Machine Learning
Finding Patterns in the World

Mark Dredze
mdredze@cs.jhu.edu     www.dredze.com

Human Language Technology Center of Excellence (HLTCOE)
Center for Language and Speech Processing (CLSP)
Johns Hopkins University
• What is machine learning?
  – Teaching a computer about the world

• How to apply machine learning?
  – Observe the world
  – Develop models that match observations
  – Teach computer to learn these models
  – Computer applies learned model to the world
Example: Animal Pictures
Machine Learning: Fundamental Questions

• What does it mean to learn?
• What can we output?
• How to represent the world for the computer?
• Motivations and learning algorithms
• How can we guide machine learning?
What Does it Mean to Learn?

• Learn patterns in data
• A system has learned when it can:
  – Take input
    • Pictures
  – Provide output:
    • Belongs in house/doesn’t belong in house
Finding Patterns of Interest
Finding Patterns of Interest
Unsupervised Learning

• Look for patterns in data
• No examples of output
• Pro:
  – No labeling of examples for output
• Con:
  – Cannot demonstrate specific types of output
• Applications:
  – Data mining
    • Finds interesting patterns in data
Learning Provided Patterns
Supervised Learning

• Learn patterns to simulate given output
• Pro:
  – Can learn complex patterns
  – Good performance
• Con:
  – Requires many examples of output for examples
• Applications:
  – Classification
    • Sorts data into predefined groups
What Does it Mean to Learn?

• **Input:** \( \{x_i\}_{i=1}^N \) \( x_i \in \mathbb{R}^M \) \( \{y_i\}_{i=1}^N \) \( y_i \in \mathbb{R} \) or \( y_i \in \{L\} \)

• **Loss function** \( \ell(h(x),y) \geq 0 \)

• **Hypothesis class** \( h^* \in H \)

\[
\sum_{i=1}^{N} \ell(h^*(x_i),y_i) \leq \varepsilon, \quad \varepsilon = 0 \vee \varepsilon \text{ is small}
\]

• **Learning algorithm** \( A \)

\( \hat{h} = A(\{x_i,y_i\}_{i=1}^N) \)

— such that \( \hat{h} = \arg\min_h \sum_{i=1}^{N} \ell(h(x_i),y_i) \)
• **Loss**
  – Measures system performance
  – Depends on output type/goal
  – Generalization error (over-fitting)

• **Input:**
  – Divide into
    – Train: learn $h$
    – Development: tune parameters of $A$
    – Test: Evaluate $h$

– **Output:**
  – Learned model
    • Type depends on learning algorithm
Types of Learning: Output

• Classification
  – Binary, multi-class, multi-label, hierarchical, etc.
    • Classify email as spam vs. ham
    • Loss: accuracy

• Ranking
  – Order examples by preference
    • Rank results of web search
    • Loss: Swapped pairs

• Regression
  – Real-valued output
    • Predict the price of tomorrow’s stock price
    • Loss: Squared loss

• Structured prediction
  – Sequences, trees, segmentation
    • Find faces in an image
    • Loss: Precision/Recall of faces
Example: Document Classification

• Group a collection of articles by topic

  The Chicago Cubs played a great game of baseball.
  **Sports**

  The markets rallied today sending the S&P 500 to a new 3 week high to start the financial year.
  **Finance**

• Problem characteristics
  – Classification- assign a class label (topic) to each document
    • Assume binary
  – Supervised- provided with labeled articles
How Do We Represent Data?

• NLP
  – Bag of words, bi-grams
  – “The Chicago Cubs played a great game of baseball.”
    • the | chicago | cubs | played | a | great ... 
    • the chicago | chicago cubs | cubs played | played a

• Other problems have different representations
  – Speech signal
  – Images
  – DNA
Supervised Approach to Learning

• Remainder of tutorial
  – Supervised
    • Nearest Neighbors
    • Decision Trees
    • Artificial Neural Networks
    • Perceptron
    • Support Vector Machines
  – Unsupervised
    • K-Means
    • Gaussian Mixture Models
Nearest Neighbors

• Motivation:
  – “This document has the same label as the most similar document I have seen.”

• Example:
  – This document is about baseball. The last baseball article I saw was about sports. This is about sports.

