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
  - 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)

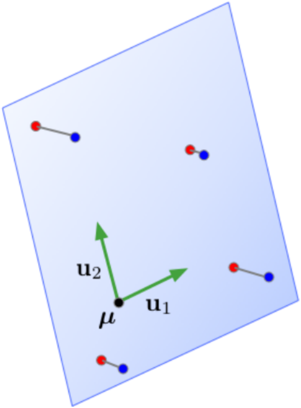
- Lecture 4
  - Gradient descent
  - L^1,L^2 regularization (covered in previous tutorial, see Q3 in 19 midterm)
  - SVMs (covered in previous tutorial, see Q6 in 19 midterm)
  - Boosting, additive models

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.

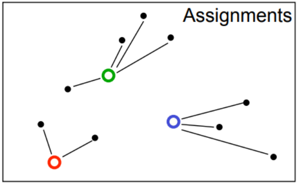


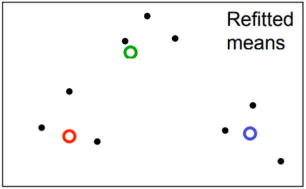
PCA Question

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.

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 α, the "refitting" step is equivalent to performing batch gradient descent on the original loss function.

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?

These both give point estimates for θ^*!

Full Bayesian

- $p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} = \frac{1}{Z}p(\theta)p(D|\theta) \propto p(\theta)p(D|\theta)$
- $Z = \int p(\theta)p(D|\theta)\ d\theta \quad$ (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.

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

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 (0), remove the first half of the (remaining) numbers.
  - If it is a tail (1), 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 α?

MAP Question

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

Naive Bayes

- $p(t = k | x_1, ..., x_D) \propto p(t = k, x_1, ..., x_D)$
$$= p(t = k) p(x_1, ..., x_D | t = k)$$
$$= p(t = k) \prod_{j=1}^{D} p(x_j | t = k)$$
- Learn $p(x_j | t = k)$ separately (ex: by MLE)
  - Bernoulli Naive Bayes
  - 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 $p(t=1|x)$