Text Classification & Linear Models

CMSC 723 / LING 723 / INST 725

Marine Carpuat

Slides credit: Dan Jurafsky & James Martin, Jacob Eisenstein
Logistics/Reminders

• Homework 1 – due Thursday Sep 7 by 12pm.

• Project 1 coming up

• Thursday lecture time: project set-up office hour in CSIC 1121
Recap: Word Meaning

2 core issues from an NLP perspective

• **Semantic similarity**: given two words, how similar are they in meaning?

• Key concepts: vector semantics, PPMI and its variants, cosine similarity

• **Word sense disambiguation**: given a word that has more than one meaning, which one is used in a specific context?

• Key concepts: word sense, WordNet and sense inventories, unsupervised disambiguation (Lesk), supervised disambiguation
Today

- Text classification problems
  - and their evaluation
- Linear classifiers
  - Features & Weights
  - Bag of words
  - Naïve Bayes
Text classification
Is this spam?

From: "Fabian Starr"
From: <Patrick_Freeman@pamietaniepeerelu.pl>
Subject: Hey! Sofware for the funny prices!

Get the great discounts on popular software today for PC and Macintosh
http://iiled.org/Cj4Lmx

70-90% Discounts from retail price!!!
All sofware is instantly available to download - No Need Wait!
What is the subject of this article?

**MeSH Subject Category Hierarchy**

- Antagonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- ...
Text Classification

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language Identification
- Sentiment analysis
- ...
Text Classification: definition

• **Input:**
  • a document $d$
  • a fixed set of classes $Y = \{ y_1, y_2, \ldots, y_J \}$

• **Output:** a predicted class $y \in Y$
Classification Methods:
Hand-coded rules

• Rules based on combinations of words or other features
  • spam: black-list-address OR (“dollars” AND “have been selected”)

• Accuracy can be high
  • If rules carefully refined by expert

• But building and maintaining these rules is expensive
Classification Methods:
Supervised Machine Learning

• **Input**
  • a document \(d\)
  • a fixed set of classes \(Y = \{y_1, y_2, \ldots, y_J\}\)
  • a training set of \(m\) hand-labeled documents \((d_1, y_1), \ldots, (d_m, y_m)\)

• **Output**
  • a learned classifier \(d \rightarrow y\)
Aside: getting examples for supervised learning

- Human annotation
  - By experts or non-experts (crowdsourcing)
  - Found data

- How do we know how good a classifier is?
  - Compare classifier predictions with human annotation
  - On held out test examples
  - Evaluation metrics: accuracy, precision, recall
The 2-by-2 contingency table

<table>
<thead>
<tr>
<th></th>
<th>correct</th>
<th>not correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>selected</td>
<td>tp</td>
<td>fp</td>
</tr>
<tr>
<td>not selected</td>
<td>fn</td>
<td>tn</td>
</tr>
</tbody>
</table>
Precision and recall

• **Precision**: % of selected items that are correct
  **Recall**: % of correct items that are selected

<table>
<thead>
<tr>
<th></th>
<th>correct</th>
<th>not correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>selected</td>
<td>tp</td>
<td>fp</td>
</tr>
<tr>
<td>not selected</td>
<td>fn</td>
<td>tn</td>
</tr>
</tbody>
</table>
A combined measure: F

• A combined measure that assesses the P/R tradeoff is F measure (weighted harmonic mean):

$$ F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} $$

• People usually use balanced F1 measure
  • i.e., with $\beta = 1$ (that is, $\alpha = \frac{1}{2}$):

$$ F = \frac{2PR}{P + R} $$
Linear Classifiers
Bag of words

\[ \mathbf{w}_1 = \{ \text{great, sunset, tonight, ...} \} \quad \mathbf{w}_2 = \{ \text{ugly, skies, buford, ...} \} \]

\[
\begin{array}{cccccccccccc}
\text{aardvark} & \text{abacus} & \ldots & \text{behind} & \ldots & \text{buford} & \ldots & \text{clouds} & \ldots & \text{great} & \ldots & \text{ugly} \\
\mathbf{x}_1^T &=& 0 & 0 & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\
\mathbf{x}_2^T &=& 0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & 1 & 0 & .
\end{array}
\]

\[ \mathbf{x}_1 = \{ \text{great : 1, sunset : 1, tonight : 1, ...} \} \]
\[ \mathbf{x}_2 = \{ \text{ugly : 1, skies : 1, buford : 1, ...} \} \]
Defining features

Suppose $y \in \mathcal{Y} = \{\text{pos, neg, neut}\}$. Then,

$$\begin{align*}
\mathbf{f}(\mathbf{x}, y = \text{pos}) &= [\mathbf{x}^T, \mathbf{0}^T, \mathbf{0}^T, 1]^T \\
\mathbf{f}(\mathbf{x}, y = \text{neg}) &= [\mathbf{0}^T, \mathbf{x}^T, \mathbf{0}^T, 1]^T \\
\mathbf{f}(\mathbf{x}, y = \text{neut}) &= [\mathbf{0}^T, \mathbf{0}^T, \mathbf{x}^T, 1]^T
\end{align*}$$
Defining features

Suppose \( y \in \mathcal{Y} = \{\text{pos, neg, neut}\} \). Then,

\[
\begin{align*}
f(x, y = \text{pos}) &= [x^T, 0^T, 0^T, 1]^T \\
f(x, y = \text{neg}) &= [0^T, x^T, 0^T, 1]^T \\
f(x, y = \text{neut}) &= [0^T, 0^T, x^T, 1]^T
\end{align*}
\]

