ESE 402/542 Final Review

Core Concepts in the Second-Half

- Regression: least-squares.
- Classification: Logistic Regression, Linear Discriminant Analysis (LDA).
- Unsupervised Learning: k-means clustering, Principal Components Analysis (PCA).
- A Unified Framework: Probably-Approximately-Correct (PAC) Learning. VC-dimension.

Notes on the Final

- More emphasis on general understanding of methods discussed in class: motivations, where to use them, limitations.
- Should be *little* computational math once question is understood.

Classification: Logistic Regression

- Motivation: given data points $\{x_i\}$, want to assign labels $y_i \in \{0, 1\}$ (can be extended beyond two-class) to each point.
- Simple approach: fit probabilities $\mathbb{P}[y_i = 0|x]$ (probability of $y_i = 1$ follows). Whichever probability is greater for y_i, we choose that label.
- How to fit these probabilities as a function of x? First attempt: linear!

$$\mathbb{P}[y = 0|X = x] = \beta_0 + \beta_1 x.$$

What's the problem? Linear fitting can be negative, can be greater than 1. No way to enforce these constraints without ruining simplicity. Have to do something (slightly) smarter.

Classification: Logistic Regression

- What's a way to parameterize a probability? Exponential function is natural choice to ensure non-negativity. Normalizing is way to ensure $\ell 1$:

$$\begin{aligned} \mathbb{P}\left[y=0 \mid X=x\right] &= \frac{\exp\left(\beta_0+\beta_1 x\right)}{1+\exp\left(\beta_0+\beta_1 x\right)} \\ \implies \mathbb{P}\left[y=1 \mid X=x\right] &= \frac{1}{1+\exp\left(\beta_0+\beta_1 x\right)} \end{aligned}$$

- Logistic regression problem is now to fit parameters β_0 and β_1.
- Natural way to do this? Maximum likelihood estimation.

Classification: LDA

- Broad idea: assuming that the conditional distribution of data x given fixed label $y = 0$ (accordingly, $y = 1$) is Gaussian.
- Main simplifying assumption: covariance matrix of $y = 0$ and $y = 1$ conditional distributions are assumed to be identical (if not, leads to Quadratic Discriminant Analysis).
- Solve for LDA using Maximum Likelihood Estimation. Check class notes for details.

PCA

- Have data $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$, assume 0-mean (o.w. center it)
- Want to project each x_i to a lower dimensional representation $z_i \in \mathbb{R}^{d'}$ using a linear transformation
- Let $\Sigma = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^\top$ be the covariance matrix of the data
- Use SVD $\Sigma = U \Lambda U^\top$ where $\Lambda = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2)$
- Pick the first d' eigenvectors (columns of U)
- This is the basis to project onto
- Total energy (variance of centered data) is given by $\sum_{i=1}^n \sigma_i^2$
- Variance explained is given by $\sum_{i=1}^{d'} \sigma_i^2$
- Error of projection is given by $\sum_{d'+1}^n \sigma_i^2$

Vapnis-Chervonenkis (VC) Dimension

- General question: why don't we just use very expressive functions to perfectly classify everything?
- Intuition developed so far: we should worry about generalization error of such classifiers (overfitting).
- What is a way to capture "expressivity" of a function class? Enter the VC dimension.

Vapnis-Chervonenkis (VC) Dimension

- Given classifier function class $\mathcal{H} = \{h(\cdot)\}$ (e.g. linear predictors, logistic classification, neural nets etc), quantify its expressivity by "shattering number".
- Shattering number: if $\mathcal{H}$ has shattering number at least n, means for *any* choice of n data points $\{x_i\}_{i=1}^n$, and for *any* labels $\{y_i\}_{i=1}^n$, there exists a function $h \in \mathcal{H}$ such that

  $$h(x_i) = y_i \quad \text{for all } i = 1, \ldots, n.$$

  VC dimension can be thought of as the largest shattering number $\mathcal{H}$ can achieve.
- Idea: if function class has high VC dimension, it can perfectly classify *any* labelling for *any* large dataset. This might arouse suspicion, as this includes *randomly assigned* labels.

PAC

- “Probably-Approximately-Correct” Learning. To break it down: fixing failure probability $\delta \in [0, 1]$, we want to know the a lower bound on the dataset size n such that taking probability over all datasets S of size n,

$$\mathbb{P}_S\Big[\underbrace{|\mathrm{err}(h_S) - \mathrm{err}(h^*)| < \varepsilon}_{\text{“Approximately Correct”}}\Big] \underbrace{\geq 1 - \delta}_{\text{“Probably”}}$$

- General idea of PAC: want to ensure that *empirical* solution, i.e. any classifier/function fit that we learn from finite amounts of data, gets comparable performance to the *best* choice of function (Bayes optimal classifier), at least most of the time. This is a theoretical quantification of generalization performance.

2019SP P1 Parts 1-3

- done in lecture/recitation (MLE under normally distributed noise is equivalent to least-squares)

2019SP P1 Part 4

Problem: Suppose $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$, where $\epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_i^2)$. Find the MLE of β_0, β_1.
Solution: $y_i \sim \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma_i^2)$, so
$f(y_i) = \frac{1}{\sqrt{2\pi\sigma_i}} \exp\left(-\frac{1}{2\sigma_i^2}(y_i - \beta_0 - \beta_1 x_i)^2\right)$

$$\max_{\beta_0,\beta_1} \sum \log f(y_i) = \max_{\beta_0,\beta_1} \sum -\frac{1}{\sigma_i^2}(y_i - \beta_0 - \beta_1 x_i)^2$$

(constant terms don't depend on betas)

$$= \min_{\beta_0,\beta_1} \sum \frac{1}{\sigma_i^2}(y_i - \beta_0 - \beta_1 x_i)^2$$

2019SP P1 Part 4 cont.

