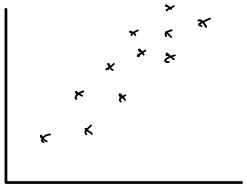
e.g. DALL-E

synthetic microstructure conditioned on stress state

Ex: Regression

Given $X = x_1, \dots x_N$

$y_i = f(x_i) + \varepsilon$ (noise)



- Choose parameterized regressor

$f_\theta(x) = \theta_1 + \theta_2 x$

and solve $\theta^* = \underset{\theta_1, \theta_2}{\operatorname{argmin}} \sum_i |y_i - f_\theta(x_i)|^2$ ($\mathcal{L}$: empirical loss)

Note that we can rewrite

$\mathcal{L} = (y - f_\theta(x))^T (y - f_\theta(x))$

in vector notation $x \in \mathbb{R}^N$, $y \in \mathbb{R}^N$,

$f_\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}^T \begin{pmatrix} 1 \\ x \end{pmatrix} = \hat{\theta}^T P(x)$

To solve minimization

$0 = \nabla_\theta \mathcal{L} = 2 P(x)^T (P(x)\theta - y)$

AKA Normal $\Rightarrow$ $P^T(x) P(x) \theta = P(x)^T y$

$$0 = \nabla_\theta \mathcal{L} = 2 P(x)^T (P(x)\theta - y)$$

AKA Normal Eqn.

$$\Rightarrow P^T(x) P(x) \theta = P(x)^T y$$

And we are left with

$$f_\theta^*(x) = (P^T(x) P(x))^{-1} P(x)^T y$$

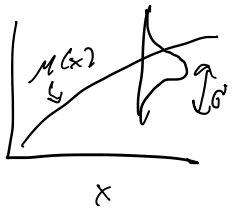
which can be used to approximate data

Probabilistic Interpretation

- To account for noise in data assume the model

$$y|x \sim \mathcal{N}(\mu_\theta(x), \sigma^2)$$

normal distribution; parameterized mean; noise param.



- To find the parameters which align

$$p(y|x, \theta, \sigma^2) \sim p(\mathcal{D})$$

we perform maximum log likelihood estimation

- Assuming IID data

$$p(y_1, \dots y_N) = \prod_{i=1}^{N} p(y_i | x_i, \theta, \sigma^2)$$

$$\log p(y_1, \dots y_N) = \sum_{i=1}^{N} \log \mathcal{N}(y = y_i | x_i, \theta, \sigma^2)$$

$$= \sum_{i=1}^{N} \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} (y_i - \mu_i)^2 \right) \right]$$

$$= -\sum_{i=1}^{N} \frac{1}{2} |y_i - \mu_\theta(x)|^2 + C(\sigma^2)$$

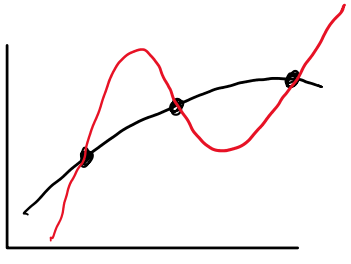
$\log p(y_1, \dots y_N)$ = LL

Taking $\mu_\theta = \theta^T P(x)$, Maximizing LL wrt θ gives identical result. To characterize noise, $\nabla_{\sigma^2} LL = 0$ gives an estimate of noise

Notes

- overfitting / underfitting

many classes of models become oscillatory when interpolating data



? Don't minimize empirical

interpolating data

? Does minimizing empirical loss give good results at new points

- curse of dimensionality
  - what space does data live in? $X \in \mathbb{R}^d$ (eg images)
  - How do the parameters in model scale w/ dimension d?
  - ex FEM $\rightarrow$ $\#\theta \sim N^{-1/d}$

Fixes
- Dimensionality Reduction
  - transform data into $d_f < < d$
- Use nonlinear approximators that scale independent of dim (DNN)

HW1
- Sample $x \sim U(0,1)$ (uniform dist) to generate a vector of 100 pts
- Calculate $y = 2x + \varepsilon$ where $\varepsilon \sim N(0, 0.01^2)$ $\sigma = 0.01$ about 1% noise
- I've provided code to estimate a deterministic model $f_\theta(x) = \theta_1 + \theta_2 x$
  - (A) By direct calculation of the normal equations
  - (B) By gradient descent
    $$\theta^{n+1} = \theta^n - \eta \nabla_\theta \mathcal{L}$$
    (η: step size chosen "sufficiently small")
- Modify the code to instead fit the noisy model
  $$y \sim N(\theta_1 x + \theta_2, \sigma^2)$$

2 registers

$$y \sim N(\theta_1 x + \theta_2, \sigma^2)$$

2 registers

identifying the parameters $\theta_1, \theta_2, \sigma^2$ from the negative log likelihood by

2^{-23}

(A) Setting $\nabla_{\begin{pmatrix}\theta_1\\ \theta_2\\ \sigma^2\end{pmatrix}} LL = 0$ and solving the normal equations

(B) Applying gradient descent