Review for Midterm Exam 2

Chapters 5—8
(Need to still know chapters1—4)

CMPUT 267: Basics of Machine Learning

Logistics

- Midterm Exam 2 **during class on Tuesday March 26 in the usual classroom**
- Formula sheet provided (on the course site already). It will be printed for you - do not bring your own.
- The practice exam and the real exam are similar. Please review the practice exam!
  - But they are definitely not the same. Do not simply try to pattern match. You need to understand the practice exam, and be able to apply that knowledge.
  - The exam is meant to test the basics, not to challenge you; answers can be short (the exam is short so each question is worth a lot)

Midterm Details

- The content is up to Chapter 8 (Linear and Polynomial Regression), and up to and including the March 8 lecture
- The focus is on Chapters 5-8, but Chapter 1-4 are important background

Language of Probabilities

- Define random variables, and their distributions
  - So that we can formally reason about data and estimators
- Express our beliefs about behaviour of these RVs, and relationships to other RVs
- Examples:
  - p(x) Gaussian means we believe X is Gaussian distributed
  - p(y | X = x)—or written p(y | x)— is Gaussian means that when conditioned on x, y is Gaussian
  - p(w) and p(w | Data)

Very brief summary of Ch 1-4

- Probability
- Estimators
- Optimization

Probability

- Define a **random variable**
- Define **joint** and **conditional probabilities** for continuous and discrete random variables
- Define **probability mass functions** and **probability density functions**
- Define **independence** and conditional independence
- Define **expectations** for continuous and discrete random variables
- Define **variance** for continuous and discrete random variables

Probability (2)

- Represent a problem probabilistically
  - e.g., how likely was the outcome?
- Use a provided distribution
  - I will always remind you of the density expression for a given distribution
- Apply **Bayes' Rule** to manipulate probabilities

Estimators

- Define **estimator**
- Define **bias**
- **Demonstrate that an estimator is/is not biased**
- Derive an expression for the variance of an estimator
- Define **consistency**
- Demonstrate that an estimator is/is not consistent
- Justify when the use of a **biased estimator** is **preferable**

Poll Question: When is the use of a biased estimator preferable?

- 1. It is always better because it biases towards the true solution
- 2. If the bias reduces the mean-squared error by reducing the variance
- 3. If the bias reduces the mean-squared error by increasing the variance
- 4. It is rarely justifiable

Answer: 2

Summary

- **Concentration inequalities** let us bound the probability of a given estimator being at least ϵ from its mean (expected value). $\Pr\left(\left|\bar{X} - \mu\right| \leq \epsilon\right) \geq 1 - \delta$
- **Sample complexity** is the **number of samples** needed to attain a desired error bound ϵ at a desired probability $1 - \delta$
  - We only discussed sample complexity for unbiased estimators
- The **mean squared error** of an estimator **decomposes** into **bias** (squared) and **variance**
- Using a **biased** estimator can have **lower error** than an unbiased estimator
  - Bias the estimator based on some **prior information**
  - *But this only helps if the prior information is **correct,** cannot reduce error by adding in arbitrary bias*

Optimization

- Represent a problem as an optimization problem
- Solve an optimization problem by finding **stationary points**
- **Define first-order gradient descent**
- **Define second-order gradient descent**
- Define **step size** and **adaptive step size**
- Explain the role and importance of step sizes in first-order gradient descent
- Apply gradient descent to numerically find local optima

Closed-form solutions

- $c(w) = (w-3)^2$ has a closed-form solution because

$$c'(w) = 2(w-3) = 0 \implies w - 3 = 0 \implies w = 3.$$

- $c(w) = w^2 + \exp(w)$ does not have a closed-form solution because

$$c'(w) = 2w + \exp(w) = 0 \implies \exp(w) = -2w$$

- Can't isolate w on one side, to get an explicit formula (closed-form)
  - Note: this c is not a hard optimization problem, it is convex

Stochastic gradient descent

- If $c(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} c_i(\mathbf{w})$, then we can be more computationally efficient by using a stochastic approximation to the gradient on each step
- Each update consists of taking a mini-batch $\mathscr{B}$ and updating with
- $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \frac{1}{b}\sum_{i \in \mathscr{B}} \nabla c_i(\mathbf{w}_t)$