Take derivatives...

2019SP P2 Part 1

Problem: Solve $\min_{\boldsymbol{c}} \sum_{i=1}^{n} w_i \|\boldsymbol{x}_i - \boldsymbol{c}\|_2^2$
Solution: Take derivative w.r.t $\boldsymbol{c}$ and set to 0.

$$\sum_i 2w_i(\boldsymbol{x}_i - \boldsymbol{c}) = 0$$

$$\sum_i w_i \boldsymbol{x}_i - \boldsymbol{c} \sum_i w_i = 0$$

$$\boldsymbol{c} = \frac{\sum_i w_i \boldsymbol{x}_i}{\sum_i w_i}$$

2019SP P2 Part 2

Problem: Extend k-means to the weighted setting:

$$\sum_{i=1}^{n} w_i \min_{j \in [K]} \|\boldsymbol{x}_i - \boldsymbol{c}_j\|_2^2$$

Solution: Each data point has a weight w_i. Alternate finding centroids and assigning points to clusters, except with weights. So to find centroids, instead of setting $\boldsymbol{c}_j$ to the average of the points in the clusters, use the weighted mean (in previous part). To assign points to clusters, this remains the same as before.

2019SP P3

Given in HW7, and in previous slide.

2020FA P2 Part 1

Q: Given $P(Y=+1) = 1 - P(Y=-1) = 3/4$, $P(X=x|Y=y) = \frac{1}{2}e^{-|x-2y|}$, find $P(X=x, Y=y)$
A: Y is Bernoulli(3/4). $P(X=x, Y=y) = P(Y=y)P(X=x|Y=y) = (3/4)^y(1/4)^{1-y} \cdot \frac{1}{2}e^{-|x-2y|}$.

2020FA P2 Part 2

Q: Plot $P(X = x, Y = +1)$, $P(X = x, Y = -1)$.
A: Two Laplace distributions, one centered at 2, the other at -2.
The one centered at 2 should be larger.

2020FA P2 Part 3

Q: Derive Bayes optimal classifier.
A: Find $\text{argmax}_y P(Y = y|X = x) = \text{argmax}_y P(Y = y|X = x)P(X = x) = \text{argmax}_y P(Y = y, X = x) = \text{argmax}_y \{\frac{3}{8}e^{-|x-2|}, \frac{1}{8}e^{-|x+2|}\}$ where the first option corresponds to $y = +1$, second is $y = -1$.
Decision rule: $\frac{3}{8}e^{-|x-2|} \underset{-1}{\overset{+1}{\gtrless}} \frac{1}{8}e^{-|x+2|}$. Taking logs and stuff:
$\log 3 - |x-2| \underset{-1}{\overset{+1}{\gtrless}} -|x+2|$. By the plot, only need to consider when $x \in [-2, 0]$. Then, $\log 3 - (-x+2) \underset{-1}{\overset{+1}{\gtrless}} -x-2 \implies x \underset{-1}{\overset{+1}{\gtrless}} -\frac{\log 3}{2}$.

2020FA P2 Part 4

Q: Error rate of Bayes optimal?

A: Let $c = -\frac{\log 3}{2}$. $P(h^*(x) \neq y) = P(X > c | Y = -1) \cdot 1/4 + P(X < c | Y = +1) 3/4 = \frac{1}{4}\Phi(-2) + \frac{3}{4}\Phi(-2)$ where Φ is the Laplace(0,1) CDF.

2020FA P2 Part 5

QDA (will not be covered)

2020FA P2 Part 6

QDA

2020FA P2 Part 7

QDA

2020FA P2 Part 8

QDA

2020FA P3 Part 1

Q: Find the VC dimension of $\mathcal{H}_1$.
A: We claim the VC dimension is 1. To see this, we observe that any 1 point can be classified correctly. Without loss of generality, let $x = c$ for any real number $c \in \mathbb{R}$. Then if the corresponding label is $y = 1$, we can simply set $a = c$ for our classifier. If the corresponding label is $y = 0$, we can set $a = c + 10^6$ (no real significance to the number, just ensures the classifier chooses $h(c) = 0$).
On the other hand, if we have 2 data points, we observe that for any two data points labelled $y = 1$, they must necessarily satisfy $|x_1 - x_2| \leq 6$, since the only two intervals on which any classifier classifies $h(x) = 1$ have distance upper bounded by 6. So, all we need to do is select $x_1 = c$, and $x_2 = c + 10^6$, so that no single classifier can find $h(x_1) = h(x_2) = 1$ simultaneously.

2020FA P3 Part 2

Q: Find the VC dimension of $\mathcal{H}_2$.
A: 0. As defined, for any $h_{a,b}$, the origin $x = 0$ is necessarily classified as $h_{a,b}(x) = 1$. Therefore, let us consider one point placed at the origin with label $y = 0$. No function $h_{a,b}$ will be able to classify this point correctly.