• Approach:
  – Save every example in the training set
  – For a test example:
    • Find the closest training example
    • Apply the label from this training example
Nearest Neighbors
• Similarity function
  – Euclidian distance
• How to choose K?
  – Small K is fast, finds single closest example
  – Large K slower, smooths outliers
• How do we know we have learned?
  – Training error: select parameters (K) that minimize error on training examples?
    • Note: K=1 always gives an error of 0
  – Some work on estimating true generalization error
• Bias/variance tradeoff
  – As we increase K
    • Bias- increases towards most popular labels
    • Variance- decreases
  – In practice:
    • Select K using development data
    • Reflective of actual performance
• Tradeoff key to generalization to new data
Summary

• Pros:
  – Easy to implement, understand output, complex functions

• Cons:
  – Need to store every observed example
  – Choosing similarity metric
  – Slow classification

• Useful extensions:
  – Learn similarity metrics
    • Large Margin Nearest Neighbor (LMNN)
  – Efficient comparisons
• Motivation:
  – “I can decide about a document by incrementally considering its properties.”

• Example:
  – This document says the word “baseball.” So its about sports. If it did not, I would next check if it said “finance”, then...

• Approach:
  – Construct a “tree of decisions” to follow, where a leaf applies a label to the document
Decision Tree

GAME

Yes

BASEBALL

Yes

Sports

No

FOOTBALL

Yes

Sports

No

Finance

No

MARKETS

Yes

Finance

No

Sports
• ID3 Algorithm
  – Greedily add most discriminating features

• ID3 (Examples, target_attribute, attributes)
  – If all target_attribute examples have the same label, apply label
  – Else
    • A = attribute that best classifies examples
    • Add branches for each value of attribute
    • Create subtree from: ID3(examples, A, attributes – A)
• Complexity of tree
  – How many levels?

• Pruning
  – Reduces over-fitting
  – C45 algorithm

• Selecting informative choices
  – Which features to select at each point?
  – “attribute that best classifies examples”
    • Information entropy common choice
• **Pros:**
  – Very easy to understand (“white box”), good for identifying a few critical features, fast

• **Con:**
  – Very slow to train, over-fitting, limited powers of representations (XOR), optimal trees NP-Complete

• **Useful extensions:**
  – Real valued data
  – Decision stumps for boosting
  – Random forests (ensemble approach)
Artificial Neural Networks

• Motivation
  – Extract linear combinations from input
  – Output nonlinear function of these combinations
  – Multiple functions performed in parallel
  – Based on neural networks in the brain
    • The result is a lot of hype

• Approach
  – Construct a graph of neural connections
  – Define input and output nodes
  – Learn hidden internal nodes
Single Hidden Layer Feed Forward Neural Network

Output

Hidden Layer

Input

Y₁ … Yₖ

Z₁ Z₂ … Zₘ

X₁ X₂ X₃ … Xₚ

K(M+1) weights

M(P+1) weights
• Activation function for nodes
  – Often chosen as the sigmoid

\[
\sigma(x) = \frac{1}{1 + e^{-x}}
\]

– Source for algorithm name
  • Neuron’s have activation threshold
• Minimize loss function
  – Regression: squared error

\[ R(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ik} - f_k(x_i))^2 \]

K- outputs
N- examples
y- correct output
f- NN output
x- example

• Back-propogation
  – Gradient descent on minimizing R
  – Sweep forward and backward over the network
    • Only need to compute local values
    • Similar to EM learning, HMM forward/backward training
  – Problems: local minima, over-fitting, initialization
Design Decisions

• Learning Algorithm
  – Issues: local minima, over-fitting, training time

• Network Structure
  – Number of hidden layers, nodes per layer

• Loss function
  – Regression: squared error
  – Classification: cross-entropy

• Network type
  – Different functions in network
    • Radial basis function networks
• Pros:
  – Can learn non-linear functions
  – Multiple outputs at once
• Cons:
  – Not easily interpretable (difficult to influence)
• Extensions:
  – Many! Whole conferences and journals on NNs
  – Applications to supervised, unsupervised learning
  – Deep belief networks (Hinton et al.)
Single Hidden Layer Feed Forward Neural Network
Single Hidden Layer Feed Forward Perceptron Neural Network
• 1958 by Frank Rosenblatt (Cornell)  
  – Grew from work on learning and the brain  
• One of the oldest and (still) most effective learning algorithms  
  – Works well on a large number of problems  
• Known as single layer neural network
Perceptron Algorithm

