Text Classification Contd + Document Representations

Prof. Sameer Singh

CS 295: STATISTICAL NLP
WINTER 2017

January 17, 2017

Based on slides from Nathan Schneider, Noah Smith, Dan Klein and everyone else they copied from.
Outline

- Logistic Regression
- Brief Intro to Neural Networks
- Document Representations
Outline

- Logistic Regression
- Brief Intro to Neural Networks
- Document Representations
Text Classification

Human machine interface for ABC computer applications

- Human Computer Interaction
  - Theory
  - Artificial Intelligence
  - Systems
Linear Models

Human machine interface for ABC computer applications

\[ f(x) = \text{argmax}_y \bar{w} \cdot \phi(x, y) \]

\[ y \in \{HCl, AI, Th, Sys\} \]

\[ \text{(v+1)} \]

\[ \omega \]

\[ \phi(x, y) \]

\[ o \]
Matrix/Neural View

\[ \begin{align*}
\phi(x) \times V' \times W' \times L = & W \times L \\
\phi(x, y) = & W^T \cdot \phi(x, y) \\
\phi(x) \times V' \times 3 \times W \times L = & y^* 
\end{align*} \]
Naïve Bayes as a Linear Model

\[
f_{NB}(x) = \arg \max_y P(x, y) = \arg \max_y P(y) \prod_i P(x_i | y) \\
= \arg \max_y \log P(y) + \sum_i \log P(x_i | y)
\]
Joint vs Conditional Likelihood

\[ f_{\text{cond}}(x) = \arg \max_y p(y|x) \]

\[ \Theta^*_{\text{cond}} = \arg \max_{ \theta } \prod_{x \in D} P_{\theta}(y_{a|x} \mid x_a) \]
Logistic Regression Model

\[
f(x) = \arg\max_y \ w \cdot \phi(x, y)
\]

\[
p(y|x) = \frac{e^{w \cdot \phi(x, y)}}{\sum_y e^{w \cdot \phi(x, y)}}
\]

\[
f(x) = \arg\max_y p(y|x)
\]

\[
= \arg\max_y \frac{e^{w \cdot \phi(x, y)}}{\sum_y e^{w \cdot \phi(x, y)}}
\]

\[
= \arg\max_y w \cdot \phi(x, y)
\]
Logistic Regression: 2 classes

\[ p(y=1 | x) = \frac{e^{w \cdot \phi(x)}}{1 + e^{w \cdot \phi(x)}} \]

\( \phi_k(x) = \text{# word } k \)

"interface" \( W_{\text{interface}} > 0 \)

"tree" \( W_{\text{tree}} < 0 \)
Estimating the parameters, \( \omega \)

\[
\omega^* = \arg\max_{\omega} \prod_{a} P(y_a | x_a) \quad e^{w \cdot \phi(x_a, y_a)} - \log \sum_{y} e^{w \cdot \phi(x_a, y)} \\
= \arg\max_{\omega} \sum_{a} \log P(y_a | x_a) \quad e^{w \cdot \phi(x_a, y_a)} - \log \sum_{y} e^{w \cdot \phi(x_a, y)} \\
= \arg\max_{\omega} \sum_{a} w \cdot \phi(x_a, y_a) - \log \sum_{y} e^{w \cdot \phi(x_a, y)} \\
= \arg\min_{\omega} - \sum_{a} w \cdot \phi(x_a, y_a) + \log \sum_{y} e^{w \cdot \phi(x_a, y)}
\]
Gradient Descent

$$\text{argmin} - \frac{1}{n} \sum \omega_i \phi(x_a,y_a) - \log \frac{1}{y} \frac{e^{-z}}{z}$$

$$\frac{\delta L}{\delta \omega_i} = \frac{1}{n} \sum \phi_i(x_a,y_a) - E_{p(y|x)} [\phi(x_a,y)]$$
Sparsity of Words

- Remember Zipf’s Law? Lots of rare words
- For classification, they can be more informative!

\[
\phi(x_i) = \text{tf}(i,d) \text{idf}(i)
\]

\[
\text{idf}(i) = - \log P(i) = - \log \frac{D_i}{D} = \log \frac{D}{D_i}
\]

\[
\text{term freq (TF)}
\]

\[
D_i = \# \text{docs that } i \text{ appears in}
\]

\[
D = \# \text{docs}
\]

\[
\text{Nid} = \# \text{word } i \text{ in doc } d
\]
Why use log(proportion)

• It works...