Stochastic gradient descent

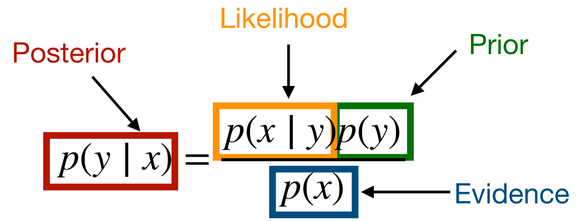
- Each update consists of taking a mini-batch $\mathscr{B}$ and updating with
- $$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \frac{1}{b} \sum_{i \in \mathscr{B}} \nabla c_i(\mathbf{w}_t)$$
- We do this for T iterations (where T is likely more than the number of iterations used for GD)
- Example, if T = 640, n = 4096 and the mini-batch size is b = 32, then we need to do numepochs = 5 to get T = (n/b)*numepochs = 640 updates

You do not need to know

- Specific step-size selection algorithms
  - Adagrad
  - Line search
- stopping criteria, for GD or SGD
  - for GD we usually check if the gradient norm becomes small enough
  - for SGD we just fixed the number of epochs (in practice, you might periodically check if improvement in the objective function has plateaued)

Parameter Estimation

- **Formalize a problem as a parameter estimation problem**
  - e.g., formalize modeling commute times as parameter estimation for a Poisson distribution, using maximum likelihood
- **Describe the differences between MAP, MLE, and Bayesian parameter estimation**
  - MAP $\max_{\theta} p(\theta \mid \mathscr{D})$ versus MLE $\max_{\theta} p(\mathscr{D} \mid \theta)$
  - Bayesian learns $p(\theta \mid \mathscr{D})$, reasons about plausible parameters
- Define a **conjugate prior**



The Likelihood Term and the Prior

- Likelihood:
  $p(\mathscr{D} \mid w) = \Pi_{i=1}^{n} p(x_i \mid w)$
- e.g., Poisson
  $$p(x_i \mid w) = \frac{w^{x_i} \exp(-w)}{x_i!}$$

- Prior:
  $p(w \mid \theta_0)$ for pdf or pmf
  parameters of $p(w)$: θ_0
- e.g., conjugate prior for Poisson is Gamma with parameters $\theta_0 = (a, b)$
  $$p(w \mid \theta_0) = \frac{w^{a-1} \exp(-w/b)}{b^a \Gamma(a)}$$

The Likelihood Term and the Prior

- Likelihood:
  $p(\mathscr{D} \mid w) = \Pi_{i=1}^{n} p(x_i \mid w)$
- e.g., Poisson
  $p(x_i \mid w) = \dfrac{w^{x_i} \exp(-w)}{x_i!}$

- Prior:
  $p(w \mid \theta_0)$ for pdf or pmf
  parameters of $p(w)$: θ_0
- e.g., conjugate prior for Poisson is Gamma with parameters $\theta_0 = (a, b)$
  $p(w \mid \theta_0) = \dfrac{w^{a-1} \exp(-w/b)}{b^a \Gamma(a)}$

- MLE: maximize
  $p(\mathscr{D} \mid w) = \Pi_{i=1}^{n} p(x_i \mid w)$
- MAP: maximize
  $p(\mathscr{D} \mid w) p(w \mid \theta_0) = p(w \mid \theta_0) \Pi_{i=1}^{n} p(x_i \mid w)$

The Likelihood Term and the Prior

- MLE: maximize
  $p(\mathscr{D} \mid w) = \Pi_{i=1}^{n} p(x_i \mid w)$
- MAP: maximize $p(\mathscr{D} \mid w)p(w \mid \theta_0) = p(w \mid \theta_0)\Pi_{i=1}^{n} p(x_i \mid w)$
- Bayesian: obtain posterior $p(w \mid \mathscr{D})$
- e.g., if Poisson likelihood with conjugate prior Gamma with prior parameters $\theta_0 = (a, b)$, then after observing evidence $\mathscr{D}_1 = \{(x_i)\}_{i=1}^{n_1}$ posterior is Gamma with $\theta_1 = (a_1, b_1)$
  where $a_1 = a + \sum_{i=1}^{n_1} x_i$ and $b_1 = \dfrac{1}{n_1 + 1/b}$

