Ensemble Classifiers

Mausam

(based on slides of Dan Weld)
Ensembles of Classifiers

• Traditional approach: Use one classifier
• Alternative approach: Use lots of classifiers
• Approaches:
  • Cross-validated committees
  • Bagging
  • Boosting
  • Stacking
Voting
Ensembles of Classifiers

• Assume
  – Errors are independent (suppose 30% error)
  – Majority vote

• Probability that majority is wrong...

  If individual area is 0.3
  Area under curve for $\geq 11$ wrong is 0.026
  Order of magnitude improvement!
Constructing Ensembles

Cross-validated committees

• Partition examples into \( k \) disjoint equiv classes

• Now create \( k \) training sets
  – Each set is union of all equiv classes except one
  – So each set has \( (k-1)/k \) of the original training data

• Now train a classifier on each set
Ensemble Construction II

Bagging

• Generate \( k \) sets of training examples
• For each set
  – Draw \( m \) examples randomly (with replacement)
  – From the original set of \( m \) examples
• Each training set corresponds to
  – 63.2\% of original (+ duplicates)
• Now train classifier on each set
• Intuition: Sampling helps algorithm become more robust to noise/outliers in the data
Ensemble Creation III

Boosting

- Maintain prob distribution over set of training ex
- Create k sets of training data iteratively:
  - On iteration $i$
    - Draw m examples randomly (like bagging)
    - But use probability distribution to bias selection
    - Train classifier number $i$ on this training set
    - Test partial ensemble (of $i$ classifiers) on all training exs
    - Modify distribution: increase $P$ of each error ex

- Create harder and harder learning problems...
- “Bagging with optimized choice of examples”
Ensemble Creation IV
Stacking

• Train several base learners
• Next train meta-learner
  – Learns when base learners are right / wrong
  – Now meta learner arbitrates

Train using cross validated committees
  • Meta-L inputs = base learner predictions
  • Training examples = ‘test set’ from cross validation
Example: Random Forests

- Create \( k \) decision trees
- For each decision tree
  - Pick training data as in bagging
  - Randomly sample \( f \) features in the data
  - Construct best tree based only on these features
- Voting for final prediction
- Advantages
  - Efficient, highly accurate, thousands of vars
Semi-Supervised Learning

Mausam

(based on slides of Dan Weld, Oren Etzioni, Tom Mitchell)
Semi-supervised learning Motivation

- Learning methods need labeled data
  - Lots of $<x, f(x)>$ pairs
  - Hard to get… (who wants to label data?)

- But unlabeled data is usually plentiful…
  - Could we use this instead??????

- Semi-supervised learning
Training Data Size

• Machine Translation and speech recognition are quite successful. Why?
• Plenty of labeled data
  – European parliament proceedings
  – Closed-caption broadcasts
• In MT, we have phrase tables
  – Blue bicycle $\Rightarrow$ bicicleta azul

• Side note: this is also a key win for price prediction for Farecast and Zillow.
NLP Challenges

- Document classification
- Named-entity recognition (person, place, or organization?)
- Part-of-speech tagging (verb, noun, or adjective?)
- Limited amount of labeled data.
- Labeling is expensive and slow.
Statistical learning methods require LOTS of training data

Can we use all that unlabeled text?
Our energy exploration, production, and distribution operations span the globe, with activities in more than 100 countries.

At TOTAL, we draw our greatest strength from our fast-growing oil and gas reserves. Our strategic emphasis on natural gas provides a strong position in a rapidly expanding market.

Our expanding refining and marketing operations in Asia and the Mediterranean Rim complement already solid positions in Europe, Africa, and the U.S.

Our growing specialty chemicals sector adds balance and profit to the core energy business.

```
<table>
<thead>
<tr>
<th>Word</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>aardvark</td>
<td>0</td>
</tr>
<tr>
<td>about</td>
<td>2</td>
</tr>
<tr>
<td>all</td>
<td>2</td>
</tr>
<tr>
<td>Africa</td>
<td>1</td>
</tr>
<tr>
<td>apple</td>
<td>0</td>
</tr>
<tr>
<td>anxious</td>
<td>0</td>
</tr>
<tr>
<td>gas</td>
<td>1</td>
</tr>
<tr>
<td>oil</td>
<td>1</td>
</tr>
<tr>
<td>Zaire</td>
<td>0</td>
</tr>
</tbody>
</table>
```
Twenty NewsGroups

Given 1000 training documents from each group. Learn to classify new documents according to which newsgroup it came from.

