Introduction, Course Overview

CSci 5525: Machine Learning

Instructor: Nicholas Johnson

September 8, 2020

General Information

- Course Number: CSci 5525
- Class: Tue, Thu 11:15 AM - 12:30 PM
- Location: Online asynchronous
- Instructor: Nicholas Johnson
- TA: Tiancong Chen, Logan Stapleton
- Office Hours:
  - Nick: Zoom Mon/Wed 5:00 - 6:00 PM
  - Tiancong: Zoom TBD
  - Logan: Zoom TBD
- Canvas page: https://canvas.umn.edu/courses/194060
- Email:
  - Nick: njohnson@cs.umn.edu
  - Tiancong: chen6271@umn.edu
  - Logan: stapl158@umn.edu

Course Activities

- **Please** read the syllabus carefully
- Individual activities
  - Homeworks: 1+4
  - Exams: 2 take-home exams
- Group activities
  - Project: proposal, progress report, final report

Individual Activity: Homeworks

- There will be 4 homeworks
  - HW0 is on background/preparation; must be completed/submitted to remain enrolled
  - All homeworks due at 11:59 PM central time on due date
  - No late homeworks will be accepted
  - All submissions in PDF format
  - All programming in Python 3.6
- Dates/times:
  - HW 0: posted Sept 08 (Tue), due Sept 15 (Tue)
  - HW 1: posted Sept 22 (Tue), due Oct 1 (Thu)
  - HW 2: posted Oct 01 (Thu), due Oct 15 (Thu)
  - HW 3: posted Oct 22 (Thu), due Nov 5 (Thu)
  - HW 4: posted Nov 19 (Thu), due Dec 08 (Tue)

Individual Activity: Exams

- Take-home Exam 1: posted Oct 20 (Tue), due Oct 22 (Thu)
- Take-home Exam 2: posted Nov 10 (Tue), due Nov 12 (Thu)

Group Activity: Project

- Groups of at most 3 students
- Project components (each due at 11:59 PM central on due date)
  - Proposal: 1 page, due Sept 24 (Thu)
  - Progress Report: 2 pages, due Nov 17 (Tue)
  - Final Report: 5 pages + refs, due Dec 18 (Fri)
- Helpful resources
  - Project ideas, e.g., http://www.kaggle.com/competitions
  - ML packages, e.g., http://scikit-learn.org/stable/, https://www.tensorflow.org

Grading

- Individual Activity:
  - Homeworks: 40 % $= 4 \times 10$ %
  - Exams: 30 % $= 2 \times 15$ %
- Group Activity:
  - Project: 30 % $= 5 + 10 + 15$
    (Proposal + Progress Report + Final Report)

- Grading is absolute: A = 90-100%, A- = 88-90%, B+ = 86-88%, B = 76-86%, B- = 74-76%, C+ = 72-74%, C = 62-72%, C- = 60-62%, D+ = 58-60%, D = 50-58%, F = less than 50%

Topics

- Linear regression, linear discriminants

Topics

- Linear regression, linear discriminants
- Models: Generative (naive Bayes), Discriminative (logistic regression)

Topics

- Linear regression, linear discriminants
- Models: Generative (naive Bayes), Discriminative (logistic regression)
- Support Vector Machines

Topics

- Linear regression, linear discriminants
- Models: Generative (naive Bayes), Discriminative (logistic regression)
- Support Vector Machines
- Optimization: (Stochastic) Gradient Descent

Topics

- Linear regression, linear discriminants
- Models: Generative (naive Bayes), Discriminative (logistic regression)
- Support Vector Machines
- Optimization: (Stochastic) Gradient Descent
- Nonlinear methods: Kernels

Topics

- Linear regression, linear discriminants
- Models: Generative (naive Bayes), Discriminative (logistic regression)
- Support Vector Machines
- Optimization: (Stochastic) Gradient Descent
- Nonlinear methods: Kernels
- Nearest Neighbor methods

Topics

- Linear regression, linear discriminants
- Models: Generative (naive Bayes), Discriminative (logistic regression)
- Support Vector Machines
- Optimization: (Stochastic) Gradient Descent
- Nonlinear methods: Kernels
- Nearest Neighbor methods
- Deep Learning

Topics

- Linear regression, linear discriminants
- Models: Generative (naive Bayes), Discriminative (logistic regression)
- Support Vector Machines
- Optimization: (Stochastic) Gradient Descent
- Nonlinear methods: Kernels
- Nearest Neighbor methods
- Deep Learning
- Learning Theory

Topics

- Linear regression, linear discriminants
- Models: Generative (naive Bayes), Discriminative (logistic regression)
- Support Vector Machines
- Optimization: (Stochastic) Gradient Descent
- Nonlinear methods: Kernels
- Nearest Neighbor methods
- Deep Learning
- Learning Theory
- Classification and Regression Trees

Topics

- Linear regression, linear discriminants
- Models: Generative (naive Bayes), Discriminative (logistic regression)
- Support Vector Machines
- Optimization: (Stochastic) Gradient Descent
- Nonlinear methods: Kernels
- Nearest Neighbor methods
- Deep Learning
- Learning Theory
- Classification and Regression Trees
- Ensembles: Boosting, Bagging, Random Forests

Topics

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

Topics

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

Topics

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

Topics

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

Topics

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

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 (cont.)

