Introduction, Course Overview

CSci 5525: Machine Learning

Instructor: Paul Schrater

September 5, 2017

Course Activities

- **Please** read the syllabus carefully
- Individual activities
  - Homeworks: 1+4
  - Midterm: Closed book, four sheets of notes allowed
  - Finals: Take home
- Group activities
  - Project: Proposal, progress report, presentation, final report

Individual Activity: Homeworks

- There will be 4 homeworks
  - HW0 is on background/preparation
  - We will also test the submission system (moodle)

- All submissions in pdf
- All programming in Python or Matlab

- Dates/times (central):
  - HW 0: Sept 06 (Tue), due Sept 12 (Mon) at 11:55 pm (6 days)
  - HW 1: Sept 20 (Tue), due Sept 30 (Fri) at 11:55 pm (10 days)
  - HW 2: Oct 04 (Tue), due Oct 14 (Fri) at 11:55 pm (10 days)
  - HW 3: Nov 01 (Tue), due Nov 11 (Fri) at 11:55 pm (10 days)
  - HW 4: Nov 22 (Tue), due Dec 02 (Fri) at 11:55 pm (10 days)

Individual Activity: Homeworks (contd)

- Late submission policy:
  - You have a total of 4 grace days
  - You can choose to use them as convenient to delay one/more homework submissions
  - Grace days cannot be used to delay project components or the finals
  - Delays beyond the grace days:
    - Late by 0-24 hrs: 50% of actual score
    - Late by 24-48 hrs: 25% of actual score
    - Late by more than 48 hrs: Will receive a zero

Individual Activity: Midterm

- Tue, Oct 25, in class
- Closed book exam
- Allowed 4 sheets of notes

Individual Activity: Final

- Due Sat, Dec 17, 11:55 pm, in moodle
- Exam will be posted 48 hours before the due time
- Primary focus on material covered after the midterm
- A few selected topics from before the midterm

Group Activity: Project

- Groups of 3 students, form groups by Sept 28
- Project components
  - Proposal: 1-page + refs, due Thu, Oct 19, 11:55 pm
  - Progress Report: 2-page + refs, due Thu, Nov 23, 11:55 pm
  - Final Report: 10-12 -page + refs, appendix, etc., due Sun, Dec
  - 19, 11:55 pm
- Helpful resources
  - Project ideas, e.g., http://www.kaggle.com/competitions
  - ML packages, e.g., http://scikit-learn.org/stable/

Grading

- Individual Activity:
  - Homeworks: 50 % $= 4 \times 12.5$ %
  - Mid-Term: 20 %
  - Final Project: 30 %
- Group Activity:
  - Project: 30 %

- Grading is absolute: A $=$ 90-100, A- $=$ 85-90, B$+$ $=$ 80-85, B $=$ 70-80, B- $=$ 65-70, C$+$ $=$ 60-65, C $=$ 50-60, F $=$ less than 50.

Topics

- Linear regression, linear discriminants
- Models: Generative (naive Bayes), Discriminative (logistic regression)
- Support Vector Machines, Constrained Optimization, Duality
- Optimization: (Stochastic) Gradient Descent
- Nonlinear methods: Kernels
- Classification and Regression Trees
- Ensembles: Boosting, Bagging, Random Forests
- Nearest Neighbor methods
- Deep Learning
- Learning Theory, Online Learning, Online Optimization
- Clustering: Kmeans, EM, Spectral
- Dimensionality Reduction: Linear, Nonlinear
- Gaussian Processes

Overview

- Applications
  - Type of data: vectors, time-series, sequences, spatiotemporal, etc.
  - Domain: text, image, speech, videos, social networks, finance, biology, climate, healthcare, etc.
  - Type of problem: regression, classification, anomaly detection, ranking, etc.
- Models and Methods
  - Model: assumptions, parameters
  - Learning algorithms: training models based on data
  - Representation: native features vs. learning representations
- Theory
  - Generalization in batch learning
  - Regret in online learning

Overview (Contd)

- Key Concepts:
  - Representation
  - Model Selection
  - Over-fitting, Regularization
- Trade-offs:
  - Generative vs Discriminative
  - Max Likelihood vs Max Margin
- Algorithms:
  - Representations: Hierarchical, Deep, Nonlinear, Sparse/Structured
  - Linear Models, Layered Linear Models
  - Optimization: Stochastic, Parallel, Streaming
  - Ensemble Models: Bagging, Boosting, Random Forests
  - Exploratory Analysis: Clustering, Dimensionality Reduction
- Theory:
  - Basics, Risk Minimization
  - Generalization Bounds, Regret Bounds

Key Concepts

- Representation
  - Feature selection, extraction
  - Pairwise non-linear similarity, kernels
  - Learning representations
- Model selection
  - "Bias" $\equiv$ manual model selection
  - "Learning" $\equiv$ algorithmic model selection
- Regularization
  - Guides model selection
  - Trade-off prior belief with learning from observations
  - Similar to Bayesian priors and Bayesian conditionals
  - Being conservative is a good idea
- Overfitting
  - Predict well on training set, poorly on test set/future data
  - Result of greedy/non-conservative learning
  - To be avoided using regularization, large training sets, etc.

