Machine Learning (1/2)
Outline

• This Lecture (Wes)
  - Intro to Machine Learning
  - Relationship to Programming Languages
  - Taxonomy of ML Approaches
  - Basic Clustering
  - Basic Linear Models

• Next Lecture (Ray)
  - Advanced ML Algorithms (e.g., Baysean Learning, Decision Trees, Support Vector Machines, Neural Networks…)
  - Concerns and Evaluation Techniques
Machine Learning Defined

- **Machine learning** is a subfield of AI concerned with algorithms that allow computers to *learn*. There are two types of learning:
  - **Deductive** learning uses axioms and rules of inference to construct new true judgments. See “Automated Theorem Proving” lecture.
  - **Inductive** learning method extract rules and patterns out of massive datasets. Given many examples, they attempt to generalize. We'll discuss this now.
Machine Learning in Context

• Machine Learning is sometimes called the part of AI that **works in practice**. (cf. “AI complete”)

• ML combines statistics and data mining with algorithms and theory

• Successful applications of ML:
  - detecting credit card fraud; stock market prediction; speech and handwriting recognition; medical diagnosis; market basket analysis; …
ML in PL?

Why does ML belong in a PL course?

ML in PL?

- Often in PL we try to form judgments about complex human-related phenomena
- ML can help form the basis of an analysis:
  - e.g., readability, bug reports, path frequency, ...
- or ML can help automate an action:
  - e.g., specification mining, documentation, regression testing ...
- PL is often concerned with scalable analyses, which give rise to huge data sets
  - ML helps us to make sense of them
What You'll Learn

• What kinds of problems can & can't it solve?
• What should you know about ML?
  - How to cast a problem in ML terms (e.g., creating a descriptive model)
  - How to pick the right ML algorithm
  - How to evaluate the results
    • Relevant statistics (e.g., precision, recall)
    • Relative feature importance
    • Practical details
No Silver Bullet

- ML can be handy, but using it takes practice
- Researchers often incorrectly apply ML without understanding its principles
  - “They threw machine learning at it …”
- ML rarely gives guarantees about performance
- ML takes creativity
  - Forming the model (e.g., picking features)
  - Interpreting the results
ML Algorithm Types

• Output Types
  - **Numeric**. Examples: How tall will you be, based on your birth weight? How much will you charge to your credit card this month, based on last month?
    - ML example: linear regression
  - **Binary**. Example: Does this image contain a human face or not? Is calling A() after B() a bug or not?
    - ML example: decision tree
  - **Discrete**. Example: Is this office, game or system software? How many sorts of computer intrusions are there, based on attacker behavior?
    - ML example: k-means clustering
ML Algorithm Types

• Input Types
  - **Supervised.** Some provided training examples are labeled with the right answer. Example: here are five images with faces and five without to get you started, now tell me if this next image has a face or not; here are five resolved bug reports and five that were never resolved, now tell me if this next report will get resolved or not.
  - **Unsupervised.** No labeled answers. Example: here are ten network intrusions: how would you organize them? Here's some seismic data: notice anything?
Clustering

- **Clustering** is the classification of objects into different groups
- Clustering partitions a dataset into subsets such that elements of each subset share common traits
  - Most commonly: proximity in some distance metric
- Clustering is an unsupervised learning method
- **Hierarchical clustering** finds successive clusters using previously-established clusters
  - Top-down = divisive. Bottom-up = agglomerative.
Clustering Example

• Hierarchical agglomerative clustering, Euclidean distance
Clustering Example

- Hierarchical agglomerative clustering, Euclidean distance
Clustering Example

- Hierarchical agglomerative clustering, Euclidean distance
Clustering Example

- Hierarchical agglomerative clustering, Euclidean distance
Clustering Intuition

• Why is \{A,C\} \{B,D\} a bad clustering?
Q: Music (246 / 842)

• Identify this 1962 #1 hit song, written by Carol King and Gerry Goffin and first recorded by Little Eva, from the partial lyrics: "Everybody's doing a brand new dance now / I know you'll get to like it / If you give it a chance now / My little baby sister can do it with ease / It's easier than learning your ABC's"
Q: Cartoons (676 / 842)

• Name the magical girl franchise in which the eleven names and nicknames Usagi Tsukino, Serena, Princess Serenity, "odango atama", "meatball head", Mamoru Chiba, Prince Endymion, Prince Darien, Tuxedo Kamen, Tuxedo Mask and Darien Shields refer to only two distinct characters.
Learning Prose

1 (Clustering). It might be different if they had something in common, but they were on such polar ends of the pole.

