Final project

1. Your project should relate to something involving NLP

2. Your project must include a solid experimental evaluation

3. Your project should be in a pair or group of three. If you’d like to do it solo or in a group of four, please come talk to me.

Final project

<table>
<thead>
<tr>
<th>date</th>
<th>time</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/18</td>
<td>in-class</td>
<td>Project proposal presentation</td>
</tr>
<tr>
<td>11/20</td>
<td>11:00pm</td>
<td>Project proposal write-up</td>
</tr>
<tr>
<td>12/2</td>
<td>2-4pm</td>
<td>Status report</td>
</tr>
<tr>
<td>12/10</td>
<td>5pm</td>
<td>Paper draft</td>
</tr>
<tr>
<td>12/16</td>
<td>2pm</td>
<td>Final paper, code and presentation</td>
</tr>
</tbody>
</table>

Read the final project handout ASAP!

Start forming groups and thinking about what you want to do.
Final project ideas

- Pick a text classification task
  - Evaluate different machine learning methods
  - Implement a machine learning method
  - Analyze different feature categories

- n-gram language modeling
  - Implement and compare after-smoothing techniques
  - Implement alternative models

- Parsing
  - Lexicalized PCFG (with smoothing)
  - n-best list generation
  - Parse output reranking
  - Implement another parsing approach and compare
  - Parsing non-traditional domains (e.g., Twitter)

- EM
  - Try and implement EM model 2
  - Word-level translation models

Final project ideas

- Spelling correction
- Part of speech tagger
- Text chunker
- Dialogue generation
- Pronoun resolution
- Compare word similarity measures (more than the ones we looked at)
- Word sense disambiguation
- Machine translation
- Information retrieval
- Information extraction
- Question answering
- Summarization
- Speech recognition

The mind-reading game

How good are you at guessing random numbers?

Repeat 100 times:
  - Computer guesses whether you’ll type 0/1
  - You type 0 or 1

http://seed.ucsd.edu/~mindreader/
[written by Y. Freund and R. Schapire]
The mind-reading game

The computer is right much more than half the time...

Strategy: computer predicts next keystroke based on the last few (maintains weights on different patterns)

There are patterns everywhere... even in “randomness”!

Machine Learning is...

Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data.

Machine Learning is...

Machine learning is programming computers to optimize a performance criterion using example data or past experience.

-- Ethem Alpaydin

The goal of machine learning is to develop methods that can automatically detect patterns in data, and then to use the uncovered patterns to predict future data or other outcomes of interest.

-- Kevin P. Murphy

The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions.

-- Christopher M. Bishop
Machine Learning is…

Machine learning is about predicting the future based on the past.
-- Hal Daume III

Why machine learning?

Lot's of data
Hand-written rules just don't do it
Performance is much better than what people can do

Why not just study machine learning?
- Domain knowledge/expertise is still very important
- What types of features to use
- What models are important

Be able to laugh at these signs
What high-level machine learning problems have you seen or heard of before?
Supervised learning: given labeled examples

Supervised learning: learn to predict new example
Supervised learning: classification

- apple
- banana

Classification: a finite set of labels

Supervised learning: given labeled examples

NLP classification applications

- Document classification
  - spam
  - sentiment analysis
  - topic classification
- Does linguistic phenomena X occur?
- Digit recognition
- Grammatically correct or not?
- Word sense disambiguation
- Any question you can pose as to have a discrete set of labels/answers!

Supervised learning: regression

- label
  - -4.5
  - 10.1
  - 3.2
  - 4.3

Regression: label is real-valued

Supervised learning: given labeled examples

Regression Example

Price of a used car

\[ y = wx + w_0 \]

\( x \): car attributes (e.g. mileage)
\( y \): price
Regression applications

- How many clicks will a particular website, ad, etc. get?
- Predict the readability level of a document
- Predict pause between spoken sentences?
- Economics/Finance: predict the value of a stock
- Car/plane navigation: angle of the steering wheel, acceleration, …
- Temporal trends: weather over time
- …

Supervised learning: ranking

Supervised learning: given labeled examples

NLP Ranking Applications

- Reranking N-best output lists (e.g. parsing, machine translation, …)
- User preference, e.g. Netflix “My List” -- movie queue ranking
- iTunes
- Flight search (search in general)
- …

Ranking example

Given a query and a set of web pages, rank them according to relevance
Unsupervised learning

Unsupervised learning: given data, i.e. examples, but no labels

Unsupervised learning applications

- learn clusters/groups without any label
  - cluster documents
  - cluster words (synonyms, parts of speech, ...)
- compression
- bioinformatics: learn motifs
  ...

