Overview and Exam Review
Daniel Lowd
Winter 2016

Chapter 1a: Inductive Learning
- Definitions
  - Data distribution, loss function, expected loss, training error
  - Training data, test data, validation data (or development data)
  - Overfitting, underfitting
  - Hyperparameters, tuning
- Inductive bias
- How to define a machine learning problem.

Chapter 1b: Decision trees
- Representation
  - What functions can a decision tree represent?
  - How do we learn a decision tree?
  - How many does a decision tree need to be in order to represent different kinds of functions?
  - How much data do you need to learn different functions as a decision tree?
- Learning
  - Mutual information vs. accuracy (0/1 loss)
  - Lookahead, pruning (basic concepts)
  - What hyperparameters? How to choose them?
  - How do we evaluate decision tree effectiveness?
- NOT NEEDED:
  - Regression, multiclass prediction
  - Gini score
  - Convexity of mutual information

Chapter 2: k-Nearest Neighbor
- Representation
  - What can k-nearest neighbor represent?
  - Effect of changing k
- How are predictions computed? Efficiency?
- Curse of dimensionality – how distances change in high dimensions (qualitative)
- NOT NEEDED: k-means, kd-trees, weighted k-means

Chapter 3: Perceptron
- Representation
  - What functions can a linear model represent?
- Linear separability
- Learning
  - Perceptron update
  - Convergence
  - Effect of example order
- NOT NEEDED: Averaged perceptron, analytic geometry, convergence rate or proof

Chapter 6: Linear models
- Regularized learning objective
- Convex loss functions
- Weight regularization
- Properties of regularizers (L1, L2)
- Gradient descent
- Hyperparameters (learning rate, choice of regularizer, weight of regularizer) and their effect
- Soft-margin SVM
- NOT NEEDED: Convergence rate of gradient descent, closed form optimization of squared loss
Chapter 7: Probabilistic models

- Naive Bayes
  - Naive Bayes parameter estimation with Beta prior
  - Making predictions with Naive Bayes
- Logistic regression
  - Conditional likelihood
  - Logistic function, logistic loss
  - Prior distribution: Gaussian / L2
- Naive Bayes vs. logistic regression: flexibility, accuracy, convergence rate
- NOT NEEDED: Lagrange multipliers, Gaussian naive Bayes, generative stories, linear regression, Laplace prior, how to maximize likelihood

Chapter 8: Neural networks

- Representation – what can you represent with different architectures? How?
- Nonlinearities - threshold units, tanh units
- Optimization – gradient descent, back-propagation (basic concepts)
- NOT NEEDED: Deep learning, detailed understanding of back-propagation update equations, RBF networks, theory about # of units required to approximate functions

Chapter 9: Kernel methods

- How to make a prediction using a kernel
- What can a kernelized SVM (or perceptron) represent?
- Polynomial kernel
- RBF kernel
- NOT NEEDED: Derive the dual formulation of the support vector machine, invent a new kernel, kernelized perceptron updates

Exam

- Closed book, 1 page of notes
- 1 hour and 20 minutes to complete
- Conceptual questions, minor calculation
- Covers everything we’ve discussed in class so far

Simulate Classifier Behavior

Given a dataset:
- Draw the decision tree that would be generated from it, using information gain to select the splits.
- Specify the naive Bayes parameters (the conditional probabilities, not the log-odds-style parameters from the homework), assuming a prior of Beta(3,3) on each parameter.
- Show the results of a perceptron update for the first two instances, assuming initial weights of 0; all variables are represented as 0/1 (not -1/+1).
- Draw the decision boundary for an SVM.

What would the error rate of each be on the training data?

Compare and Contrast Methods

For each problem, which of the classifiers that we have discussed (decision tree, nearest neighbor, naive Bayes, logistic regression, perceptron, neural net, SVM) would probably be the best and why?

Problem 1: Recognizing handwritten letters (A-Z) from 32x32 pixel images, given 1 million training examples.

Problem 2: Predicting heart attack probability based on 50 risk factors, each of which has an independent effect on the output. 1000 training examples.

Problem 3: Predicting grade (A,B,C) on an essay test using a bag-of-words model. Only 50 training examples available.
Other Questions

• How many nodes are requested to represent parity with a decision tree? Or as a neural net with one hidden layer?
• Draw an example of a dataset in 2D that will be learned correctly by k-NN when k=3 but not when k=1.
• Given a 2D topology landscape, draw the direction of the gradient.
• Suppose you have m training examples. What is the largest number of hidden units required to build a neural net that labels every training example correctly? (Do not worry about learning, just about representation.)
• Describe how to properly compare the accuracy of two machine learning methods, each with one parameter to tune, on a dataset of 1000 examples.
• How do you identify overfitting? How do you reduce overfitting in specific models?