6.034 Introduction to Artificial Intelligence

Machine learning and applications
Problems we will cover

• Computational biology
  - cancer classification
  - functional classification of genes

• Information retrieval
  - document classification/ranking

• Recommender systems
  - predicting user preferences (e.g., movies)
Learning with documents

- Binary classification
  - e.g., filtering spam, retrieving similar documents from literature, web,...

- Multi-class classification
  - e.g., topic annotation

- Preference learning
  - e.g., rating reviews

- Learning relations
  - e.g., whether two documents are similar
Documents as feature vectors

• Does the word order matter?

White House officials consulted with the Justice Department in preparing a list of U.S. attorneys who would be removed.

(NYT 03/13/07)
Documents as feature vectors

• Does the word order matter?

White House officials consulted with the Justice Department in preparing a list of U.S. attorneys who would be removed. (NYT 03/13/07)
Does the word order matter?

- Doesn’t appear to matter to us...

(Wolf et al. 2006)
White House officials consulted with the Justice Department in preparing a list of U.S. attorneys who would be removed.

(NYT 03/13/07)
White House officials consulted with the Justice Department in preparing a list of U.S. attorneys who would be removed. (NYT 03/13/07)
White House officials consulted with the Justice Department in preparing a list of U.S. attorneys who would be removed. (NYT 03/13/07)
Document classification

- A few relevant documents (+1)
- A large number of irrelevant documents (-1)

\[
\phi(x) = \begin{bmatrix}
0 \\
1 \\
0 \\
0 \\
1 \\
\ldots
\end{bmatrix}
\]

politics
Justice
government
president
House

\[
\begin{pmatrix}
0 \\
1 \\
0 \\
0 \\
1 \\
\ldots
\end{pmatrix}
\]
Document classification

• Why is the problem challenging?
  - lots of possible words
  - only a small subset appears in any particular document
  - most frequent words are not content words
  - document classes typically tied to words that occur relatively infrequently
  - any two documents in the same class may have only a few content words in common
Some tricks

• We can transform the counts in the feature vectors so as to emphasize more “relevant” words

• TFIDF weighting

\[ \phi_w(x) = \frac{TF(w \text{ in doc. } x)}{\text{freq. of word } w \text{ in doc. } x} \cdot \log \left[ \frac{\# \text{ of docs with word } w}{\# \text{ of docs}} \right] \]
Learning with documents

- Binary classification
  - e.g., filtering spam, retrieving similar documents from literature, web,...

- Multi-class classification
  - e.g., topic annotation

- Preference learning
  - e.g., rating reviews

- Learning relations
  - e.g., whether two documents are similar

White House officials consulted with the Justice Department in preparing a list of U.S. attorneys who would be removed.
Multi-class classification

• We can turn a multi-class classification problem back into a binary problem by using multiple classifiers

  ![Diagram]