The feature vector is composed of individual feature functions, e.g.:

\[
\begin{align*}
f_{176}(x, y) &= x_{176} \times \delta(y = \text{pos}) \\
&= \delta(\text{great } \in w \land y = \text{pos}) \\
f_{177}(x, y) &= x_{177} \times \delta(y = \text{pos}) \\
f_{10176}(x, y) &= x_{176} \times \delta(y = \text{neg}) \ldots
\end{align*}
\]

We usually add an “offset” feature at the end of each vector.
Linear classification

We can then define weights for each feature:

$$\theta = \{ \langle \text{great}, \text{pos} \rangle = 1, \langle \text{great}, \text{neg} \rangle = -1, \langle \text{great}, \text{neut} \rangle = 0, $$
$$\langle \text{ugly}, \text{pos} \rangle = -1, \langle \text{ugly}, \text{neg} \rangle = 1, \langle \text{ugly}, \text{neut} \rangle = 0, $$
$$\langle \text{buford}, \text{pos} \rangle = 0, \langle \text{buford}, \text{neg} \rangle = 0, \langle \text{buford}, \text{neut} \rangle = 0, $$
$$\ldots \}$$

We can arrange these weights into a vector.

The score for any instance and label is equal to the sum of the weights for all features in the instance:

$$\psi_{y,x} = \sum_{n} \theta_n f_n(x, y)$$

$$= \theta^T f(x, y)$$

$$\hat{y} = \arg \max_{y} \theta^T f(x, y)$$
Linear Models for Classification

\[ \hat{y} = \arg \max_y \theta^T f(x, y) \]

Weights

Feature function representation
How can we learn weights?

• By hand

• Probability
  • e.g., Naïve Bayes

• Discriminative training
  • e.g., perceptron, support vector machines
Generative Story for Multinomial Naïve Bayes

- A hypothetical stochastic process describing how training examples are generated

For each document $i$,

- draw the label $y_i \sim \text{Categorical}(\mu)$
- draw the vector of counts $x_i \sim \text{Multinomial}(\phi_{y_i})$.

$$p_{\text{mult}}(\mathbf{x}; \phi) = \frac{\left(\sum_j x_j\right)!}{\prod_j x_j!} \prod_j \phi_j^{x_j}$$
Prediction with Naïve Bayes

\[ \text{Score}(x,y) := \log P(x, y; \phi, \mu) \]
\[ = \log P(x|y; \phi) P(y; \mu) \]
\[ = \log P(x|y; \phi) + \log P(y; \mu) \]
Prediction with Naïve Bayes

\[ \text{Score}(x,y) := \log P(x, y; \phi, \mu) \]
\[ = \log P(x|y; \phi) P(y; \mu) \]
\[ = \log P(x|y; \phi) + \log P(y; \mu) \]
\[ = \log \text{Multinomial}(x; \phi_y) + \log \text{Cat}(y; \mu) \]
\[ = \log \left( \frac{\prod_n x_n!}{\prod_n x_n!} \right) + \log \prod_n \phi_{y,n}^{x_n} + \log \mu_y \]
\[ \propto \sum_n x_n \log \phi_{y,n} + \log \mu_y \]
\[ = \theta^T f(x, y) \]

where
\[ \theta = [\log \phi_1^T, \log \mu_1, \log \phi_2^T, \log \mu_2, \ldots]^T \]
\[ f(x, y) = [0, \ldots, 0, x^T, 1, 0, \ldots, 0]^T \]
Parameter Estimation

• “count and normalize”

• Parameters of a multinomial distribution

\[
\phi_{y,j} = \frac{\sum_{i: Y_i = y} x_{i,j}}{\sum_{j'} \sum_{i: Y_i = y} x_{i,j'}} = \frac{\text{count}(y, j)}{\sum_{j'} \text{count}(y, j')}
\]

• Relative frequency estimator

• Formally: this is the maximum likelihood estimate
  • See CIML for derivation
Smoothing (add alpha / Laplace)

\[
\phi_{y,j} = \frac{\alpha + \sum_{i:Y_i=y} x_{i,j}}{\sum_{j'=1}^{V} \left( \alpha + \sum_{i:Y_i=y} x_{i,j'} \right)} = \frac{\alpha + \text{count}(y,j)}{V \alpha + \sum_{j'=1}^{V} \text{count}(y,j')}
\]
Naïve Bayes recap

- Define $p(x, y)$ via a generative model
- Prediction: $\hat{y} = \arg \max_y p(x_i, y)$
- Learning:

$$\theta = \arg \max_{\theta} p(x, y; \theta)$$

$$p(x, y; \theta) = \prod_i p(x_i, y_i; \theta) = \prod_i p(x_i | y_i)p(y_i)$$

$$\phi_{y,j} = \frac{\sum_{i:Y_i=y} x_{ij}}{\sum_{i:Y_i=y} \sum_j x_{ij}}$$

$$\mu_y = \frac{\text{count}(Y = y)}{N}$$

This gives the maximum likelihood estimator (MLE; same as relative frequency estimator)
Today

• Text classification problems
  • and their evaluation

• Linear classifiers
  • Features & Weights
  • Bag of words
  • Naïve Bayes