Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. (https://en.wikipedia.org/wiki/Machine_learning, retrieved 8/22/16)

- Our focus will be on supervised learning, where the ML system learns from labeled examples
  - There will be some space for unsupervised learning, too

In contrast:
- NLP systems and language resources for NLP based on machine learning techniques
  - require less human effort
  - are data-driven & require large-scale data sources
  - achieve coverage directly proportional to the richness of the data source
  - are more adaptive to noisy data

One priority for us, then, is to obtain appropriate data (more on that in the coming weeks)
- Adapting to new domains is also a challenge

Classification
- Classification:
  - Supervised learning = classification
  - Classification = assigning a label from a limited set of labels to an instance
  - Instance = pre-defined list of feature-value pairs
  - Unsupervised approaches: more general results (hierarchies, models)

We may also want to employ ranking amongst choices
- May be as simple as comparing classifier confidence for multiple guesses
- Makes the most sense for gradient properties
Example: Part-of-Speech Tagging

- Task: find the appropriate POS tag for a word in context
- e.g., They manVB the boat.
- Some useful links:

Sample instance:

<table>
<thead>
<tr>
<th>feature</th>
<th>word 2</th>
<th>tag 2</th>
<th>word 1</th>
<th>tag 1</th>
<th>word</th>
<th>POS tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>NULL</td>
<td>NULL</td>
<td>They</td>
<td>PRP</td>
<td>man</td>
<td>VB</td>
</tr>
</tbody>
</table>

The Learning Problem

- instance: a vector of feature values \( <f_1, f_2, \ldots, f_n > \)
- let \( X \) be the space of possible instances
- let \( Y \) be the set of classes
  - the goal of the ML system is to learn a target function \( c : X \rightarrow Y \)

The Learning Problem (2)

- Restrictions
  - Some learners require real-valued features, while some (e.g., timbl) allow for things like text features
  - Text features:
    - Per | Num | Gen | Class
    - 2   | pl   | neut | 1
  - Real-valued features:
    - Per | Num | Gen | Class
    - 1 2 3 | Sg Pl M F N | 0 1 0 0 1 1
- Likewise, some learners require binary decisions
  - though, they usually have techniques to convert \( n \)-ary decisions to binary internally

Some Important Concepts

- generalization: generalize from experience
- abstraction
  - vs. lazy learning: generalize when needed
  - non-abstraction can be useful when there are many exceptions & sub-regularities
- online learning: learn one instance at a time & thus continually refine model
- offline learning: learn as a batch

Text Classification

- Our task this semester falls under the general field of document classification, or text classification
- text classification ⊂ classification ⊂ machine learning
- Classification might be in terms of topics/subjects, document type, etc.
  - Generally different from our task, documents may contain multiple topics (any-of problem)
  - Topics may also be hierarchical, e.g., poultry and coffee as subclasses of industries
- Some useful links:

The Learning Problem (3)

- Training example: instance \( x \in X \) labeled with the correct class \( c(x) \)
  - let \( D \) be the set of all training examples
- Hypothesis space, \( H \): set of functions \( h : X \rightarrow Y \) of possible definitions
  - the goal is to find an \( h \in H \) such that for all \( x \in X \), \( c(x) \in D \), \( h(x) = c(x) \)
Some Machine Learning Algorithms

Supervised:
- decision tree learning
- memory-based learning (k-nearest neighbors)
- support vector machines (SVM)
- maximum entropy learning
- neural networks
- genetic programming
- naive Bayes

Unsupervised:
- clustering
- minimum description length

Evaluation

- **Training set**: data on which the ML program is trained
- **Test set**: data on which the performance of the ML program is measured
- **Gold standard**: data against which the ML program is evaluated
- **(Tenfold) Cross validation**: split data into 10 parts:
  - 10 rounds: use 1 part as test set and remaining parts as training set

Evaluation Metrics

Common metrics include:
- **Accuracy**: percentage of correctly classified instances from test set
- **Recall**: percentage of the items in the gold standard that were found by the ML program
- **Precision**: percentage of the items selected by the ML program that are correct

Sometimes also: sensitivity, specificity, area under the ROC curve (http://gim.unmc.edu/dxtests/roc3.htm), ...

Problems with ML

- Difficult to distinguish between noise and subregularities / irregularities
- Feature selection mainly by intuition
  - Though, see, e.g., deep learning (http://nlp.stanford.edu/courses/NAACL2013/)
  - Irrelevant information can deteriorate performance
- Skewed class distribution deteriorates performance
- Some tasks defined as classification may not naturally be classification

Do’s and Don’t’s in ML

**Don’t**:
- report without evaluation
- test on (any part of) your training set

**Do**:
- make your features independent (required for most algorithms)
- optimize your parameters
- optimize features & parameters at the same time
  - comparisons of learning algorithms are only meaningful if both are optimized
- beware of overfitting
- do what you can to get more data
Some Available Packages

Packages of multiple algorithms:
- weka (http://www.cs.waikato.ac.nz/ml/weka/)
- mallet (http://mallet.cs.umass.edu/): MMachine Learning for LanguageE Toolkit
- MLlib (https://spark.apache.org/mllib/): good for very large-scale learning
- MLPack (http://www.mlpack.org): “emphasis on scalability, speed, and ease-of-use”
- sofia-ml (https://code.google.com/p/sofia-ml/)

Choosing a Classifier

Bias & Variance

- Bias: (non-)ability to approximate the data, degree to which model makes assumptions about data distribution
  - “High bias is related to under-fitting” (e.g., linear regression for a quadratic relationship)
- Variance: (non-)stability in the face of new training examples
  - “High variance is related to over-fitting” (e.g., k-NN changes a lot depending on training set)
- Regularization parameters help control bias-variance tradeoff
  - e.g., penalize complex models, set $k$ higher
  (http://followthedata.wordpress.com/2012/06/02/practical-advice-for-machine-learning-bias-variance/)

Choosing a classifier

Some advice from Manning, Raghavan, & Schütze (2008):
- Can use rules, if the task is relatively simple
  - Rules can also work well for post-processing ML output
- If you have little data, use a classifier with high bias (e.g., Naïve Bayes)
  - Also can try semi-supervised training methods (e.g., bootstrapping, EM)
- If you have a huge amount of data, the classifier may have less of an impact on accuracy
  - Speed & ease of use become bigger questions

More packages

- Python Machine Learning kits:
  - Orange: http://orange.biolab.si/
  - mlpy: https://mlpy.fbk.eu/
  - PyML: http://pyml.sourceforge.net/
  - Anaconda Python has much pre-installed:
    - https://store.continuum.io/cshop/anaconda

Choosing a Classifier

Accounting for bias & variance

What to do? (originally from Andrew Ng)
- High variance?
  - get more training examples
  - try smaller sets of features
- High bias?
  - try new (more/different) features

To find whether it’s high variance or bias, compare training & testing (i.e., development) learning curves:
- cf. slides 7 & 8 of:
  (http://followthedata.wordpress.com/2012/06/02/practical-advice-for-machine-learning-bias-variance/)

Other Ways to Boost Performance

- Combine multiple classifiers
- Engage in feature engineering
  - Group together similar features (i.e., with similar votes)
  - Create new features for potentially informative concepts (e.g., subwords)
- Account for document structure (e.g., upweight items in a title, use separate feature spaces for different document zones)
Next time

We’ll spend some time getting into practical matters
  ▪ See your assignment for installations