58 (Regression). It was arguable that of his last ten relationships in the past four months that he didn't care about either one of them.

86 (Predictive Models). "Really? My best friends, the girl you look like parents are dentist too."

150 (Feature Power). But the most noticing thing that had changed was the color of her hair. It was titan red, bouncing waves running through it.

448 (Feature Correlation). He noticed her pet the dog that burly officer that was tied to the leash he was holding.
Q: Movies (349 / 842)

• Name the movie described by this Marin Independent Journal summary: "A young girl awakens in an alien landscape to discover she has accidentally killed a woman. She later conspires with three strangers to kill again."
K-Means Clustering

• The objects in a cluster should be close to each other

• Given a cluster $C$ and its mean point $m$, the badness (i.e., error or intra-cluster variance) of the cluster is the sum, over all objects $x$ in $C$, of distance$(x,m)$.

• The objective of the \textit{k-means algorithm} is to partition objects into $k$ clusters such that the sum of the intra-cluster variances is minimized
K-Means Algorithm

make $k$ initial mean points somehow
   each one is (will be) the center of a cluster!
assign each object to a cluster randomly
while you're not done
   put each object in the cluster it is closest to
      (i.e., in the cluster with the mean point it is closest to)
   for each cluster, recalculate where the mean point is
      (i.e., average all the objects now in the cluster)
K-Means Example (01/10)
K-Means Example (02/10)
K-Means Example (03/10)
K-Means Example (05/10)
K-Means Example (06/10)
K-Means Example (07/10)
K-Means Example (08/10)
K-Means Example (09/10)
K-Means Example (10/10)
K-Means is Usually Decent
But What If You Don't Know $K$?
Parameter Selection

- Glenn Ammons, Rastislav Bodík, James R. Larus: Mining specifications. POPL 2002: 4-16

they are merged. The process repeats until no more merges are possible. The PFSA learner modifies k-tails by comparing how likely two states are to generate the same k-strings.

The resulting PFSA accepts a superset of all the strings in the training scenarios, due to the generalizations performed by the learner. The parameter $N$ that controls the size of the training scenarios is chosen by the user to be large enough to include all of the interesting behavior. It is therefore very likely that the ends of the training scenarios contain uninteresting behavior. This is in fact what we see experimentally: the typical PFSA has a “hot” core with a few transitions that occur frequently, with the core surrounded by a “cold” region with many transitions, each of which occurs infrequently. The corer whittles away the “cold” region, leaving just the “hot” core.

The corer can not simply drop edges with low weights. Consider the PFSA in Figure 19 (edge labels are not important and are omitted). Four edges have a weight of 5, which is low compared to the three edges with a weight of 10000. However, any string through this PFSA must traverse the edge out of the start state and the edge into the end state. Despite their low weight, a string is more likely to traverse these edges than it is to traverse the edges with a weight of 10000. Thus, a better measure of an edge’s “heat” is its likelihood of being traversed while generating a string from the PFSA. The problem of computing this measure is known as the Markov
Linear Regression

• If only we could get something to pick those parameters for us!
  - Let's look at an algo that doesn't need them.

• **Linear regression** models the relationship between a **dependent variable** (what you want to predict) and a number of **independent variables** (features you can already measure) as a linear combination:
  - \( \text{Dep} = c_0 + c_1 \times \text{Indep}_1 + \ldots + c_n \times \text{Indep}_n \)
  - Linear regression finds \( c_0 \ldots c_n \) for you
Linear Regression as Machine Learning

- Linear regression is a **supervised** learning task
  - You provide labeled training data, consisting of the values of the features *and* the dependent variable associated with a number of instances

- The output is a **linear model**
  - A function that, given values for all the features, produces a numeric value for the dependent variable

- How is this model produced?
  - Call SAS, Minitab, Matlab, R, take a Stats course …
Regression Case Study: Bug Reports

- Software maintenance accounts for over $70 billion each year and is centered around bug reports. Unfortunately, 26-36% of bug reports are invalid or duplicates and must manually triaged and removed by developers. This takes time and money.

