Statistical Machine Learning (CSE 575)

About this Course
The link between inference and computation is central to statistical machine learning, which combines the computational sciences with statistics. In addition to artificial intelligence, fields such as information management, finance, bioinformatics, and communications are significantly influenced by developments in statistical machine learning. This course investigates the data mining and statistical pattern recognition that support artificial intelligence. Main topics covered include supervised learning; unsupervised learning; and deep learning, including major components of machine learning and the data analytics that enable it.

Specific topics covered include:

- Probability distributions
- Maximum likelihood estimation
- Naive Bayes
- Logistic regression
- Support vector machines
- Clustering
- Principal component analysis
- Neural networks
- Convolutional neural networks

Learning Outcomes
Learners completing this course will be able to:

- Distinguish between supervised learning and unsupervised learning
- Apply common probability distributions in machine learning applications
- Use cross validation to select parameters
- Use maximum likelihood estimate (MLE) for parameter estimation
- Implement fundamental learning algorithms such as logistic regression and k-means clustering
- Implement more advanced learning algorithms such as support vector machines and convolutional neural networks
- Design a deep network using an exemplar application to solve a specific problem
- Apply key techniques employed in building deep learning architectures
Course Content

**Instruction**
- Video lectures
- Other videos (animations, demos, etc.)
- Readings
- Live sessions (office hours, webinars, etc.)

**Assessments**
- Practice activities and quizzes (auto-graded)
- Practice assignments (instructor- or peer-reviewed)
- Team and/or individual project(s) (instructor-graded)
- Midterm or final exam (proctored, auto- and/or instructor-graded)

Estimated Workload/ Time Commitment Per Week
Average of 20 hours per week

Required Prior Knowledge and Skills
- Basics of linear algebra, statistics, calculus, and algorithm design and analysis
- Programming (language such as Python or MATLAB)

Technology Requirements

**Hardware**
Standard with major OS

**Software and Other**
Standard - technology integrations will be provided through Coursera

Course Outline

Unit 1: Introduction to Machine Learning

**Learning Objectives**
1.1 Describe common misconceptions of machine learning
1.2 Define machine learning
1.3 Distinguish between supervised learning and unsupervised learning
1.4 Compare numerical and graphical data representations
1.5 Describe applications of machine learning

**Module 1: Defining Machine Learning**

Statistical Machine Learning
Updated April 2018
Common misconceptions
What is Machine Learning?
Related fields

**Module 2: Styles of Machine Learning**
Supervised learning
Unsupervised learning

**Module 3: Data Representations**
Data representation
Numerical representation
Graph representation

**Module 4: Applications of Machine Learning**
Recognizing examples
Familiar applications
Emerging applications

**Unit 2: Statistical Core of Machine Learning**

**Learning Objectives**

2.1 Apply common probability distributions in machine learning applications
2.2 Use maximum likelihood estimate (MLE) for parameter estimation

**Module 1: Probability**
Discrete Random Variables
Probability Mass Function (PMF)
Common Distributions of PMF
  - Uniform
  - Binomial
Joint Probability Mass Function
Conditional Probability
Relationship Between Marginal and Joint Probability
Bayes Theorem
Independent Random Variables
Continuous Random Variables
Probability Density Function (PDF)
Common Distributions of PDF
  - Normal
Module 2: Maximum Likelihood Estimation
Likelihood function
  For discrete probability distribution
  For continuous probability distribution
Maximum likelihood estimation
  For discrete probability distribution
  For continuous probability distribution
  For mean and standard deviation

Unit 3: Supervised Learning: Two Models

Learning Objectives
3.1 Differentiate between generative and discriminative models for supervised learning
3.2 Implement fundamental learning algorithms such as Naive Bayes and Logistic Regression
3.3 Interpret empirical comparisons of Naive Bayes and Logistic Regression

Module 1: Generative vs Discriminative Model of Supervised Learning
Generative vs Discriminative models for supervised learning
  Essential distinction
  Generative model: Naive Bayes
  Discriminative model: Logistic Regression

Module 2: Naive Bayes
  Naive Bayes Assumption
  Decision Rule
Parameters of Naive Bayes
Maximum Likelihood Estimation (MLE) for Naive Bayes Parameters
Text Classification using Naive Bayes
  Bag of Words Model for Text

Module 3: Logistic Regression
Logistic Function
Linear Classifier
Module 4: Comparing the Models
Empirical Comparison of Naive Bayes and Logistic Regression

