RANKING

David Kauchak
CS 451 – Fall 2013

Course feedback

Thanks!

Course feedback

Difficulty

0 1 2 3 4 5 6 7 8 9 10 11 12

1 2 3 4 5

Assignment 4

Assignment 5
Course feedback

Time/week

Experiments/follow-up questions

More interesting data sets

- large margin classifiers
- probabilistic modeling
- Unsupervised learning/clustering
- Ensemble learning
- Collaborative filtering
- MapReduce/Hadoop

Java tip for the day

```java
static

ArrayList<Example> data = dataset.getData();

// How can I iterate over it?
```

```java
// can do on anything that implements the Iterable interface
for( Example ex : data ){
}
```
### An aside: text classification

#### Raw data | labels
---|---
Chardonnay
Pinot Grigio
Zinfandel

#### Text: raw data

#### Raw data | labels | Features
---|---|---
Chardonnay
Pinot Grigio
Zinfandel

#### Feature examples

#### Raw data | labels | Features
---|---|---
Chardonnay
Pinot Grigio
Zinfandel

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

(1, 1, 0, 0, 1, 0, 0, …)

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

(4, 1, 1, 0, 0, 1, 0, 0, …)

Occurrence of words

Frequency of word occurrences

This is the representation we’re using for assignment 5
Each internal node represents whether or not the text has a particular word.

- **wheat** is a commodity that can be found in states across the nation.

- The US views technology as a commodity that it can export by the bushel.
Ranking problems

Suggest a simpler word for the word below:

vital

Suggest a simpler word for the word below:

acquired

Suggest a simpler word for the word below:

important
necessary
essential
needed
critical
crucial
mandatory
required
vital

Suggest a simpler word for the word below:

gotten
received
gained
obtained
got
purchased
bought
get hold of
acquired
Suggest a simpler word

- vital
- important
- necessary
- essential
- needed
- critical
- crucial
- mandatory
- required
- acquired
- gotten
- received
- gained
- obtained
- get
- purchased
- bought
- get hold of
- acquired
- purchased

list of synonyms ➔ ranker ➔ list ranked by simplicity

Ranking problems in general

- training data: a set of rankings where each ranking consists of a set of ranked examples
- training data: a set of rankings where each ranking consists of a set of ranked examples

Ranking problems in general

Real-world ranking problems?

Netflix My List
**Search**

- reranking N-best output lists
  - machine translation
  - computational biology
  - parsing
  - ...

**Ranking Applications**

- flight search
  - ...

**Black box approach to ranking**

Abstraction: we have a generic binary classifier, how can we use it to solve our new problem.

Can we solve our ranking problem with this?

**Predict better vs. worse**

Train a classifier to decide if the first input is better than second:
- Consider all possible pairings of the examples in a ranking
- Label as positive if the first example is higher ranked, negative otherwise

$\text{ranking} \in \{+1, -1\}$
Predict better vs. worse

Train a classifier to decide if the first input is better than second:
- Consider all possible pairings of the examples in a ranking
- Label as positive if the first example is higher ranked, negative otherwise

<table>
<thead>
<tr>
<th>ranking</th>
<th>new examples</th>
<th>binary label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>f_1, f_2, ..., f_n</td>
<td>+1</td>
</tr>
<tr>
<td></td>
<td>f_1, f_2, ..., f_n</td>
<td>+1</td>
</tr>
<tr>
<td></td>
<td>f_1, f_2, ..., f_n</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>f_1, f_2, ..., f_n</td>
<td>+1</td>
</tr>
<tr>
<td></td>
<td>f_1, f_2, ..., f_n</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>f_1, f_2, ..., f_n</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>f_1, f_2, ..., f_n</td>
<td>-1</td>
</tr>
</tbody>
</table>

How can we do this?
We want features that compare the two examples.

Combined feature vector

Many approaches! Will depend on domain and classifier

Two common approaches:
1. difference: $f'_i = a_i - b_i$
2. greater than/less than:

$$f'_i = \begin{cases} 
1 & \text{if } a_i > b_i \\
0 & \text{otherwise}
\end{cases}$$
Training

new examples
\[ f_1, f_2, \ldots, f_n \]
\[ f'_1, f'_2, \ldots, f'_n \]

label
\[ +1 \]
\[ +1 \]
\[ -1 \]
\[ +1 \]
\[ -1 \]

Testing

unranked
\[ f_1, f_2, \ldots, f_n \]
\[ f'_1, f'_2, \ldots, f'_n \]

extract features

binary classifier

ranking?
What is the ranking?
Algorithm?

for each binary example $e_{jk}$:
label[j] += $f_{jk}(e_{jk})$
lable[k] -= $f_{jk}(e_{jk})$
rank according to label scores

An improvement?

Weighted binary classification

Are these two examples the same?
Weighted binary classification

New examples | Weighted label
--- | ---
\( f_1, f_2, \ldots, f_n \) | +1
\( f_1, f_2, \ldots, f_n \) | +2
\( f_1, f_2, \ldots, f_n \) | -1
\( f_1, f_2, \ldots, f_n \) | -2
\( f_1, f_2, \ldots, f_n \) | -1

In general can weight with any consistent distance metric

Can we solve this problem?

Testing

If the classifier outputs a confidence, then we've learned a distance measure between examples

During testing we want to rank the examples based on the learned distance measure

Ideas?

Ranking evaluation

<table>
<thead>
<tr>
<th>ranking</th>
<th>prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Ideas?
Idea 1: accuracy

<table>
<thead>
<tr>
<th>ranking</th>
<th>prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>f₁, f₂, ..., fₙ</td>
<td>1, 1, 2, 3, 4, 5</td>
</tr>
</tbody>
</table>

Any problems with this?

Doesn’t capture “near” correct ranking

<table>
<thead>
<tr>
<th>ranking</th>
<th>prediction</th>
<th>prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>f₁, f₂, ..., fₙ</td>
<td>1, 1, 2, 3, 4, 5</td>
<td></td>
</tr>
</tbody>
</table>

1/5 = 0.2

Idea 2: correlation

<table>
<thead>
<tr>
<th>ranking</th>
<th>prediction</th>
<th>prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 4, 5</td>
<td>1, 3, 2, 5, 4</td>
<td></td>
</tr>
</tbody>
</table>

Look at the correlation between the ranking and the prediction