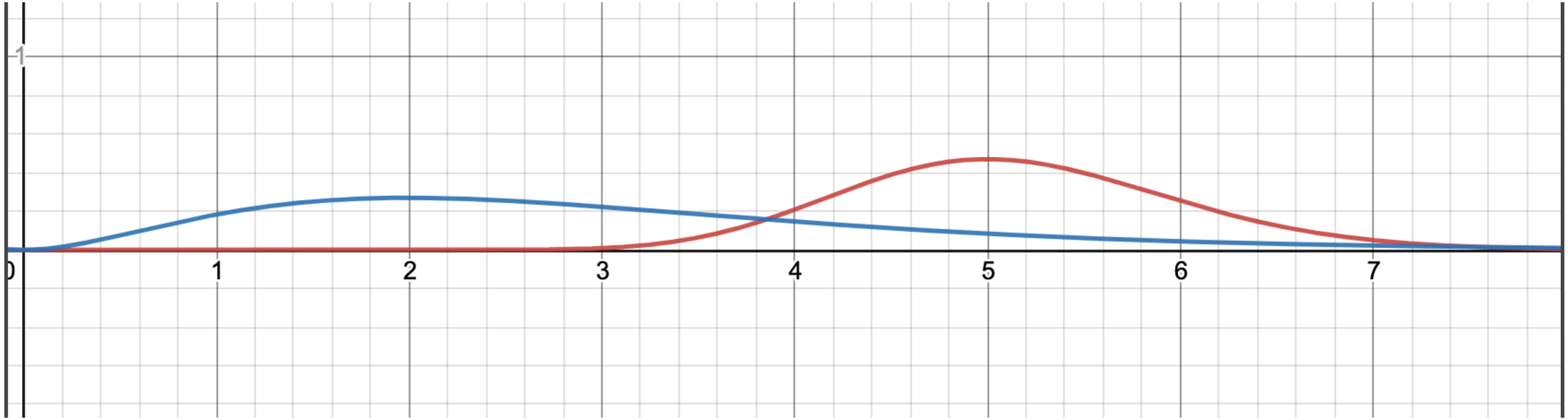
- Prior:
  $p(w \mid \theta_0)$ for pdf or pmf
  parameters of $p(w)$: θ_0
- e.g., conjugate prior for Poisson is Gamma with parameters $\theta_0 = (a, b)$
  $p(w \mid \theta_0) = \dfrac{w^{a-1} \exp(-w/b)}{b^a \Gamma(a)}$

Gamma Prior and Posterior

- For a = 3 and b = 1, we have $p(w) = \frac{1}{2} w^2 \exp(-w)$ because $\Gamma(3) = 2$
- For $\mathscr{D} = \{2,5,9,5,4,8\}$ we have $\sum_{i=1}^{n_1} x_i = 33$
- $a_1 = a + \sum_{i=1}^{n_1} x_i = 36$ and $b_1 = \frac{1}{n_1 + 1/b} = 1/7$
- $p(w \mid \mathscr{D}) = \frac{w^{a_1 - 1} \exp(-w/b_1)}{b_1^{a_1} \Gamma(a_1)} = \frac{w^{35} \exp(-7w)}{7^{-36} \Gamma(36)}$

Gamma Prior and Posterior

- For a = 3 and b = 1, we have $p(w) = \frac{1}{2} w^2 \exp(-w)$ as $\Gamma(k) = (k-1)!$
- $p(w \mid \mathcal{D}) = \frac{w^{a_1 - 1} \exp(-w/b_1)}{b_1^{a_1} \Gamma(a_1)} = \frac{w^{35} \exp(-7w)}{7^{-36} \Gamma(36)}$ (Red)



What is not a conjugate prior?

- Example: X= number of accidents in a day.
- Assume p(x) is Poisson.
- Imagine we pick the prior p(w) to be a Beta distribution or some other distribution than a Gamma distribution.
- Then the posterior may be in a form that we cannot solve - i.e. it doesn’t reduce to the form of a known distribution class.