- Key Concepts:

Overview (cont.)

- Key Concepts:
  - Representation

Overview (cont.)

- Key Concepts:
  - Representation
  - Model Selection

Overview (cont.)

- Key Concepts:
  - Representation
  - Model Selection
  - Over-fitting, Regularization

Overview (cont.)

- Key Concepts:
  - Representation
  - Model Selection
  - Over-fitting, Regularization
- Trade-offs:

Overview (cont.)

- Key Concepts:
  - Representation
  - Model Selection
  - Over-fitting, Regularization
- Trade-offs:
  - Generative vs Discriminative

Overview (cont.)

- Key Concepts:
  - Representation
  - Model Selection
  - Over-fitting, Regularization
- Trade-offs:
  - Generative vs Discriminative
  - Max Likelihood vs Max Margin

Overview (cont.)

- Key Concepts:
  - Representation
  - Model Selection
  - Over-fitting, Regularization
- Trade-offs:
  - Generative vs Discriminative
  - Max Likelihood vs Max Margin
- Course Sections (i.e., Learning Paradigms):

Overview (cont.)

- Key Concepts:
  - Representation
  - Model Selection
  - Over-fitting, Regularization
- Trade-offs:
  - Generative vs Discriminative
  - Max Likelihood vs Max Margin
- Course Sections (i.e., Learning Paradigms):
  - Supervised: learning from a labeled dataset

Overview (cont.)

- Key Concepts:
  - Representation
  - Model Selection
  - Over-fitting, Regularization
- Trade-offs:
  - Generative vs Discriminative
  - Max Likelihood vs Max Margin
- Course Sections (i.e., Learning Paradigms):
  - Supervised: learning from a labeled dataset
  - Learning Theory: guarantees and analysis of learning algorithm performance/behavior

Overview (cont.)

- Key Concepts:
  - Representation
  - Model Selection
  - Over-fitting, Regularization
- Trade-offs:
  - Generative vs Discriminative
  - Max Likelihood vs Max Margin
- Course Sections (i.e., Learning Paradigms):
  - Supervised: learning from a labeled dataset
  - Learning Theory: guarantees and analysis of learning algorithm performance/behavior
  - Ensemble models: ways of combining data/algorithms

Overview (cont.)

- Key Concepts:
  - Representation
  - Model Selection
  - Over-fitting, Regularization
- Trade-offs:
  - Generative vs Discriminative
  - Max Likelihood vs Max Margin
- Course Sections (i.e., Learning Paradigms):
  - Supervised: learning from a labeled dataset
  - Learning Theory: guarantees and analysis of learning algorithm performance/behavior
  - Ensemble models: ways of combining data/algorithms
  - Unsupervised: learning structure of data (no labels)

Overview (cont.)

- Key Concepts:
  - Representation
  - Model Selection
  - Over-fitting, Regularization
- Trade-offs:
  - Generative vs Discriminative
  - Max Likelihood vs Max Margin
- Course Sections (i.e., Learning Paradigms):
  - Supervised: learning from a labeled dataset
  - Learning Theory: guarantees and analysis of learning algorithm performance/behavior
  - Ensemble models: ways of combining data/algorithms
  - Unsupervised: learning structure of data (no labels)
  - Reinforcement: learning from interacting with environment (i.e., trial and error)

Key Concepts

- Representation

Key Concepts

- Representation
  - Feature selection, extraction

Key Concepts

- Representation
  - Feature selection, extraction
  - Pairwise non-linear similarity, kernels

Key Concepts

- Representation
  - Feature selection, extraction
  - Pairwise non-linear similarity, kernels
  - Learning representations

Key Concepts

- Representation
  - Feature selection, extraction
  - Pairwise non-linear similarity, kernels
  - Learning representations
- Model selection

Key Concepts

- Representation
  - Feature selection, extraction
  - Pairwise non-linear similarity, kernels
  - Learning representations
- Model selection
  - "Bias" $\equiv$ manual model selection

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

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

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

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

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

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

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

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

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

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 $\mathcal{X} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ 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, $(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R}$

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

Supervised Learning

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

Learning Theory

- Batch Learning
  - Empirical and Structured Risk Minimization
  - Generalization, PAC learning
- Online Learning
  - Regret bounds (stochastic, adversarial)
- Complexity measures
  - VC dimension, Rademacher complexity

Ensemble Models

- Hierarchical Models
  - Decision and Regression Trees
- Global Ensembles
  - Boosting, Bagging, Random Forests

Unsupervised Learning

- Dimensionality Reduction
  - Principal Component Analysis (PCA)
- Clustering
  - Kmeans, Mixture of Gaussians, Expectation Maximization
  - Spectral clustering
- Generative Models
  - Autoencoders
  - Generative Adversarial Networks (GANs)

Reinforcement Learning

- Online
  - Online learning, Online convex optimization
- Sequential Decision Making
  - Q-learning, Deep Q-learning

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
- Kernel learning
- Applications: Vision, Speech, NLP, IR, Bioinformatics, etc.
- Matrix factorization and recommendation systems
- ... and many other topics