Pre-Midterm Review

CMPUT 296: Basics of Machine Learning

Textbook Ch.1 - §9.1

Logistics

"In-class" midterm **Thursday Oct 29** (day after tomorrow!)

- Covers all material through section 9.1
- Midterm will be on eClass during a 24 hour period
- Random spot checks scheduled starting the following week

Recap: Regularization

- **Regularization:** minimize the training cost plus a complexity penalty
  - $c(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \text{cost}(f(\mathbf{x}_i; \mathbf{w}), y_i) + \lambda\, \text{penalty}(\mathbf{w})$
  - Only make a model more complex if it improves loss "enough"
  - The **hyperparameter** λ controls our notion of "enough"
- **L2 Regularization:** penalty is sum of squared weights: $\text{penalty}(\mathbf{w}) = \sum_{j=1}^{d} w_j^2$
  - L2 regularized linear regression corresponds to **MAP inference** with independent zero-mean **Gaussian priors** on each weight (except w_0)
- **L1 Regularization:** Penalty is sum of absolute values: $\text{penalty}(\mathbf{w}) = \sum_{j=1}^{d} |w_j|$
  - Corresponds to MAP inference with independent **Laplacian prior** on weights
  - Produces **sparse** solutions (many entries of $\mathbf{w}$ are set to **exactly 0**)

Lecture Structure

1. Recap & Logistics
2. Midterm structure and details
3. Optimal predictors question walkthrough
4. Learning objectives walkthrough
    - **Clarifying questions** are the point of this class
5. Other questions, clarifications

Midterm Details

- The midterm is **Thursday, October 29** via **eClass**
- There will be a **3 hour** time limit for the midterm
    - Starting at **any time** between 12:01am and 11:59pm Mountain time
    - It should *not* take anywhere near this long (I aimed for it to take **90 minutes**)
- You may use a **single**, **handwritten cheat sheet** if you wish
- You may use a non-programmable calculator if you wish
- Weeks 1 through 8 are included
    - Everything up to and including Regularization

Midterm Structure

- There will be **130 marks** total
- There will be **5-6** multi-part questions
  - **How** you got your answer will be the bulk of the marks
- There will be **no coding** questions
  - But you may be asked to **execute a few steps** of an algorithm
- Every question will be based on the **learning objectives** that we are about to walk through
- **There will be five marks for uploading a picture of your cheat sheet**

Optimal Classifier

		Y	
		0 (No disease)	1 (Has disease)
$\hat{Y}$	0 (don't test)	0	1000
	1 (test)	1	1

Example: Recall that the **optimal binary classifier** for a given $p(y \mid \mathbf{x})$ is:

$$f^*(\mathbf{x}) = \arg\min_{\boxed{\hat{y}}\in\{0,1\}} \boxed{\mathbb{E}}[\text{cost}(\boxed{\hat{y}}, \boxed{Y}) \mid \mathbf{x}]$$

$$= \arg\min_{\hat{y}\in\{0,1\}} \text{cost}(\hat{y}, y=0)p(y=0 \mid \mathbf{x}) + \text{cost}(\hat{y}, y=1)p(y=1 \mid \mathbf{x})$$

Question: What is the optimal classifier for the cost function above?

$$= \arg\min_{\hat{y}\in\{0,1\}} (1-\hat{y})\left[\text{cost}(\hat{y}=0, y=0)p(y=0 \mid \mathbf{x}) + \text{cost}(\hat{y}=0, y=1)p(y=1 \mid \mathbf{x})\right]$$

$$+\hat{y}\left[\text{cost}(\hat{y}=1, y=0)p(y=0 \mid \mathbf{x}) + \text{cost}(\hat{y}=1, y=1)p(y=1 \mid \mathbf{x})\right]$$

$$= \arg\min_{\hat{y}\in\{0,1\}} (1-\hat{y})\left[0p(y=0 \mid \mathbf{x}) + 1000p(y=1 \mid \mathbf{x})\right] + \hat{y}\left[1p(y=0 \mid \mathbf{x}) + 1p(y=1 \mid \mathbf{x})\right]$$

$$= \arg\min_{\boxed{\hat{y}}\in\{0,1\}} (1-\boxed{\hat{y}})1000p(y=1 \mid \mathbf{x}) + \boxed{\hat{y}}$$

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
- Compute joint and conditional probabilities
- Use a provided distribution
  - I will always remind you of the density expression for a given distribution
- Apply **Bayes' Rule** to derive 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**

Estimators (2)

- Apply **concentration inequalities** to derive **error bounds**
- Apply the **weak law of large numbers** to derive error bounds
- Apply concentration inequalities to derive **confidence bounds**
- Define **sample complexity**
- Apply concentration inequalities to derive sample complexity bounds
- Explain when a given concentration inequality can/cannot be used

Optimization

- Represent a problem as an optimization problem
- Solve an analytic 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

Parameter Estimation

- **Describe the differences between MAP, MLE, and Bayesian parameter estimation**
- **Define the posterior, prior, likelihood, and model evidence distributions**
- **Represent a problem as parameter estimation**
- **Represent a problem as a formal prediction problem**
- Define a **conjugate prior**

Prediction

- Represent a problem as a **supervised learning problem**
- 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
- Describe the difference between **discriminative** and **generative** models
- Describe the difference between **irreducible** and **reducible error**
- Describe the assumptions implied by a given error model

Linear Regression

- Represent a problem as **linear regression**
- Derive the **optimal predictor** for a linear model with squared cost and Gaussian errors
- Derive the computational cost of the **analytical** solution to linear regression
- Derive the computational cost of the **gradient descent** and **stochastic gradient descent** solutions to linear regression
- Represent a **polynomial regression** problem as linear regression
- Represent a **nonlinear regression** problem as linear regression

Generalization Error

- Describe the difference between **empirical error** and **generalization error**
- Explain why **training error** is a **biased estimator** of generalization error
- Define **overfitting**
- Describe how to **estimate generalization** error given a dataset
- Describe how to **detect overfitting**
- Apply **k-fold cross-validation** to select hyperparameters and/or features
- Apply **bootstrap resampling** to select hyperparameters and/or features

Generalization Error (2)

- Describe how to compare two models using **confidence intervals**
- Describe how to compare two models using a **hypothesis test**
- Describe how to compare two models using a **paired t-test**
- Define a p**-value**
- Define the **power** of a hypothesis test

Regularization

- Explain how to **avoid overfitting** using cross-validation
- Define a **hyperparameter**
- Define **regularization**
- Define the **L1 regularizer**
- Define the **L2 regularizer**
- Represent L2-regularized linear regression as **MAP inference**
- Explain how to use **regularization** to fit a model
- Describe the effects of the **regularization hyperparameter** λ

Other Questions?