- If we could separate valid from invalid bug reports, we could save time and money.

- Goal: highlight some design decisions when using ML in practice
Regression Case Study: Bug Reports Preliminaries

- **Dependent Variable**: We want to know how long (in minutes) it will take a bug report to be resolved.
  - Low quality or invalid reports that take more than 30 days to resolve (say) are an expensive use of developer time. If we could predict this, we'd win!

- **Independent Variables**:
  - self-reported severity, readability, daily load, submitter reputation, comment count, attachment count, operating system used, …
Regression Case Study: Bug Reports Instances

- Gather all 27,984 non-empty bug reports between 01/01/2003 and 07/31/2005 (Firefox 1.5).
  - Each report is an instance (or feature vector)
  - Note the indep features (e.g., priority, readability)
  - Note the dependent feature (minutes to resolved)

- Feed to Linear Regression, get out coeffs
  - Are we done?
  - Let's look at some design decisions in using ML.
Regression Case Study: Input Dataset Threats

- Can I cherry-pick random bug reports?
- What if I take all reports 1 month after a beta release?
- What is the purpose of having a larger dataset?
Regression Case Study: Independent Variables

- All features for linear regression are real-valued (see next lecture for discrete features)
  - Comment count is easy enough
  - 1-bit saturating comment count
- How to encode “high/medium/low priority”?
- How to encode “operating system used”?
Regression Case Study: Dependent Variable

- How would these be different:
  - Resolved in X minutes
  - Resolved in X days
  - Resolved within 30 days => 1, otherwise => 0

- Linear Models give continuous output!
  - If you want a binary classifier, may need to pick a cutoff (e.g., model < 0.7 => 0, otherwise => 1)
Regression Case Study: Evaluation

- You have a **binary** classifier for “will this report be resolved in <= 30 days”
- You have 27,984 reports with known answers
  - C = *correct* set of reports resolved in 30 days
  - R = set of reports the **model** returns

- **Precision** = \(|C \cap R| / |R| |
- **Recall** = \(|C \cap R| / |C| |
- **F-Measure** = \((2 \times \text{Prec} \times \text{Rec}) / (\text{Prec} + \text{Rec})\)
Regression Case Study: Evaluation Baselines

• Say you have 100 instances

• 50 yes instances, 50 no instances, at random
  - “Flip Fair Coin”: Prec=0.5, Rec=0.5, F=0.5
  - “Always Guess Yes”: Prec=0.5, Rec=1.0, F=0.66

• 70 yes instances, 30 no instances, at random
  - “Flip Fair Coin”: Prec=0.7, Rec=0.5, F=0.58
  - “Flip Biased Coin”: Prec=0.7, Rec=0.7, F=0.7
  - “Always Guess Yes”: Prec=0.7, Rec=1.0, F=0.82

• May want to **subsample** to 50-50 split for evaluation purposes
Regression Case Study: Threats To Validity

• **Overfitting** occurs when you have learned a model that is too complex with respect to the data.
  - i.e., no actual abstraction has occurred
  - e.g., “memorize all input instances”

• **N-Fold Cross-Validation** can mitigate or detect the threat of overfitting
  - Partition instances into n subsets
  - Train on 2..n and test on 1
  - Train on 1, 3..n and test on 2, etc.
Regression Case Study: Final Results

• Given one day's worth of features, our best F-Measure for predicting “resolved within 30 days” was 0.76, and the industrial practice baseline was 0.73. Win?

• F-Measure assumes false positives and false negatives are equally bad
  - For bug reports, missing a bug report is much worse than triaging an invalid one

• IR metrics are good, but relating your results back to the real world is key:
  - “For the purposes of comparison, however, if Triage is $30 and Miss is $1000, using our model as a filter saves between five and six percent of the development costs for this data set.”
Next Time

- How to design features!
- Which features mattered?
- More exotic ML algorithms!
- How should we pick parameters?
- Practical information!