# CSC2515 Midterm Review
# Part 2

Haoran Zhang  Sicong Huang

Oct 20, 2020

# Midterm Information

- Thursday Oct 22 11:59am EDT to Friday Oct 23 11:59am EDT
- Designed to take 2.5 to 3 hours for well-prepared students
- What might be on the midterm?
  - Everything covered in detail during lecture
- What will NOT be on the midterm?
  - Programming
  - New concepts introduced in Homeworks
  - Anything from tutorials that wasn't in the lectures slides
  - Anything from the textbooks that wasn't in the lecture slides

# Midterm Information (Cont'd)

- Week 7 **is** on the midterm.
- Piazza will be set to "private posts only" during the midterm.
- Please refrain from starting discussions in existing threads on Piazza.
- Submission (detailed instructions on midterm):
  - (recommended) print and scan
  - (not recommended) write from scratch; LaTeX
- Leave plenty of time at the end for submission.
- Open book
  - Allowed: all lecture slides, course notes, textbooks, Piazza
  - Not allowed: Google, any computational software (Ex: Wolfram Alpha, graphing tools)
- SGS course drop deadline: Monday October 26

# Midterm Topics (Last Time)

- Lecture 1
  - k-Nearest Neighbors
  - Bayes Optimality
- Lecture 2
  - Decision trees and Information theory (information gain)
  - Bias Variance
  - Bagging
- Lecture 3
  - Linear regression
  - Logistic regression

# Midterm Topics (Last Time)

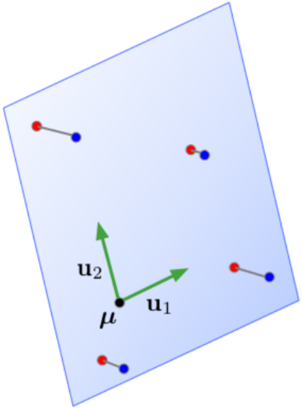
- Lecture 4
  - Gradient descent
  - $L^1$,$L^2$ regularization (covered in previous tutorial, see Q3 in 19 midterm)
  - SVMs (covered in previous tutorial, see Q6 in 19 midterm)
  - Boosting, additive models

# Midterm Topics (This Time)

- Lecture 5
  - PCA
  - K-means
  - Maximum likelihood estimation (MLE)
- Lecture 6
  - Maximum a-posteriori (MAP)
  - Full Bayesian parameter estimation
  - Naive Bayes
  - Gaussian discriminant analysis
- Lecture 7
  - EM algorithm
  - Gaussian mixture models

# PCA

- **Goal**: reduce $\boldsymbol{x} \in \mathbb{R}^D$ to $\boldsymbol{z} \in \mathbb{R}^K$
- **Idea**: find orthonormal basis $\boldsymbol{U} \in \mathbb{R}^{D \times K}$
- $z = \boldsymbol{U}^T(\boldsymbol{x} - \boldsymbol{\mu})$
- Solving for $\boldsymbol{U}$: argmin of reconstruction error = argmax of code vector variance
- **Solution**: columns of $\boldsymbol{U}$ are eigenvectors of $\boldsymbol{\Sigma}$ with top $K$ eigenvalue magnitudes.



# PCA Question

Show that PCA is translationally invariant. Shifting all data $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta}$ does not change the principal components or the code vectors.

$$\mu' = \mu + \delta$$

$$\Sigma' = \frac{1}{N} \sum_{i=1}^{N} (x^{(i)'} - \mu')(x^{(i)'} - \mu')^T$$

$$= \frac{1}{N} \sum_{i=1}^{N} (x^{(i)} - \mu)(x^{(i)} - \mu)^T$$

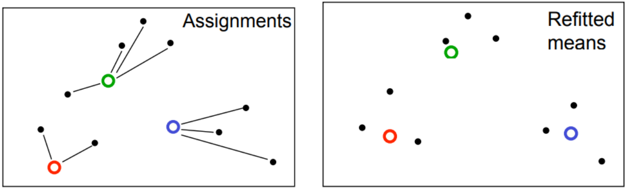
$$= \Sigma$$

1) $U' = U$

2) $z' = U^T(x' - \mu') = U^T(x - \mu)$

# K-Means

- **Goal**: reduce $\boldsymbol{x} \in \mathbb{R}^D$ to $\boldsymbol{z} \in \{1, ..., K\}$
- $\min_{\mu_j, S_j} \sum_{j=1}^{K} \sum_{x_i \in S_j} ||x_i - \mu_j||^2$
- Two steps:
  - Assignment: $x_i \in S_k \longleftrightarrow k = arg \min_j ||x_i - \mu_j||^2$
  - Refitting: $\mu_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$