Poll Question: Why is MAP useful, namely why is it useful to include a prior over the weights? (Select all that apply)

- 1. It incorporates bias to reduce the variance
- 2. The prior makes our solution closer to the true solution
- 3. It lets us reason about uncertainty in our parameters
- 4. It let's us incorporate expert knowledge about plausible weight values

Answer: 1, 4

You do not need to know

- Any specific conjugate priors, or specific formulas for pmfs/pdfs
  - I will tell you if something is a conjugate prior, you just need to know what that means
- I will not get you to do complex derivations, to solve MLE or MAP

Formalizing Prediction

- **Supervised learning problem:** Learn a **predictor** $f : \mathcal{X} \to \mathcal{Y}$ from a dataset $\mathcal{D} = \left\{(\mathbf{x}_i, y_i)\right\}_{i=1}^{n}$
  - $\mathcal{X}$ is the set of **observations**, and $\mathcal{Y}$ is the set of **targets**
- **Classification** problems have discrete, unordered targets
- **Regression** problems have continuous targets
- Once a predictor is learned, its performance is measured by the **expected** $\text{cost}(\hat{y}, y)$ of predicting $\hat{y}$ when the true value is y
- An **optimal predictor** for a given distribution $p(x, y)$ **minimizes** the expected cost

Difference between Classification and Regression

- If I learn a classifier f(x), for classes {0, 1, 2, 3}, what is the range of the predictor f?
- What is the optimal predictor for 0-1 cost for classification?
- Can I use classes like {apples, oranges, pineapples}? How would we write our optimal predictor for this set of classes?
- What is the optimal prediction for squared error costs for regression?

Prediction Concepts

- Describe the differences between **regression** and **classification**
- **Derive the optimal classification predictor for a given cost**
- Derive the **optimal regression predictor** for a given cost
- Understand that the optimal predictor is different depending on the cost
- Describe the difference between **irreducible** and **reducible error**
  - Even an optimal predictor has some **irreducible error**.
    Suboptimal predictors have additional, **reducible error**

$$\mathbb{E}[C] = \mathbb{E}\left[\left(f(X) - f^*(X)\right)^2\right] + \mathbb{E}\left[\left(f^*(X) - Y\right)^2\right]$$

Reducible error Irreducible error

Is Cost the Same as our Objective c?

- We gave this a **different name** to indicate it might not be
- The **Cost** is the penalty we incur for inaccuracy in our predictions
- We parameterize our function or distribution with parameters $\mathbf{w}$
- Our **objective** to find $\mathbf{w}$ has typically been the negative log likelihood
- Example: we might learn $p(y \mid \mathbf{x}, \mathbf{w})$ using $c(\mathbf{w}) = -\ln p(\mathscr{D} \mid \mathbf{w})$
- For the **0-1 cost,** we **evaluate** the predictor $f(\mathbf{x}) = \arg\max_y p(y \mid \mathbf{x}, \mathbf{w})$

Optimal predictors vs MLE/MAP

- Why do we learn $p(y \mid \mathbf{x})$ if we only care about $\mathbb{E}[Y \mid x]$?

- Why do we have to learn a predictor $f(\mathbf{x})$ that returns one prediction $\hat{y}$ instead of just learning $p(y \mid \mathbf{x})$ and returning the whole distribution?

- Is the optimal predictor an MLE or MAP estimator?

Optimal predictors vs MLE/MAP

- Why do we learn $p(y \mid \mathbf{x})$ if we only care about $\mathbb{E}[Y \mid x]$?
  - We still want to recognize that y is stochastic for a given x, so we reason about $p(y \mid \mathbf{x})$ and about modelling it
  - For regression, we don't need $p(y \mid \mathbf{x})$, but we do for other predictors
- Why do we have to learn a predictor $f(\mathbf{x})$ that returns one prediction $\hat{y}$ instead of just learning $p(y \mid \mathbf{x})$ and returning the whole distribution?
  - At some point you have to make a decision: are you going to treat or not?
- Is the optimal predictor an MLE or MAP estimator?
  - The optimal predictor f* has nothing to do with data. We learn f on data (using MAP or MLE) to try to best approximate f*. Chapter 7 is not about learning nor data

Linear Regression

- Represent a problem as a linear regression
- Understand that we assume $p(y \mid \mathbf{x})$ is Gaussian and that the resulting MLE objective corresponds to the sum of squared errors $\sum_{i=1}^{n} (\mathbf{x}_i^\top \mathbf{w} - y_i)^2$
- Understand the computational cost of the **gradient descent** and **stochastic gradient descent** solutions to linear regression
- Represent a **polynomial regression** problem as linear regression
- **Will not be directly tested**
  - Do not need to know the closed-form solution with matrices