- comp.graphics
- comp.os.ms-windows.misc
- comp.sys.ibm.pc.hardware
- comp.sys.mac.hardware
- comp.windows.x
- alt.atheism
- soc.religion.christian
- talk.religion.misc
- talk.politics.mideast
- talk.politics.misc
- talk.politics.guns
- misc.forsale
- rec.autos
- rec.motorcycles
- rec.sport.baseball
- rec.sport.hockey
- sci.space
- sci.crypt
- sci.electronics
- sci.med

Naive Bayes: 89% classification accuracy
What if we have labels missing?

Learn $P(Y|X)$

<table>
<thead>
<tr>
<th>Y</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>?</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>?</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

EM Algorithm
Unsupervised Learning: Clustering

• K-means clustering algorithm:
  • Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
  • Assign each object to the group that has the closest centroid.
  • When all objects have been assigned, recalculate the positions of the K centroids.
  • Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.
20 Newsgroups

![Graph showing accuracy vs. number of documents for 10,000 unlabeled documents and no unlabeled documents.](image)

- Accuracy increases with the number of documents.
- Two lines represent different scenarios: 10,000 unlabeled documents and no unlabeled documents.
Co-training

• Have *little* labeled data + *lots* of unlabeled

• Each instance has two parts:
  \[
  x = [x_1, x_2]
  \]
  \(x_1, x_2\) conditionally independent given \(f(x)\)

• Each half can be used to classify instance
  \[
  \exists f_1, f_2 \text{ such that } f_1(x_1) \sim f_2(x_2) \sim f(x)
  \]

• Both \(f_1, f_2\) are learnable
  \[
  f_1 \in H_1, \quad f_2 \in H_2, \quad \exists \text{ learning algorithms } A_1, A_2
  \]
Co-training Example

Prof. Mausam
Students: Andrey,…
Projects: NLP, Prob. planning
I teach a class on Artificial intelligence

CSE 573: Artificial Intelligence
Course Description:…
Topics:…
Homework: …

Andrey
Classes taken:
1. Data mining
2. Artificial Intelligence
Research: Prob. planning
Without Co-training

A Few Labeled Instances

\[ \langle [x_1, x_2], f() \rangle \]

A_1 learns \( f_1 \) from \( x_1 \)

A_2 learns \( f_2 \) from \( x_2 \)

\( f_1(x_1) \sim f_2(x_2) \sim f(x) \)

Unlabeled Instances

Bad!! Not using Unlabeled Instances!

Combine with ensemble?
Co-training

A *Few Labeled Instances*

Unlabeled Instances

Lots of Labeled Instances

\[ f_1(x_1) \sim f_2(x_2) \sim f(x) \]

\[ A_1 \text{ learns } f_1 \text{ from } x_1 \]

\[ A_2 \text{ learns } f_2 \text{ from } x_2 \]

\[ <[x_1, x_2], f()> \]

\[ A \]

\[ f_1 \]

\[ A_2 \]

\[ f_2 \]
Observations

• Can apply $A_1$ to generate as much training data as one wants
  – If $x_1$ is conditionally independent of $x_2 / f(x)$,
  – then the error in the labels produced by $A_1$
  – *will look like random noise to $A_2$!!*

• Thus *no limit* to quality of the hypothesis $A_2$ can make
Co-training

Lots of Labeled Instances

<[^x_1, x_2], f()>  
\[x_1, x_2]\] 
\(f_2\) 
A_1 learns \(f_1\) from \(x_1\)

Hypothesis

Lots of Labeled Instances

<[^x_1, x_2], f_1(x_1)> 
\(f_1(x_1) \sim f_2(x_2) \sim f(x)\) 
A_2 learns \(f_2\) from \(x_2\)

Unlabeled Instances
It really works!

- Learning to classify web pages as course pages
  - $x_1 =$ bag of words on a page
  - $x_2 =$ bag of words from all anchors pointing to a page
- Naïve Bayes classifiers
  - 12 labeled pages
  - 1039 unlabeled

<table>
<thead>
<tr>
<th></th>
<th>Page-based classifier</th>
<th>Hyperlink-based classifier</th>
<th>Combined classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised training</td>
<td>12.9</td>
<td>12.4</td>
<td>11.1</td>
</tr>
<tr>
<td>Co-training</td>
<td>6.2</td>
<td>11.6</td>
<td>5.0</td>
</tr>
</tbody>
</table>

Table 2: Error rate in percent for classifying web pages as course home pages. The top row shows errors when training on only the labeled examples. Bottom row shows errors when co-training, using both labeled and unlabeled examples.