Exam Review

Introduction to Machine Learning
T-529-ITME

Instructor: Dan Lizotte

Exam Logistics

- When: Tuesday, 15 May 2007 at 9:00am
- Where: Ofanleiti 131a, 131b
- Materials/aids: None. No books, no calculators, no laptops.
- You don't need to memorize formulas **except as noted in this document**, but you should know what they mean.

Introduction

- What is classification?
- What is regression?
  - What is the difference?
- What do these have in common with Reinforcement learning?
  - They are all prediction problems.
- What is different?
  - RL is Evaluative learning
  - Classification and Regression are Instructive learning
    - "Supervised" Learning
- What is a "feature"?

Decision Trees

- Understand the meaning of Entropy
  - More entropy -> more uncertainty
- Understand the meaning of Information Gain
  - IG = Entropy Before - Entropy After
- Know how a tree is constructed
  - Choose a feature, split, choose a new feature, split…
  - When do we stop?
- Know how to use a tree to classify an instance
- Why is pruning important?

Decision Trees, General Classifier Stuff

- Understand the difference between “Training Error” and “Test Error”
- Why do we care about the difference?
  - Want to avoid overfitting.
  - Test set error is more representative of future error
- How can we avoid overfitting?
  - Pruning
  - chi-squared test estimates “what is the probability we would see these data by accident?”
  - And therefore “Should we maybe just ignore this split?”

PAC Learning

- PAC Stands for…?
- Know what a hypothesis space is
  - The space of all functions representable by your learning machine.
- How to count a simple hypothesis space
  - Figure out what the independent choices are
    - e.g. “To include x_i or not to include x_i.”
  - Multiply the number of independent choices together

PAC Learning

- Understand that if we have a hypothesis space of size H, and we want to have test error < ε with probability $(1 - \delta)$ then we need R data points to guarantee this, where

$$R \geq \frac{1}{\varepsilon}\left(\log_2 H + \log_2 \frac{1}{\delta}\right)$$

- BIG IDEA: *Bigger hypothesis space needs more data.*

VC Dimension

- When do we use VC dimension?
  - When $H = \infty$, but we need to measure complexity.
- Understand Shattering
  - Show how to shatter a given set of points with a given (*simple*) classifier
- VC dimension = k if
  - Can shatter *some* set of k points. (You pick.)
  - Can**not** shatter *any* set of k+1 points.

VC Dimension

- Understand that if we have a particular TRAINERR achieved on *R* data points, and the VC dimension of our classifier is *h,* then we know the following is true with probability (1 - *η*):

$$\text{TESTERR} \le \text{TRAINERR} + \sqrt{\frac{h(\log(2R/h)+1) - \log(\eta/4)}{R}}$$

- *Structural Risk Minimization* is picking the classifier with the smallest bound

VC Dimension

- Again, notice that the more complex a classifier we have, the more data we need to guarantee good performance.

Cross-Validation

- We want good performance on *test* data. Cross validation is a good way to estimate this performance.
  - Training error is too optimistic.
- Understand
  - What a test set is
  - LOOCV - Leave One Out Cross Validation
  - k-Fold Cross Validation
- Be able to explain how each of these works
- Remember the folk-theorem:
  - You need about 10 times as much data as you have parameters in your model

Density Estimators, Bayes Classifiers

- Be able to compute simple probabilities

- KNOW
  - 0 <= P(A) <= 1
  - P(A or B) = P(A) + P(B) - P(A and B)
  - P(A|B) = P(A and B) / P(B)
  - Bayes Rule: P(A|B) = P(B|A)*P(A) / P(B)

Density Estimators, Bayes Classifiers

- Be able to produce, given a small amount of data
  - A Joint Density Estimator *or* Bayes Classifier
  - A Naïve Density Estimator *or* Bayes Classifier
- Be able to compute P(class = +) given
  - Joint Density estimates
  - Naïve Density estimates

- KNOW
  - For naïve: P(A and B | C) = P(A|C) * P(B|C)
  - For joint: P(A and B | C) = look it up in your table

Density Estimators, Bayes Classifiers

- Know that, for *m* binary variables
  - Joint Density learns 2^m numbers
    - Therefore needs lots of data
  - Naïve Density learns *m* numbers
    - Therefore needs little data

- But the Naïve Density Estimator is not very powerful
  - Assumes independence
  - Cannot capture relationships between variables

Support Vector Machines

- Know what a linear separator is.
- Given a weight vector *w* and constant *b*, and a data point *x*, KNOW how to classify that point.
  - class = sign($w \cdot x + b$)
- If I gave you a picture of some data points, draw the maximum margin separator, along with + and - planes, and indicate the margin.
- Know what a support vector is.

Support Vector Machines

- Know what a slack variable is for
  - allows training points to be misclassified
- Know why sometimes we use kernels
  - when training data are not linearly separable
- Understand why using a kernel is like inventing new features
  - a.k.a. 'basis functions'

Reinforcement Learning

- Understand the Big Four:
  - Policy
  - Reward
  - Value
  - Transition Model
- Understand what TD learning is trying to do
  - Learn a good value function in order to learn a good policy
- Know the difference between Sarsa and Q-learning
  - Understand on-policy vs. off-policy learning