Reinforcement learning

Reinforcement learning example

Given a sequence of examples/states and a reward after completing that sequence, learn to predict the action to take in for an individual example/state

Given sequences of moves and whether or not the player won at the end, learn to make good moves

Backgammon

- left, right, straight, left, left, left, straight GOOD
- left, straight, straight, left, right, straight, straight BAD
- left, right, straight, left, left, left, straight 18.5
- left, straight, straight, left, right, straight, straight -3

WIN!

LOSE!
Reinforcement learning example

http://www.youtube.com/watch?v=VCdOzpOfZmE

Other learning variations

- What data is available:
  - Supervised, unsupervised, reinforcement learning
  - Semi-supervised, active learning, …

- How are we getting the data:
  - Online vs. offline learning

- Type of model:
  - Generative vs. discriminative
  - Parametric vs. non-parametric

Text classification

- Label:
  - Spam
  - Not spam

- For this class, I’m mostly going to focus on classification

- I’ll use text classification as a running example

Representing examples

- Examples:
  - Apple
  - Banana

- What is an example?

- How is it represented?
Features

Examples

Features

How our algorithms actually “view” the data

Features are the questions we can ask about the examples

Text: raw data

Feature examples

Clinton said banana repeatedly last week on tv, “banana, banana, banana.”

Occurrence of words (unigrams)
Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, “banana, banana, banana”

(4, 1, 1, 0, 1, 0, 0, …)

Frequency of word occurrence (unigram frequency)

Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, “banana, banana, banana”

(1, 1, 1, 0, 1, 0, 0, …)

Occurrence of bigrams

Lots of other features

POS: occurrence, counts, sequence

Constituents

Whether 'Viagra' occurred 15 times

Whether 'banana' occurred more times than 'apple'

If the document has a number in it

…

Features are very important, but we're going to focus on the model
During learning/training/induction, learn a model of what distinguishes apples and bananas based on the features.

The model can then classify a new example based on the features.

**Why?**

The model can then classify a new example based on the features.
Classification revisited

<table>
<thead>
<tr>
<th>Training data</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>examples</strong></td>
<td><strong>label</strong></td>
</tr>
<tr>
<td>red, round, leaf, 3oz, ...</td>
<td>apple</td>
</tr>
<tr>
<td>green, round, no leaf, 4oz, ...</td>
<td>apple</td>
</tr>
<tr>
<td><strong>yellow, curved, no leaf, 4oz, ...</strong></td>
<td>banana</td>
</tr>
<tr>
<td>green, curved, no leaf, 5oz, ...</td>
<td>banana</td>
</tr>
</tbody>
</table>

Learning is about **generalizing** from the training data.

What does this assume about the training and test set?

Past predicts future

Not always the case, but we'll often assume it is!
More technically...

We are going to use the probabilistic model of learning.

There is some probability distribution over example/label pairs called the data generating distribution.

Both the training data and the test set are generated based on this distribution.
Probabilistic Modeling

Model the data with a probabilistic model
specifically, learn \( p(\text{features, label}) \)
\( p(\text{features, label}) \) tells us how likely these features and this example are

An example: classifying fruit

Training data

<table>
<thead>
<tr>
<th>examples</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>red, round, leaf, 3oz, ...</td>
<td>apple</td>
</tr>
<tr>
<td>green, round, no leaf, 4oz, ...</td>
<td>apple</td>
</tr>
<tr>
<td>yellow, curved, no leaf, 4oz, ...</td>
<td>banana</td>
</tr>
<tr>
<td>green, curved, no leaf, 5oz, ...</td>
<td>banana</td>
</tr>
</tbody>
</table>

Probabilistic models

Probabilistic models define a *probability distribution* over features and labels:

Probabilistic model vs. classifier

Probabilistic model:

| yellow, curved, no leaf, 6oz, banana | probabilistic model: \( p(\text{features, label}) \) | 0.004 |

Classifier:

| yellow, curved, no leaf, 6oz | probabilistic model: \( p(\text{features, label}) \) | banana |
Probabilistic models define a probability distribution over features and labels:

**probabilistic model:**
\[ p(\text{features, label}) \]

**yellow, curved, no leaf, 6oz, banana:** 0.004

Given an unlabeled example: yellow, curved, no leaf, 6oz predict the label

**How do we use a probabilistic model for classification/prediction?**

For each label, ask for the probability under the model
Pick the label with the highest probability

**Probabilistic model vs. classifier**

**Probabilistic model:**

**yellow, curved, no leaf, 6oz, banana:** 0.004

**Classifier:**

**yellow, curved, no leaf, 6oz:** banana

Why probabilistic models?

**Probabilistic models**

Probabilistic models define a probability distribution over features and labels:

**probabilistic model:**
\[ p(\text{features, label}) \]

**yellow, curved, no leaf, 6oz, banana:** 0.004

**yellow, curved, no leaf, 6oz, apple:** 0.00002

**Why probabilistic models?**

- Probabilities are nice to work with
  - range between 0 and 1
  - can combine them in a well understood way
  - lots of mathematical background/theory

- Provide a strong, well-founded groundwork
  - Allow us to make clear decisions about things like smoothing
  - Tend to be much less “heuristic”
  - Models have very clear meanings
Probabilistic models: big questions

1. Which model do we use, i.e. how do we calculate $p(\text{feature, label})$?
2. How do train the model, i.e. how to we estimate the probabilities for the model?
3. How do we deal with overfitting (i.e. smoothing)?

Basic steps for probabilistic modeling

Probabilistic models

- Step 1: pick a model
- Step 2: figure out how to estimate the probabilities for the model
- Step 3 (optional): deal with overfitting

What was the data generating distribution?

Step 1: picking a model

What we’re really trying to do is model the data generating distribution, that is how likely the feature/label combinations are.
Some maths

\[ p(\text{features}, \text{label}) = p(x_1, x_2, \ldots, x_m, y) \]

\[ = p(y)p(x_1, x_2, \ldots, x_m \mid y) \]

What rule?

Step 1: pick a model

\[ p(\text{features}, \text{label}) = p(y) \prod_{j=1}^{m} p(x_j \mid y, x_1, \ldots, x_{m-1}) \]

So, far we have made NO assumptions about the data

\[ p(x_m \mid y, x_1, x_2, \ldots, x_{m-1}) \]

How many entries would the probability distribution table have if we tried to represent all possible values and we had 7000 binary features?

Full distribution tables

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>p</th>
<th>p(y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>*</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>*</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All possible combination of features!

Table size: \(2^{7000} = ?\)
Step 1: pick a model

\[ p(\text{features}, \text{label}) = p(y) \prod_{j=1}^{m} p(x_j | y, x_1, \ldots, x_{j-1}) \]

So far we have made NO assumptions about the data

Model selection involves making assumptions about the data

We’ve done this before, n-gram language model, parsing, etc.

These assumptions allow us to represent the data more compactly and to estimate the parameters of the model

Naive Bayes assumption

\[ p(\text{features}, \text{label}) = p(y) \prod_{j=1}^{m} p(x_j | y, x_1, \ldots, x_{j-1}) \]

\[ p(x_j | y, x_1, x_2, \ldots, x_{j-1}) = p(x_j | y) \]

What does this assume?
Naïve Bayes assumption

\[ p(\text{features}, \text{label}) = p(y) \prod_{i=1}^{n} p(x_i | y, x_1, \ldots, x_{i-1}) \]

\[ p(x_i | y, x_1, x_2, \ldots, x_{i-1}) = p(x_i | y) \]

Assumes feature \( i \) is independent of the other features given the label

Is this true for text, say, with unigram features?

For most applications, this is not true!

For example, the fact that "San" occurs will probably make it more likely that "Francisco" occurs.

However, this is often a reasonable approximation:

\[ p(x_i | y, x_1, x_2, \ldots, x_{i-1}) \approx p(x_i | y) \]