  - Classifier 1
  - Classifier 2
  - Politics
  - Religion
  - Computers

• There are many other ways of using binary classifiers to solve multi-class problems (cf. “output codes”)

```
Some multi-class results

• SVM performs well in comparison to other classifiers on document classification tasks

<table>
<thead>
<tr>
<th>20 Newsgroups</th>
<th>SVM</th>
<th>NB</th>
<th>SVM</th>
<th>NB</th>
<th>SVM</th>
<th>NB</th>
<th>SVM</th>
<th>NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVA</td>
<td>0.131</td>
<td>0.146</td>
<td>0.167</td>
<td>0.199</td>
<td>0.214</td>
<td>0.277</td>
<td>0.311</td>
<td>0.445</td>
</tr>
<tr>
<td>Dense 15</td>
<td>0.142</td>
<td>0.176</td>
<td>0.193</td>
<td>0.222</td>
<td>0.251</td>
<td>0.282</td>
<td>0.366</td>
<td>0.431</td>
</tr>
<tr>
<td>BCH 15</td>
<td>0.145</td>
<td>0.169</td>
<td>0.196</td>
<td>0.225</td>
<td>0.262</td>
<td>0.311</td>
<td>0.415</td>
<td>0.520</td>
</tr>
<tr>
<td>Dense 31</td>
<td>0.135</td>
<td>0.168</td>
<td>0.180</td>
<td>0.214</td>
<td>0.233</td>
<td>0.276</td>
<td>0.348</td>
<td>0.428</td>
</tr>
<tr>
<td>BCH 31</td>
<td>0.131</td>
<td>0.153</td>
<td>0.173</td>
<td>0.198</td>
<td>0.224</td>
<td>0.259</td>
<td>0.333</td>
<td>0.438</td>
</tr>
<tr>
<td>Dense 63</td>
<td>0.129</td>
<td>0.154</td>
<td>0.171</td>
<td>0.198</td>
<td>0.222</td>
<td>0.256</td>
<td>0.326</td>
<td>0.407</td>
</tr>
<tr>
<td>BCH 63</td>
<td>0.125</td>
<td>0.145</td>
<td>0.164</td>
<td>0.188</td>
<td>0.213</td>
<td>0.245</td>
<td>0.312</td>
<td>0.390</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Industry Sector</th>
<th>SVM</th>
<th>NB</th>
<th>SVM</th>
<th>NB</th>
<th>SVM</th>
<th>NB</th>
<th>SVM</th>
<th>NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVA</td>
<td>0.072</td>
<td>0.357</td>
<td>0.176</td>
<td>0.568</td>
<td>0.341</td>
<td>0.725</td>
<td>0.650</td>
<td>0.885</td>
</tr>
<tr>
<td>Dense 15</td>
<td>0.119</td>
<td>0.191</td>
<td>0.283</td>
<td>0.363</td>
<td>0.461</td>
<td>0.542</td>
<td>0.738</td>
<td>0.805</td>
</tr>
<tr>
<td>BCH 15</td>
<td>0.106</td>
<td>0.182</td>
<td>0.261</td>
<td>0.352</td>
<td>0.438</td>
<td>0.518</td>
<td>0.717</td>
<td>0.771</td>
</tr>
<tr>
<td>Dense 31</td>
<td>0.083</td>
<td>0.145</td>
<td>0.216</td>
<td>0.301</td>
<td>0.394</td>
<td>0.482</td>
<td>0.701</td>
<td>0.769</td>
</tr>
<tr>
<td>BCH 31</td>
<td>0.076</td>
<td>0.140</td>
<td>0.198</td>
<td>0.292</td>
<td>0.371</td>
<td>0.462</td>
<td>0.676</td>
<td>0.743</td>
</tr>
<tr>
<td>Dense 63</td>
<td>0.072</td>
<td>0.135</td>
<td>0.189</td>
<td>0.279</td>
<td>0.363</td>
<td>0.453</td>
<td>0.674</td>
<td>0.745</td>
</tr>
<tr>
<td>BCH 63</td>
<td>0.067</td>
<td>0.128</td>
<td>0.176</td>
<td>0.272</td>
<td>0.343</td>
<td>0.443</td>
<td>0.653</td>
<td>0.734</td>
</tr>
</tbody>
</table>
Learning with documents

- Binary classification
  - e.g., filtering spam, retrieving similar documents from literature, web,...

- Multi-class classification
  - e.g., topic annotation

- Preference learning
  - e.g., rating reviews

- Learning relations
  - e.g., whether two documents are similar

White House officials consulted with the Justice Department in preparing a list of U.S. attorneys who would be removed.
Preference learning

• Predicting ratings on a 1-5 (star) scale is NOT a multi-class classification problem but a problem known as ordinal regression
  - class labels are symbols:
    - it doesn’t make sense to compare the “magnitude” of label ‘1’ with label ‘2’ since they are just different symbols associated with the classes
  - 1-5 rating scale is ordinal
    - comparison is relevant: 1 < 2 < 3 < 4 < 5
Preference learning

\[ \hat{y} = \text{sign}\left( \sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + w_0 \right) \]

binary classification

\[ f(x) \]

\[ b = 0 \]

-1 \quad +1
Preference learning

\[ \hat{y} = \text{sign}\left( \sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + w_0 \right) \]

- **Binary classification**