• Importance is not a linear function

• IDF is an additive function

\[
\text{idf}(w_1, w_2) = -\log P(w_1, w_2) = -\log P(w_1)P(w_2)
\]

\[
d_{12} = -\log P(w_1) - \log P(w_2) = \text{idf}(w_1) + \text{idf}(w_2)
\]
Tips and Tricks: Regularization

**Overfitting**

- Training data is finite: thus has spurious correlations
- Rare words that occur with one label!
  - Or don’t occur often enough
  - Curse of the Zipf’s Law continues

For a word that occurs 10 times...

\[ p(y_1 | w) = 0.5 \]
\[ p(y_2 | w) = 0.5 \]
\[ p(\#w = y_1 \geq 7) \approx 0.2 \]

There are many that occur \(~10\) times!
Tips and Tricks: Regularization

Fixing Overfitting

- Ignore rare words (opposite of TF-IDF)
- Penalize really high weights...

\[ w = \arg\min_w \frac{L}{2} + \lambda \|w\|_2^2 \]

\[ \frac{\partial L}{\partial w_i} + 2\lambda w_i \]

\[ \text{Reg. Strength} \]

\[ \lambda \]

\[ \text{Train} \]

\[ \text{Test dev validation} \]

\[ D_i < 10 \]
Tips and Tricks: Featurizing

\[ \phi(x) \]

- \( x_i \)
- # words
- Avg length of word
- # unique words

ML

Sentiment analyzer
Neural View of Log. Regression
Linear vs Non-linear Model
Introducing a Hidden Layer

\[ h = a(\mathbf{W}_1 \phi(x)) \]

\[ y = \mathbf{W}_2 h \]

\[ y^* = \arg\max_y \]
What is Deep Learning?

Many hidden layers

In NLP, utilize unlabeled data to learn representations... (next lecture)
Outline

Logistic Regression

Brief Intro to Neural Networks

Document Representations
Document Similarity

A survey of user opinion of computer system response time

Relation of user perceived response time to error measurement

The generation of random, binary, ordered trees
Advantages

- Between -1 and 1 (0 means no overlap)
- If all >0, it is between 0 and 1
- Size of vectors don’t matter

\[
\text{dist}(x,y) = \cos \theta = \frac{x_1 \cdot x_2}{\|x_1\| \|x_2\|} = \frac{\sum_{k=1}^{K} x_{1k} \cdot x_{2k}}{\sqrt{\sum_{k=1}^{K} x_{1k}^2} \sqrt{\sum_{k=1}^{K} x_{2k}^2}}
\]
Term Document Matrix

$X_{v \times n}$

$X_{ij} = \# \text{word}_i \text{ in doc}_j$

Sparse!
Local and Global Weighting

\[ x_{ij} = l_{ij} g_i \]

Local Weighting

- Binary: \( 1 \)
- Term Freq:
- Log:
  \[ \log(1 + \#_i) \]

Global Weighting

- Binary: \( 1 \)
- Normal:
  \[ \frac{1}{D_i} \]
- IDF:
  \[ \text{idf}(i) \]
Example: Documents

- **c1**: Human machine interface for ABC computer applications
- **c2**: A survey of user opinion of computer system response time
- **c3**: The EPS user interface management system
- **c4**: System and human system engineering testing of EPS
- **c5**: Relation of user perceived response time to error measurement

- **m1**: The generation of random, binary, ordered trees
- **m2**: The intersection graph of paths in trees
- **m3**: Graph minors IV: Widths of trees and well-quasi-ordering
- **m4**: Graph minors: A survey

From http://lsa.colorado.edu/papers/dp1.LSAintro.pdf
Example: Term-Doc Matrix

c1 c2 c3 c4 c5 m1 m2 m3 m4

human
interface
computer
user
system
response
time
EPS
survey
trees
graph
minors
Example: Distance Matrix
Problems with Sparse Vectors

c2: A survey of user opinion of computer system response time

m4: Graph minors: A survey

c1: Human machine interface for ABC computer applications
Example: Distance Matrix
Option 1: Clustering
Example: Clustering

\[ \Phi \alpha \]

\[ \begin{array}{cccccc}
  c1 & c2 & c3 & c4 & c5 \\
  m1 & m2 & m3 & m4 \\
\end{array} \]

\[ k=2 \]
Upcoming...

Homework

• Homework 1 is up!
• No more material will be covered
• Due: January 26, 2017

Project

• Project pitch is due January 23, 2017!
• Start assembling teams now
• Tons of datasets on the “projects” page on website