CS 6140: Machine Learning
Spring 2016

Instructor: Lu Wang
College of Computer and Information Science
Northeastern University
Webpage: www.ccs.neu.edu/home/luwang
Email: luwang@ccs.neu.edu

Time and Location

• Time: Thursdays from 6:00 pm – 9:00 pm
• Location: Behrakis Health Sciences Cntr 325

Course Webpage

• http://www.ccs.neu.edu/home/luwang/courses/cs6140_sp2016.html

Prerequisites

• Programming
  – Being able to write code in some programming languages (e.g. Python, Java, C/C++) proficiently

• Courses
  – Algorithms
  – Probability and statistics
  – Linear algebra

• Some knowledge in machine learning
• Background Check

Textbook and References

• Main Textbook

• Other textbooks

• Machine learning lectures

Content of the Course

• Regression: linear regression, logistic regression
• Dimensionality Reduction: Principal Component Analysis (PCA), Independent Component Analysis (ICA), Linear Discriminant Analysis
• Probabilistic Models: Naive Bayes, maximum likelihood estimation, bayesian inference
• Statistical Learning Theory: VC dimension
• Kernels: Support Vector Machines (SVMs), kernel tricks, duality
• Sequential Models and Structural Models: Hidden Markov Model (HMM), Conditional Random Fields (CRFs)
• Clustering: spectral clustering, hierarchical clustering
• Latent Variable Models: K-means, mixture models, expectation-maximization (EM) algorithms, Latent Dirichlet Allocation (LDA), representation learning
• Deep Learning: feedforward neural network, restricted Boltzmann machine, autoencoders, recurrent neural network, convolutional neural network
• and others, including advanced topics for machine learning in natural language processing and text analysis

2/26/16
The Goal

• Not only what, but also why!

Grading

• Assignments
  — 3 assignments, 10% for each

• Exam
  — 1 exam, 30%
  — March 31, 2016

• Project
  — 1 project, 35%

• Participation
  — 5%
  — Classes
  — Piazza

Exam

• Open book
• March 31, 2016

Course Project

• Option 1: A machine learning relevant research project

• Option 2: Yelp Challenge

• 2-3 students as a team

Research Project

• Machine learning relevant
  — Natural language processing
  — Computer vision
  — Robotics
  — Bioinformatics
  — Health informatics
  —...

• Novelty

Yelp Challenge

The Challenge Dataset:
- 1.6M reviews and 500K tips by 366K users for 61K businesses
- 481K business attributes, e.g., hours, parking availability, ambience.
- Social network of 366K users for a total of 2.9M social edges.
- Aggregated check-ins over time for each of the 61K businesses

Get the Data
Yelp Challenge

Natural Language Processing (NLP): How well can you guess a review’s rating from its text alone? What are the most common positive and negative words used in our reviews? Are Yelpers a sarcastic bunch? And what kinds of correlations do you see between tips and reviews: could you extract tips from reviews?

Social Graph Mining: Can you figure out who the trend setters are and who found the best waffle joint before waffles were cool? How much influence does my social circle have on my business choices and my ratings?

Yelp Challenge

Cultural Trends: By adding a diverse set of cities, we want participants to compare and contrast what makes a particular city different. For example, are people in international cities less concerned about driving in to a business, indicated by their lack of mention about parking? What cuisines are Yelpers raving about in these different countries? Do Americans tend to eat out late compared to the Germans and English? In which countries are Yelpers sticklers for service quality? In international cities such as Montreal, are French speakers reviewing places differently than English speakers?

Infer Categories: Do you see any non-intuitive correlations between business categories e.g., how many karaoke bars also offer Korean food, and vice versa? What businesses deserve their own subcategory (i.e., Szechuan or Hunan versus just “Chinese restaurants”), and can you learn this from the review text?

Course Project Grading

• We want to see novel and interesting projects!
  – The problem needs to be well-defined, novel, useful, and practical
  – machine learning techniques
  – Reasonable results and observations

Course Project Grading

• Three reports
  – Proposal (5%)
  – Progress, with code (10%)
  – Final, with code (10%)

• One presentation
  – In class (10%)

Submission and Late Policy

• Each assignment or report, both electronic copy and hard copy, is due at the beginning of class on the corresponding due date.