Unit 4: Supervised Learning: Support Vector Machines

Learning Objectives
4.1 Differentiate between linearly separable and non-separable support vector machines
4.2 Explain the role of the kernel trick in support vector machines
4.3 Explain options for picking magic parameters in support vector machines
4.4 Implement the more advanced learning algorithm known as support vector machines

Module 1: Introduction to Support Vector Machines
SVM: Separable vs non-separable

Module 2: Separable
Linearly Separable Example
Max-margin Separating Hyperplane
Margin Maximization with Canonical Hyperplanes
Optimization Problem of SVM: separable case
Dual SVM Formulation: separable case

Module 3: Non-separable
Linearly Non-separable Example
Hinge Loss
Optimization Problem of SVM: non-separable case
Dual SVM Formulation: non-separable case
Input Space to Feature Space
Kernel Trick
   Common Kernels
   Test Example
   SVM with the Kernel Trick

Module 4: Parameter Selection
Unit 5: Unsupervised Learning: Clustering

**Learning Objectives**

5.1 Differentiate between clustering in supervised vs. unsupervised learning  
5.2 Explain how to efficiently cluster data  
5.3 Apply the k-means algorithm  
5.4 Explain the relationship between the several K-means variants

**Module 1: Introduction to Clustering**

The role of clustering in machine learning  
Clustering in supervised versus unsupervised learning  
How to find good clustering  
  - Intuition  
  - An example  
  - Mathematical formulation  
How to efficiently cluster data  
  - Challenge - combinatorial nature  
  - Solution:  
    - High-level Idea: alternation  
    - Details - step 1: fix the cluster clusters, find the cluster membership  
    - Details - step 2: fix the cluster membership, update the cluster center

**Module 2: K-means**

K-means for clustering  
K-means models  
Properties of the K-means algorithm  
  - Initialization  
    - fix the cluster clusters, find the cluster membership  
    - fix the cluster membership, update the cluster center  
  - Repeat the above two steps until convergence  
Comparing K-means clusterings  
A Numerical Example  
  - Input data, plot them in 1-d space  
  - Pick the initial cluster centers  
  - Run k-means algorithm one iteration
Show how the cluster membership changes
Show how the cluster centers change
K-means algorithm considerations

**Module 3: K-means Variants**

K-means as matrix factorization

The k-means problem
  Input of k-means
  Mathematical formulation
  Two special case (k=1 vs. k=n)

Hardness of K-means problem
  When d>2, k-means is NP-hard
  When d=1, k-means is polynomially solvable

Optimality of K-means
  In general, it only finds a local optimum
  Convergence of kmeans
  The impact of initial cluster centers
  A numerical example about the impact of initial cluster centers
  Impact of outlier

Alternatives to random initialization
  Multiple runs
  kmeans++

**Unit 6: Unsupervised Learning: Dimensionality Reduction**

**Learning Objectives**

6.1 Illustrate the process of dimensionality reduction
6.2 Apply the PCA algorithm
6.3 Explain the relationship between PCA and SVD

**Module 1: Introduction to Dimensionality Reduction**

What is dimensionality reduction?
The role of dimensionality reduction in machine learning

**Module 2: Using Principal Component Analysis (PCA)**

Introduction to using PCA
  Inputs of PCA
  Outputs of PCA
  A Numerical example
Maximizing the projected variance for the numerical example (d=1)
   How to calculate the projected data using original data and projection direction
   How to calculate the projected mean
   How to calculate projected variance

Maximizing the projected variance for the general case (d=1)
   One projected data
   Projected sample mean
   Sample variance matrix
   Projected variance

Optimization formulation for PCA (d=1)
   Objective function
   Constraint & why we need it
   Optimization variable

Solving the optimization problem for PCA (d=1)
   Overall strategy: lagrangian
   Step 1: write down the lagrangian function
   Step 2: calculate the partial derivative
   Step 3: set the partial derivative to zero
   Step 4: plug in step 3 back to the objective function J
   Step 5: seek for the largest eigenvalue of S

Solving the optimization problem for PCA (d>1)
   Fact: d principle components are the first d eigenvectors of the sample variance matrix S
   Prove it by induction
      Step 0: Base case
      Step 1: projected variance when d>1
      Step 2: the optimization formulation
      Step 3: solve the optimization problem using lagrangian