# K-Means Question

Assume that cluster assignments are fixed. Show that, for one specific value of the learning rate $\alpha$, the "refitting" step is equivalent to performing batch gradient descent on the original loss function.

$$L = \sum_{j=1}^{K} \sum_{x_i \in S_j} ||x_i - \mu_j||^2$$

$$\frac{\partial L}{\partial \mu_1} = -2 \sum_{x_i \in S_1} (x_i - \mu_1)$$

$$\mu_1 \leftarrow \mu_1 + \alpha \sum_{x_i \in S_1} (x_i - \mu_1)$$

Refitting $\mu_1 \leftarrow \frac{1}{|S_1|} \sum_{x_i \in S_1} x_i$

$$\frac{1}{|S_1|} \sum_{x_i \in S_1} x_i = \mu_1 + \alpha \sum_{x_i \in S_1} (x_i - \mu_1)$$

$$\Downarrow$$

$$\mu_1 = \frac{1}{|S_1|} \sum_{x_i \in S_1} \mu_1$$

$$\alpha = \frac{1}{|S_1|}$$

$\left( \frac{1}{2|S_1|} \text{ also okay} \right)$

# MLE and MAP

MLE

- $\theta^* = arg\max_\theta p(D|\theta) = arg\max_\theta \log p(D|\theta) = arg\max_\theta \sum_{i=1}^{N} \log p(D_i|\theta)$

MAP

- $\theta^* = arg\max_\theta p(\theta|D) = arg\max_\theta \frac{p(\theta)p(D|\theta)}{p(D)}$
  $= arg\max_\theta \log p(\theta) + \log p(D|\theta)$
- How do we choose a good prior? Conjugate, effective sample size

These both give point estimates for $\theta^*$!

# Full Bayesian

- $p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} = \frac{1}{Z}p(\theta)p(D|\theta) \propto p(\theta)p(D|\theta)$
- $Z = \int p(\theta)p(D|\theta)\ d\theta$ (normalization constant)
- Instead of a point estimate, now we have a distribution for $p(\theta|D)$
- Can compute the probability distribution over the next data point:

$$p(D'|D) = \int p(\theta|D)p(D'|\theta)\ d\theta$$

# MLE/MAP/Full Bayesian T/F

- True/False - The MAP and MLE estimates can only be equal when the number of training examples is very large.

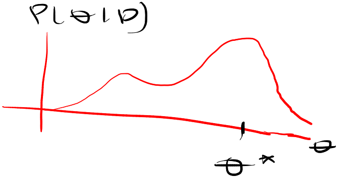
data: 3H, 2T $\longrightarrow \theta^*_{MLE} = 2/5$

Prior ~ Beta(3, 4) $\longrightarrow \theta^*_{MAP} = 2/5$

- True/False - MAP computes the mean of the posterior distribution.

mode

$$\theta^* = \arg\max_\theta p(\theta \mid D)$$



# MAP Question

Consider the following procedure used to generate random integers between 0 and $2^k - 1$ (inclusive):

- Start with the set of all integers between 0 and $2^k - 1$ .
- (*) Flip a biased coin with probability of heads $= \alpha$
    - If it is a head, remove the first half of the (remaining) numbers.
    - If it is a tail, remove the second half of the (remaining) numbers.
    - If only one number is left, return that number
    - Otherwise, go back to step (*).

# MAP Question

a) For a particular outcome $b$, let $n_1(b)$ be the number of 1's in the binary expansion of $b$, and $n_0(b)$ be the number of 0's. What is the likelihood of $b$ given $\alpha$?

$K = 4$ :

HTHT

1010

$2^3 + 2^1$

$$\alpha^{n_1(b)} (1-\alpha)^{n_0(b)}$$

$$= \alpha^{n_1(b)} (1-\alpha)^{K - n_1(b)}$$

# MAP Question

b) In order to estimate $\alpha \in [0, 1]$, we generate *n* random numbers. We assume the following prior distribution for $\alpha$: $p(\alpha) = 6\alpha(1 - \alpha)$. What is the MAP estimate for $\alpha$ using these *n* observations?