• Electronic version
  – On blackboard

• Hard copy
  – In class

Submission and Late Policy

• Assignment or report turned in late will be charged 10 points (out of 100 points) off for each late day (i.e. 24 hours).

• Each student has a budget of 5 days throughout the semester before a late penalty is applied.
How to find us?

- Course webpage:
  - http://www.ccs.neu.edu/home/luwang/courses/cs6140_sp2016.html
- Office hours
  - Lu Wang: Thursdays from 4:30pm to 5:30pm, or by appointment, 448 WVH
  - Gabriel Bakiewicz (TA), Mondays and Tuesdays from 4:00pm to 5:00pm, 362 WVH
- Piazza
  - http://piazza.com/northeastern/spring2016/cs6140
  - All course relevant questions go here

What is Machine Learning?

- “A set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decisions making under certainty.”

Real World Applications
Real World Applications

Relations with Other Areas

- Natural Language Processing
- Computer Vision
- Robotics
- A lot of other areas...
Today’s Outline

• Basic concepts in machine learning
• K-nearest neighbors
• Linear regression
• Ridge regression

Supervised vs. Unsupervised Learning

• Supervised learning
  \[ D = \{(x_i, y_i)\}_{i=1}^N \]
  Training set  Training sample  Gold-standard label
  - Classification, if categorical
  - Regression, if numerical

Supervised Learning

- Goal: \( y = f(x) \)
  – Generalizable to new input samples
  – Overfitting vs. underfitting
  – we use probabilistic models

- Typical setup:
  – Training set, test set, development set
  – Features
  – Evaluation
Supervised Learning

- Regression
  - Predicting stock price
  - Predicting temperature
  - Predicting revenue
  - ...

Supervised vs. Unsupervised Learning

- Unsupervised Learning
  \[ \mathcal{D} = \{ x_i \}_{i=1}^N \]

- More about "knowledge discovery"

Unsupervised Learning

- Dimension reduction
  - Principal component analysis
Unsupervised Learning

- Clustering (e.g. graph mining)

RolX: Role Extraction and Mining in Large Networks, by Henderson et al, 2011

Unsupervised Learning

- Topic modeling


Parametric vs. Non-parametric model

- Fixed number of parameters?
  - If yes, parametric model

- Number of parameters grow with the amount of training data?
  - If yes, non-parametric model

- Computational tractability

A non-parametric classifier: K-nearest neighbors (KNN)

- Basic idea: memorize all the training samples
  - The more you have in training data, the more the model has to remember

- Nearest neighbor:
  - Testing phase: find closest sample, and return corresponding label
  \[
  \hat{y}(x) = y_{n^*} \text{ where } n^* = \arg\min_{n \in D} dist(x, x_n)
  \]

A non-parametric classifier: K-nearest neighbors (KNN)

- Basic idea: memorize all the training samples
  - The more you have in training data, the more the model has to remember

- K-Nearest neighbor:
  - Testing phase: find the K nearest neighbors, and return the majority vote of their labels
  \[
  \hat{y}(x) = y_{n^*} \text{ where } n^* = \arg\min_{n \in D} dist(x, x_n)
  \]
About K

- $K=1$: just piecewise constant labeling
- $K=N$: global majority vote (class)

Problems of kNN

- Can be slow when training data is big
  - Searching for the neighbors takes time
- Needs lots of memory to store training data
- Needs to tune $k$ and distance function
- Not a probability distribution

Distance function

- Euclidean distance
  \[
  D(u,v)^2 = ||u-v||^2 = (u-v)^T(u-v) = \sum_{i=1}^d (u_i-v_i)^2
  \]

- Mahalanobis distance: weights on components
  \[
  D(u,v)^2 = (u-v)^T \Sigma (u-v) = \sum_i \sum_j (u_i-v_i) \Sigma_{ij} (u_j-v_j)
  \]

Probabilistic kNN

- We prefer a probabilistic output because sometimes we may get an “uncertain” result
  - 99 samples as “yes”, 101 samples as “no” → ?
- Probabilistic kNN:
  \[
  p(y|x,D) = \frac{1}{K} \sum_{j \in \text{nb}(x,K,D)} I(y = y_j)
  \]
Probabilistic kNN