Classification

- Assume: A fixed (unknown) distribution on $\mathbb{R}^d \times \{-1, +1\}$
- Given: A set $T = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ of n samples from the distribution
- Problem: Find a function $f : \mathbb{R}^d \mapsto \{-1, +1\}$ that has "low" error rate, i.e., $L(f) = P(f(\mathbf{x}) \neq y)$ is low
- Let $\mathcal{C}$ be the set of functions over which f is searched for
    - "Bias" determines the set $\mathcal{C}$
    - A learning algorithm is the search algorithm in $\mathcal{C}$
- For Multiclass problems, $(\mathbf{x}, y) \in \mathbb{R}^d \times \{1, \ldots, c\}$
- For Regression problems, $(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R}$
- For Multi-dimensional Regression problems, $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^d \times \mathbb{R}^k$

Generative Vs Discriminative

- Generative:
  - Assume a (parametric) model for $p(\mathbf{x}|y)$
  - Training $\equiv$ Estimating parameters of the model
  - Prediction using Bayes rule

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$$

  - Example: Linear Discriminant Analysis, Naive Bayes
- Discriminative:
  - Do not assume a model for $p(\mathbf{x}|y)$, and hence $p(\mathbf{x})$
    - Assume a model for $p(y|\mathbf{x})$
    - Direct formulation in terms of loss
  - Example: Logistic Regression

Max-Likelihood Vs Max-Margin

- Max-Likelihood:
  - Improve average performance
  - Consistent for parameter estimation purposes
  - Focus is on the typical
- Max-Margin:
  - Improve worst case performance
  - Consistent for classification purposes
  - Focus is on the boundary

Linear Models

- Basic Linear Models
  - Naive Bayes, Logistic Regression
  - Perceptrons, Support Vector Machines
- Layered Linear and Hierarchical Models: Representations
  - Decision and Regression Trees
  - Deep Learning
- Kernel Methods
  - Nonlinear, linear in a mapped space
  - Gaussian Processes = Bayesian + Kernel

Ensemble Models

- Global Ensembles
  - Experts, Bayesian models
  - Boosting, Bagging, Random Forests
- Local Ensembles
  - Nearest Neighbors

Clustering

- Hard clustering, centroid based
  - Kmeans, Bregman clustering
  - K-median, facility location
- Soft clustering, mixture models
  - Mixture of Gaussians
  - Bayesian mixture models
- Spectral clustering, graph cuts
  - Normalized cut, ratio cut
  - Graph Laplacian

Dimensionality Reduction

- Principal Component Analysis (PCA)
  - Probabilistic PCA
- Nonlinear manifold embedding
  - Isomap
  - Locally linear embedding
  - Laplacian eigenmaps

What we will not cover

- Bleeding edge of deep learning
- Semi-supervised learning, cost sensitive learning
- Structured prediction, ranking, preference learning
- Graphical models, nonparametric Bayes, latent variable models
- Transfer and multi-task learning
- Active learning, noisy training, noisy auto-encoders
- Kernel learning
- Policy learning, deep reinforcement learning (see AI II)
- Applications: Vision, Speech, NLP, IR, Bioinformatics, etc.
- Matrix factorization and recommendation systems
- ... and many other topics

Theory

- Basics, Models of Learning
- Empirical and Structured Risk Minimization
- Bounds based on complexity/capacity of function classes
- PAC Bayesian Bounds
- Regret Bounds

Loss

- Learning is often based on *minimizing expected loss*
- 0/1 Loss: $L(f, \mathbf{x}, y) = \mathbb{1}_{[f(\mathbf{x}) \neq y]}$, expected loss

$$L(f) = E[\mathbb{1}_{[f(x) \neq y]}] = P(f(\mathbf{x}) \neq y)$$

- Hinge Loss:

$$L(f, \mathbf{x}, y) = \max(0, 1 - yf(x)) = \begin{cases} 1 - yf(\mathbf{x}) & \text{if } yf(\mathbf{x}) < 1, \\ 0 & \text{otherwise.} \end{cases}$$

- Exponential Loss:

$$L(f, \mathbf{x}, y) = \exp(-yf(\mathbf{x}))$$

- Logistic Loss:

$$L(f, \mathbf{x}, y) = \log(1 + \exp(-yf(\mathbf{x})))$$

The Bayes Classifier

- Let $P(y|\mathbf{x})$ be the true conditional distribution
- The Bayes Classifier is given by

$$f^*(\mathbf{x}) = \begin{cases} 1 & \text{if } P(1|\mathbf{x}) > 1/2, \\ 0, & \text{otherwise }. \end{cases}$$

- For any classifier f, $L(f^*) \leq L(f)$
- The Bayes Classifier is the "optimal" classifier

"Bias" Revisited

- In practice, one chooses f_n^* from $\mathcal{C}$ given n training samples
- Clearly, $L(f_n^*) > L(f^*)$
- An important decomposition

$$L(f_n^*) - L(f^*) = \left(L(f_n^*) - \inf_{f \in \mathcal{C}} L(f) \right) + \left(\inf_{f \in \mathcal{C}} L(f) - L(f^*) \right) .$$

- First term is the *estimation error* (ee)
- Second term is the *approximation error* (ae)
- Choice of "bias" trades-off the two terms:
  - High "bias" $\Rightarrow$ low ee, high ae
  - Low "bias" $\Rightarrow$ high ee, low ae