Minimizing the reconstruction error
   Input data
   Projected data
   Reconstruction error
   Minimizing reconstruction error = maximizing projected variance

A matrix representation for minimizing reconstruction error
   Assumption
   Input data matrix
   Projected data matrix
   PC matrix
   Objective function

PCA versus SVD
Assumption
Input data matrix X
SVD of X
  Left singular matrix = projected data matrix
  Singular value matrix and right singular vector matrix = PC matrix

PCA versus Feature Selection
Input data matrix
Rows of input data matrix
Columns of input data matrix
Two key points of PCA
  Un-supervised learning
  Generate a few new features
Two key points of feature selection
  Typically supervised learning
  Select a few original features

Unit 7: Deep Learning: Key Techniques

Learning Objectives
7.1: Describe the big-picture view of how neural networks work.
7.2: Identify the basic building blocks and notations of deep neural networks.
7.3: Explain how in principle learning is achieved in a deep network.
7.4: Explain key techniques that enable efficient learning in deep networks.
7.5: Appraise the detailed architecture of a basic convolutional neural network.
7.6: Compare the basic concepts and corresponding architecture for recurrent neural networks and autoencoders.

Module 1: Neural Networks and Deep Learning
Brief historical view of artificial neural network and deep learning
Early models of artificial neural network and their learning algorithms
Deep learning: what it is and what it is not

Module 2: Key Techniques Enabling Deep Learning
Back-propagation algorithm for learning
Choice of activation functions
A few regularization methods

Module 3: Some Basic Deep Architecture
Convolutional Neural Network
Recurrent Neural Networks
Unit 8: Deep Learning: Exemplar Applications

Learning Outcomes
8.1: Appraise image classification for deep learning
8.2: Appraise video-based inference for deep learning
8.3: Appraise Generative Adversarial Networks (GANs) for deep learning
8.4: Design a deep network using an exemplar application to solve a specific problem

Module 1: Image Classification
A typical network architecture used for image classification
Parameters for defining an image classification network
Common tricks for improving classification performance

Module 2: Video-Based Inference
Challenges in using deep networks for sequential data
Difference between image-based and video-based classification
Using video action recognition to contrast the difference between these classification tasks
A sample network for video-based inference

Module 3: Generative Adversarial Networks (GANs)
Basic concepts behind GANs
GANs variants and their applications

Creators
Established in Tempe in 1885, Arizona State University (ASU) has developed a new model for the American Research University, creating an institution that is committed to access, excellence and impact.

As the prototype for a New American University, ASU pursues research that contributes to the public good, and ASU assumes major responsibility for the economic, social and cultural vitality of the communities that surround it. Recognizing the university’s groundbreaking initiatives, partnerships, programs and research, U.S. News and World Report has named ASU as the most innovative university all three years it has had the category.

The innovation ranking is due at least in part to a more than 80 percent improvement in ASU’s graduation rate in the past 15 years, the fact that ASU is the fastest-growing research university in the country and the emphasis on inclusion and student success that has led to more than 50 percent of the school’s in-state freshman coming from minority backgrounds.

Jingrui He
Jingrui He is an assistant professor in the School of Computing, Informatics, and Decision Systems Engineering at Arizona State University. She received her Ph.D. from Carnegie Mellon University. She joined ASU in 2014 and directs the Statistical Learning Lab (STAR Lab). Her research focuses on rare category analysis, heterogeneous machine learning, active learning and semi-supervised learning, with applications in social media analysis, healthcare, manufacturing process, etc.

Baoxin Li

Baoxin Li is currently a professor and the chair of the Computer Science & Engineering Program and a Graduate Faculty Endorsed to Chair in the Electrical Engineering and Computer Engineering programs. From 2000 to 2004, he was a Senior Researcher with SHARP Laboratories of America, where he was the technical lead in developing SHARP’s HiIMPACT Sports™ technologies. He was also an Adjunct Professor with the Portland State University from 2003 to 2004. His general research interests are on visual computing and machine learning, especially their application in the context of human-centered computing.

Hanghang Tong

Hanghang Tong is currently an assistant professor at School of Computing, Informatics, and Decision Systems Engineering (CIDSE), Arizona State University since August 2014. Before that,
he was an assistant professor at Computer Science Department, City College, City University of New York, a research staff member at IBM T.J. Watson Research Center and a Post-doctoral fellow in Carnegie Mellon University. His research interest is in large scale data mining for graphs and multimedia.