3-class synthetic training data

Softmax

- Class 1: 3, class 2: 0, class 3: 1
- Original probability:
  \[ P(y=1)=\frac{3}{4}, \quad P(y=2)=\frac{0}{4}, \quad P(y=3)=\frac{1}{4} \]
- Add-1 smoothing:
  
  \[ P(y=1)=\frac{4}{7}, \quad P(y=2)=\frac{1}{7}, \quad P(y=3)=\frac{2}{7} \]

A parametric classifier: linear regression

- Assumption: the response is a linear function of the inputs
  \[ y(x) = w^T x + \epsilon = \sum_{j=1}^{D} w_j x_j + \epsilon \]

- Inner product between input sample \( x \) and weight vector \( W \)
- Residual error: difference between prediction and true label

A parametric classifier: linear regression

- Assume residual error has a normal distribution
  \[ p(y|\mu(x), \sigma^2(x)) = \mathcal{N}(y|\mu(x), \sigma^2(x)) \]
A parametric classifier: linear regression

\[ p(y|\mathbf{x}, \theta) = \mathcal{N}(y|\mu(\mathbf{x}), \sigma^2(\mathbf{x})) \]

- We can further assume
  \[ \mu = \mathbf{w}^T \mathbf{x} \]
  \[ \sigma^2(\mathbf{x}) = \sigma^2 \]
- Basic function expansion
  \[ p(y|\mathbf{x}, \theta) = \mathcal{N}(y|\mathbf{w}^T \phi(\mathbf{x}), \sigma^2) \]
  \[ \phi(\mathbf{x}) = [1, x, x^2, \ldots, x^p] \]

Learning with Maximum Likelihood Estimation (MLE)

- Maximum Likelihood Estimation (MLE)
  \[ \hat{\theta} \triangleq \operatorname{arg\,max}_{\theta} \log p(D|\theta) \]

Derivation of MLE for Linear Regression

- Rewrite our objective function as
  \[ \operatorname{NLL}(\mathbf{w}) = \frac{1}{2}(\mathbf{y} - \mathbf{Xw})^T(\mathbf{y} - \mathbf{Xw}) = \frac{1}{2} \mathbf{w}^T(\mathbf{X}^T \mathbf{X}) \mathbf{w} - \mathbf{w}^T(\mathbf{X}^T \mathbf{y}) \]
Derivation of MLE for Linear Regression

- Rewrite our objective function as
  \[ \text{NLL}(w) = \frac{1}{2} (y - Xw)^T (y - Xw) = \frac{1}{2} w^T (X^T X) w - w^T (X^T y) \]

- Get the derivative (or gradient)
  \[ g(w) = \sum_{i=1}^{N} x_i (w^T x_i - y_i) \]

Geometric Interpretation

- Remember we have
  \[ \hat{w}_{OLS} = (X^T X)^{-1} X^T y \]

- Therefore the projected value of \( y \) is
  \[ \hat{y} = X\hat{w} = X (X^T X)^{-1} X^T y \]
  This corresponds to an orthogonal project of \( y \) onto the column space of \( X \)

Overfitting
A Prior on the Weight

- Zero-mean Gaussian prior
  \[ p(w) = \prod_j \mathcal{N}(w_j | 0, \tau^2) \]

- New objective function
  \[ \argmax_w \sum_{i=1}^N \log \mathcal{N}(y_i | w_0 + w^T x_i, \sigma^2) + \sum_{j=1}^{D} \log \mathcal{N}(w_j | 0, \tau^2) \]

Ridge Regression

- We want to minimize
  \[ J(w) = \frac{1}{N} \sum_{i=1}^N (y_i - (w_0 + w^T x_i))^2 + \lambda \| w \|_2^2 \]

- New estimation for the weight
  \[ \hat{w}_{ridge} = (\lambda I_D + X^T X)^{-1} X^T y \]
What we learned

• Basic concepts in machine learning
• K-nearest neighbors – non-parametric
• Linear regression – parametric
• Ridge regression – parametric

Homework

• Reading Murphy ch1, ch2, and ch7
• Sign up at Piazza
  • http://piazza.com/northeastern/spring2016/cs6140
• Start thinking about course project and find a team!
  – Project proposal due Jan 28th