Machine Learning: CIS 472/572

Introduction

Instructor: Daniel Lowd
Based on slides by Vibhav Gogate, Pedro Domingos, and others.

Logistics

• Instructor: Daniel Lowd
  – Email: lowd@cs.uoregon.edu
  – Office: Deschutes 262
  – Office hours: W 3-4pm, F 2-3pm
• Teaching Assistant: Pedram Rooshenas
  – Email: pedram@cs.uoregon.edu
  – Office hours: TBD
• Web: http://www.cs.uoregon.edu/Classes/17W/cis472/
• Discussion Board: Piazza (link on web page)
• Homework Submissions and Grades: Canvas
• Written Assignments: Gradescope (probably?)

Evaluation

• 3 homeworks (30%)
  – Some programming, some exercises
• One midterm (40%)
  – 2/3rds of the way through
• One project (30%)
  – Apply machine learning to a real problem of your choice
    (Recommended: Participate in a contest on Kaggle.com.)
  – Groups allowed
  – Written report
  – Presentations during final exam time

Kaggle

What This Course Covers

Foundations for the theory and application of machine learning.

• ML Models
• ML Algorithms
• ML Theory
• ML Best Practices

This is not a tutorial on how to use machine learning – you must also understand why.

Source Materials

Primary source of readings:
  (Good intro, focuses on machine learning concepts before math.
  Free online. Not finished.)

Excellent supplements:
  (Great reference and in-depth coverage.)
• T. Mitchell, Machine Learning, McGraw-Hill, 1997. (Great intro, but old and expensive.)
• C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006
• R. Duda, P. Hart & D. Stork, Pattern Classification (2nd ed.), Wiley, 2000
• D. Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012. (Free online!)
• T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, 2009. (Free online!)
So What Is Machine Learning?

- Automating automation
- Getting computers to program themselves
- Writing software is the bottleneck
- Let the data do the work instead!

Another view on machine learning

- Machine learning = automated science (sort of)
- Goal is to go from raw data to useful knowledge
- An ML algorithm finds a theory to fit the data and background knowledge as well as possible.
- A theory is good if it has good predictive accuracy.

Machine Learning: Applications

Examples of what you will study in class in action!

Spam Filtering

Classify as “Spam” or “Not Spam”

Collaborative Filtering
Collaborative Filtering

Image Labeling

Machine learning has grown in leaps and bounds

• The main approach for
  – Speech Recognition
  – Robotics
  – Natural Language Processing
  – Computational Biology
  – Sensor networks
  – Computer Vision
  – Web
  – ...and many more each year...

Related Fields

• Fields that use machine learning:
  – Artificial intelligence
  – Computer vision
  – Natural language processing
  – Computational biology
  – Robotics
  – ...many more...

• Fields with similar goals to machine learning:
  – Statistics
  – Data mining
  – Data science
  – Psychology (developmental, cognitive)

• Fields used by machine learning:
  – Information theory
  – Numerical optimization
  – Computational complexity

Types of Learning

• Supervised (inductive) learning
  – Training data includes desired outputs

• Unsupervised learning
  – Training data does not include desired outputs
  – Find hidden structure in data

• Semi-supervised learning
  – Training data includes a few desired outputs

• Reinforcement learning
  – The learner interacts with the world via “actions” and tries to find an optimal policy of behavior with respect to “rewards” it receives from the environment

Types of Supervised Learning Problems

• Classification: predict a discrete value from a predefined set of values

• Regression: predict a continuous/real value

• Structured prediction: predict a complex output, such as a sequence or tree
What We’ll Cover

- **Supervised learning**: Decision tree induction, Instance-based learning, Bayesian learning, Neural networks, Support vector machines, Linear Regression, Model ensembles, Learning theory, etc.
- **General machine learning concepts and techniques**: Feature selection, cross-validation, algorithm evaluation, debugging your machine learning system.

Not covering:
- Clustering and unsupervised learning (453/553)
- Reinforcement learning (471/571)
- Probabilistic graphical models (471/571, 410/510pm)
- Structured prediction (e.g., machine translation, image segmentation, multi-label classification)

What We’ll Cover: Comparison

**Computer Science**
- **Core concepts**: Variables, conditionals, loops, functions, etc.
- **Key algorithms**: Mergesort, linked lists, binary search trees, breadth-first search, etc.
- **Process**: Debugging, software engineering, etc.

**Machine Learning**
- **Core concepts**: Classification, overfitting, underfitting, training set, etc.
- **Key algorithms**: Decision trees, nearest neighbor, linear models, etc.
- **Process**: Designing and debugging ML systems

Goal: Use course attributes to predict the student’s course rating (+2 = loved it, -2 = hated it).

How should we model this?

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>-4</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>-5</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>-6</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td></td>
</tr>
</tbody>
</table>

(Data is from CML appendix.)

The Big Picture

[Diagram showing the problem domain, data representation, objective function, evaluation, learning algorithm, predictive model, useful predictions, and domain insight]

Important Concepts

- **Data**: Labeled instances, e.g., emails marked spam/ham
  - Training set
  - Held-out set
  - Test set
- **Features**: Attribute-value pairs which characterize each x
- **Experimentation cycle**
  - Learn parameters (e.g., model probabilities) on training set
  - Tune hyperparameters on held-out set
  - Compute accuracy on test set
  - Very important: never “peek” at the test set!
- **Evaluation**
  - Accuracy: fraction of instances predicted correctly
- **Overfitting and generalization**
  - Want a classifier which does well on test data
  - Overfitting: fitting the training data very closely, but not generalizing well
  - We’ll investigate overfitting and generalization formally in a few lectures

This class mostly focuses on the red pieces, but the blue pieces are just as important!
ML in a Nutshell

- Tens of thousands of machine learning algorithms
- Hundreds new every year
- Every machine learning algorithm has three components:
  - Model Representation
  - Evaluation
  - Optimization

Model Representation

- Decision trees
- Instances
- Linear function (hyperplane)
- Neural networks
- Support vector machines
- Model ensembles
  - (Sets of rules / Logic programs)
  - (Graphical models (Bayes/Markov nets))
  - Etc.

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.

Optimization

- Combinatorial optimization
  - E.g.: Greedy search
- Convex optimization
  - E.g.: Gradient descent
- Constrained optimization
  - E.g.: Linear programming