Beta(2, 2)

# MAP Question - Solution

$$\alpha^* = \arg\max_\alpha \log p(D|\alpha) + \sum_{i=1}^{n} \log p(D_i|\alpha)$$

$$= \arg\max_\alpha \log(6\alpha(1-\alpha)) + \sum_{i=1}^{n} \log(\alpha^{n_{1,i}}(1-\alpha)^{k-n_{1,i}})$$

$$= \arg\max_\alpha \log(\alpha) + \log(1-\alpha)$$

$$+ \log(\alpha)\sum_{i=1}^{n} n_{1,i} + \log(1-\alpha)(kn - \sum_{i=1}^{n} n_{1,i})$$

$$= \arg\max_\alpha (\log\alpha)(1 + \sum_{i=1}^{n} n_{1,i}) + \log(1-\alpha)(1 + kn - \sum_{i=1}^{n} n_{1,i})$$

# MAP Question - Solution

$$\frac{\partial}{\partial \alpha} = \frac{1 + \sum_{i=1}^{n} n_{1,i}}{\alpha} - \frac{1 + kn - \sum_{i=1}^{n} n_{1,i}}{1 - \alpha} = 0$$

$$(1 + \sum_{i=1}^{n} n_{1,i}) - \alpha(1 + \sum_{i=1}^{n} n_{1,i}) - \alpha(1 + kn - \sum_{i=1}^{n} n_{1,i}) = 0$$

$$\alpha^* = \frac{1 + \sum_{i=1}^{n} n_{1,i}}{kn + 2}$$

# Naive Bayes

- $p(t = k | x_1, ..., x_D) \propto p(t = k, x_1, ..., x_D)$

$$= p(t = k) p(x_1, ..., x_D | t = k)$$

$$= p(t = k) \prod_{j=1}^{D} p(x_j | t = k) \leftarrow$$

- Learn $p(x_j | t = k)$ separately (ex: by MLE)
  - Bernoulli Naive Bayes $\quad x_j | t = k \sim Ber(\theta)$
  - Gaussian Naive Bayes

- $p(t = k | x_1, ..., x_D) = \frac{1}{Z} p(t = k) \prod_{j=1}^{D} p(x_j | t = k)$

- $Z = p(\boldsymbol{x}) = \sum_k p(t = k) p(x_1, ..., x_D | t = k)$

# GDA

- If $\boldsymbol{x}$ is continuous, instead of making the Naive Bayes assumption, we can model

$$p(x_1, ..., x_D | t = k)$$

  by a multivariate Gaussian.
- $\boldsymbol{x} | t = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- Can compute $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ using MLE.

# GDA

For the binary classification case:

- General $\boldsymbol{\Sigma_k}$ : conic section
- $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$: linear decision boundary
- $\boldsymbol{\Sigma_k}$ diagonal: Gaussian Naive Bayes
- $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \sigma^2 I$: decision boundary bisects class means

# Naive Bayes Question

You are doing binary classification on a dataset with two features using a Naive Bayes classifier. You compute $p(x_j|t = k)$ as the following categorical distributions. Assume the two classes are equally likely.

| | $t = 0$ | $t = 1$ |
|---|---|---|
| $x_1 = -1$ | 0.2 | 0.3 |
| $x_1 = 0$ | 0.4 | 0.6 |
| $x_1 = 1$ | 0.4 | 0.1 |

| | $t = 0$ | $t = 1$ |
|---|---|---|
| $x_2 = -1$ | 0.4 | 0.1 |
| $x_2 = 0$ | 0.5 | 0.3 |
| $x_2 = 1$ | 0.1 | 0.6 |

For a data point $x = (-1, 1)$, calculate $p(t = 0|x)$ and $p(t = 1|x)$

$$p(t=0, x_1=-1, x_2=1)$$

$$= p(t=0)\, p(x_1=-1 \mid t=0)\, p(x_2=1 \mid t=0)$$

$$= (0.5)(0.1)(0.2)$$

$$= 0.01$$

$$p(t=1, x_1=-1, x_2=1)$$

$$= (0.5)(0.3)(0.6)$$

$$= 0.09$$

$$p(t=0 \mid x) = \frac{0.01}{0.01+0.09} = 0.1$$

$$p(t=1 \mid x) = 0.9$$