• Linear classification
  \( \hat{y} = \text{sign}(w \cdot x) \)  
  w- weight vector 
  x- example

• Learn weight vector w
  – Incrementally update w based on input 
  – Minimize number of mistakes 
  – Yields stochastic gradient descent algorithm 

  \[ w_i = w_{i-1} + y_i x_i \]

  mistake driven
Geometric Motivation
• Learning guarantee
  – If a separating hyperplane exists, will find separator with finite number of examples

• Problems:
  – Finite can still be large (slow convergence)
  – Many correct hyperplanes
    • Which one is the best
  – Sometimes separator doesn’t exist
    • Outliers, noisy labels, etc.
    • Will never converge
• Motivation:
  – Similar to perceptron
  – Which is best separator:
    • Hyperplane with maximum margin
      – Margin- distance between examples and separator

• Approach:
  – Define optimization problem given training data
  – Learn best separator
Geometric Motivation
SVM Learning

• “Support vectors”
  – Problem can formulated as combination of input vectors

• Convex QP problem
  – Many efficient algorithms to solve

• Formulation allows outliers
  – Tolerance set through parameter
Summary

• Pro:
  – Very good performance, efficient learning

• Cons:
  – Hard to interpret results, scaling to large datasets

• Extensions:
  – Transductive/Semi-Supervised
  – Regression
  – One-class
  – Kernels for non-linear learning
• So far supervised
• Many applications without labels
  – No time/money to create labels
  – Very large dataset
  – Discover natural patterns in the data

• Example: document clustering
  – Group documents into groups
  – Idea: natural groups correspond to topics
Clustering: K-Means

• Motivation:
  – Examples are points in high dimensional space
  – Examples cluster together

• Approach:
  – Find a set of K clusters that best describe the data
  – Each cluster defined by centroid
  – Examples belong to cluster with closest centroid
Clustering: K-Means
Learning the Clusters

- **Learning**
  - Given K, find the best K clusters given the data
  - Assume Euclidean distance as similarity metric
    \[ d(x_i, x_{i'}) = \left\| x_i - x_{i'} \right\|^2 \]
  - Minimize the cluster scatter
    \[ C^* = \min_{C, \{m_k\}_{k=1}^K} \sum_{k=1}^K N_k \sum_{C(i)=k} \left\| x_i - m_k \right\|^2 \]
• Expectation-Maximization Algorithm
  – Expectation
    • Assign instances to closest clusters
  – Maximization
    • Compute mean of clusters based on assigned documents

• Recall
  – Minimizes the objective (reduces cluster scatter)
  – Guarantee: will converge to an optimal value
  – Note: optimal not necessarily global
• Model parameter
  – K: number of clusters

• Model parameter impacts bias/variance
  – Bias- smaller K biases towards popular clusters
    • Larger K, smaller bias
  – Variance- larger K means fewer clusters
    • Larger K, higher variance
  – Ideally- know the exact number of clusters
Gaussian Mixture Models

• K-means
  – Pro: simple, easy to learn
  – Con: clusters are the same geometric size
    • Just have a mean for each cluster

• Extension: Gaussian Mixture Models
  – Assume each cluster is a Gaussian distribution
  – Mean: center of the cluster
  – Variance: geometric size of cluster
  – Soft clustering in learning
    • Probability of instance from each cluster
• **Expectation**
  – Compute responsibility of clusters for examples

• **Maximization**
  – Compute new mean/variances based on example assignments

• GMM are subject of this afternoon’s lab
Other Types of Supervision

• Tutorial covered
  – Supervised (mostly)
  – Unsupervised

• Can combine supervised and unsupervised
  – Semi-supervised
    • Some labeled examples, many unlabeled examples
  – Partially-supervised
    • Incomplete information about labels
  – Semi-supervised clustering
    • Discover groups with some guidance
• **Survey Books in Machine Learning**
  – The Elements of Statistical Learning
    • Hastie, Tibshirani, Friedman
  – Pattern Recognition and Machine Learning
    • Bishop
  – Machine Learning
    • Mitchell

• **Questions?**

• **Contact info:**
  – mdredze@cs.jhu.edu  www.dredze.com