

The second midterm covers in-class material days 9-15 and 19-21, labs 5-8, and reading weeks 5-10 (excluding week 8). It is closed notes/books/internet/other, but you may use a 1 page (front and back), hand-written “resource sheet” (created by *you*) and a calculator. (You shouldn’t need a calculator, but it may make things quicker). I have put vocab in blue.

1. Logistic Regression

- Why don’t we use linear regression for classification problems?
- **Logistic function** of a linear transformation of \mathbf{X} as our model in logistic regression.
- Logistic regression creates a *linear* decision boundary (visualize for $p = 1, 2$).
- Idea of a **likelihood function** and finding the **MLE** (maximum likelihood estimator).
- **Bernoulli random variable** example of MLE calculation.
- In logistic regression our cost is the **negative log likelihood**.
- Intuition behind the cost function.
- Derivation of SGD for logistic regression, relationship to linear regression.
- Idea of multi-class logistic regression (not the mathematical details).

2. Evaluation Metrics

- **Confusion matrices** as evaluation metrics for classification problems
- Binary classification problems with “**positive**” (atypical) and “**negative**” (typical) results
- Confusion matrices for the above specific case: **FP, TP, FN, TN**
- **Precision**: fraction of flagged examples that are truly positive (intuition: “purity”)
- **Recall/TPR**: fraction of true positives that we found (intuition: “completeness”)
- Using the **FPR** and **TPR** at different thresholds to create a **ROC curve**
- What is a “random guessing” ROC curve and what is the ideal ROC curve?
- Where you want to be on the ROC curve depends on the application

3. Ensemble Methods

- Idea of using an **ensemble** of classifiers (ideally with low **bias**) to reduce **variance**
- To test, let each classifier in the ensemble “vote” (could be weighted or unweighted)
- **Bootstrap**: sampling from our data with replacement (usually keeping n the same)
- **Bagging** (Bootstrap Aggregation): create a classifier for each bootstrapped training dataset
- How does averaging the results of many “weak” classifiers reduce the overall error?
- Ensemble notation and example of reducing the error via bagging!
- **Random Forest** classifiers as ensembles of **decision stumps** (or small-depth trees)
- What was the idea behind Random Forests? Why might they be better than regular Bagging?
- **AdaBoost**: upweight training examples that were classified incorrectly in the previous iter
- AdaBoost details: weighted error, score, update example weights, testing with weighted vote

- Decision Trees with weighted examples (how do we modify the probability calculations?)

4. Neural Networks

- What is a **Neural Network** (NN)? Motivation and goals when using them
- High level idea of training using gradient descent on the **loss function**
- **Fully Connected** architectures, dimensionality analysis, parameters vs. hyperparameters
- Choice of **activation function**, pros and cons of **sigmoid**, **tanh**, and **ReLU**
- **Softmax function** as the activation function for the last layer, **cross-entropy** loss after that
- Training: how to initialize the weights/biases, what is the point of **mini-batches**?
- Motivation behind **Convolutional Neural Networks** (CNNs); application to images
- CNN architectures: idea of 3D volumes, typical steps **CONV**, **RELU**, **POOL**, **FC**
- CONV layer details: filters computing **cross-correlations**, slide filter over width and height
- Dimensionality analysis (shapes of filter weights/biases, shapes of input/output)
- Skip: back-propagation, dropout, regularization, any pooling besides 2×2 with stride 2