  - \(-1\) and \(+1\)
  - \( b = 0 \)

- **Ordinal regression**
  - \( b_1, b_2, b_3, b_4 \)

 discriminant function \( f(x) \)
Preference learning as classification

- We can turn the ordinal regression problem into a classification problem by associating two classification constraints with each response.

- For example, if the response for $x$ should be ‘***’, then we would try to enforce:

  $$+1 \cdot (f(x) - b_2) \geq 1 \quad -1 \cdot (f(x) - b_3) \geq 1$$

- We will have to learn $b$’s in addition to $f(x)$. 

Learning similarities

• If we assume that the documents are either similar or not, then we need a classifier over pairs of documents.

• The training set would consist of labeled pairs of documents:

\[
\begin{align*}
&\text{pair of documents} \\
&\quad \left( x_i, \ldots, x_j \right) \\
&\quad \text{label} \\
&\quad -1
\end{align*}
\]

White House officials consulted with the Justice Department in preparing a list of U.S. attorneys who would be removed.

Prime Minister Nuri Kamal al-Maliki met with tribal leaders and the provincial governor of Ramadi.

• Most pairs can be assumed to be dissimilar (cf. imbalanced classes).
A classifier over pairs of examples

- Can we still use a linear classifier? If yes, we need a feature vector.
- We could simply use a feature vector for each pair by concatenating the individual feature vectors.

\[ \tilde{\phi}(x_i, x_j) = \begin{bmatrix} \phi(x_i) \\ \phi(x_j) \end{bmatrix} \]

- this is not the best way to solve the problem.
Problems we will cover

• Computational biology
  - cancer classification
  - functional classification of genes

• Information retrieval
  - document classification/ranking

• Recommender systems
  - predicting user preferences (e.g., movies)
### Recommender problems

- **Lot’s of companies rely on predicting user preferences**

  
  ![Rating Matrix](image)

  - The machine learning task here is to fill the entries of the rating matrix based on only a few ratings from each user
Recommender problems: movies

• The dimensions in practice
  - 400,000 users
  - 17,000 movies
  - 1% known ratings
Recommender problems: movies

• Users rate movies very differently
  - very positive, very negative, etc.

• Missing ratings are ambiguous
  - movies that users have not yet seen
  - movies they saw but didn’t rate
  - movies they know they don’t want to see
  - movies they don’t realize they would like to see
Where’s the information?

• The problem is actually a bit easier than what it seems
  - many users are similar, many movies are similar
How do we solve this with SVMs?

- Suppose we were able to obtain a feature vector for each movie

\[ \phi_1, \ldots, \phi_m \]

- We could then try to solve the problem separately for each user
The problem for predicting the ratings of each user becomes an ordinal regression problem exactly of the type we have already discussed.
Ordinal regression: movies

• Suppose we have obtained feature vectors for each user
• The problem is again ordinal regression

\[ f = \phi^T \theta \]
Solution

- We can iteratively solve feature vectors for users and movies

\[
f_{ij} = \theta_i^T \phi_j
\]
Some results

- The SVM based collaborative filtering method works very well in comparison to other methods

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MovieLens Weak NMAE</th>
<th>MovieLens Strong NMAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>URP</td>
<td>.4341 ± .0023</td>
<td>.4444 ± .0032</td>
</tr>
<tr>
<td>Attitude</td>
<td>.4320 ± .0055</td>
<td>.4375 ± .0028</td>
</tr>
<tr>
<td>MMMF</td>
<td>.4156 ± .0037</td>
<td>.4203 ± .0138</td>
</tr>
</tbody>
</table>

(Rennie et al. 2005)
Problems we will cover

• Computational biology
  - cancer classification
  - functional classification of genes

• Information retrieval
  - document classification/ranking

• Recommender systems
  - predicting user preferences (e.g., movies)

Lot’s of other problems can also be solved with SVMs...