NLP
Introduction to NLP

Generative vs. Discriminative Models
Generative vs. Discriminative

• Generative
  - Learn a model of the joint probability $p(d, c)$
  - Use Bayes’ Rule to calculate $p(c|d)$
  - Build a model of each class; given example, return the model most likely to have generated that example
  - Examples: Naïve Bayes, Gaussian Discriminant Analysis

• Discriminative
  - Model posterior probability $p(c|d)$ directly
  - Class is a function of document vector
  - Find the exact function that minimizes classification errors on the training data
  - Examples: Logistic regression, Neural Networks (NNs), Support Vector Machines (SVMs)
Assumptions of Discriminative Classifiers

• Data examples (documents) are represented as vectors of features (words, phrases, ngrams, etc)
• Looking for a function that maps each vector into a class.
• This function can be found by minimizing the errors on the training data (plus other various criteria)
• Different classifiers vary on what the function looks like, and how they find the function
Discriminative vs. Generative Classifiers

- Discriminative classifiers are generally more effective, since they directly optimize the classification accuracy. But
  - They are all sensitive to the choice of features, and so far these features are extracted heuristically
  - Also, overfitting can happen if data is sparse
- Generative classifiers are the “opposite”
  - They directly model text, an unnecessarily harder problem than classification
  - They can easily exploit unlabeled data
Introduction to NLP

Generative Classifier: Naïve Bayes
Naïve Bayes Intuition

• Simple ("naïve") classification method based on Bayes rule

• Relies on very simple representation of document
  – Bag of words
I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!
Remember Bayes’ Rule?

Bayes’ Rule:

\[
P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}
\]

- \(d\) is the document (represented as a list of features of a document, \(x_1, \ldots, x_n\))
- \(c\) is a class (e.g., “not spam”)
Naïve Bayes Classifier (I)

$$c_{MAP} = \arg\max_{c \in C} P(c | d)$$

$$= \arg\max_{c \in C} \frac{P(d | c)P(c)}{P(d)}$$

MAP is “maximum a posteriori” = most likely class

Bayes Rule

Dropping the denominator
Naïve Bayes Classifier (II)

\[ c_{MAP} = \arg \max_{c \in C} P(d \mid c) P(c) \]

But where will we get these probabilities?
Naïve Bayesian classifiers

- Naïve Bayesian classifier
  
  \[ P(d \in C \mid F_1, F_2, \ldots F_k) = \frac{P(F_1, F_2, \ldots F_k \mid d \in C)P(d \in C)}{P(F_1, F_2, \ldots F_k)} \]

- Assuming statistical independence
  
  \[ P(d \in C \mid F_1, F_2, \ldots F_k) = \frac{\prod_{j=1}^{k} P(F_j \mid d \in C)P(d \in C)}{\prod_{j=1}^{k} P(F_j)} \]

- Features = words (or phrases) typically
Multinomial Naïve Bayes
Independence Assumptions

• Bag of Words assumption
  – Assume position doesn’t matter

• Conditional Independence
  – Assume the feature probabilities $P(x_i|c)$ are independent given the class $c$.

[Jurafsky and Martin]
Multinomial Naïve Bayes Classifier

\[ c_{NB} = \arg \max_{c \in C} P(c_j) \prod_{x \in X} P(x | c) \]

This is why it’s naïve!

[Jurafsky and Martin]
Learning the Multinomial Naïve Bayes Model

• First attempt: maximum likelihood estimates
  – simply use the frequencies in the data

\[
\hat{P}(c_j) = \frac{\text{doccount}(C = c_j)}{N_{\text{doc}}}
\]

\[
\hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}
\]

[Jurafsky and Martin]
Parameter Estimation

\[
\hat{P}(w_i \mid c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}
\]

fraction of times word \( w_i \) appears among all words in documents of topic \( c_j \)

- Create mega-document for topic \( j \) by concatenating all docs in this topic
  - Use frequency of \( w \) in mega-document

[Jurafsky and Martin]
Problem with Maximum Likelihood

• What if we have seen no training documents with the word *fantastic* and classified in the topic *positive* (*thumbs-up*)?

• Zero probabilities cannot be conditioned away, no matter the other evidence!

[Jurafsky and Martin]
Laplace Smoothing

\[
\hat{P}(w_i | c) = \frac{\text{count}(w_i, c)}{\sum_{w \in V} (\text{count}(w, c))}
\]

\[
\hat{P}(w_i | c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)}
\]

\[
= \frac{\text{count}(w_i, c) + 1}{\left( \sum_{w \in V} \text{count}(w, c) \right) + |V|}
\]

[Jurafsky and Martin]
Multinomial Naïve Bayes: Learning

- From training corpus, extract *Vocabulary*

- Calculate $P(c_j)$ terms
  - For each $c_j$ in $C$ do
    \[
    docs_j \leftarrow \text{all docs with class } c_j
    \]
  \[
  P(c_j) \leftarrow \frac{|docs_j|}{|\text{total # documents}|}
  \]

- Calculate $P(w_k \mid c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - For each word $w_k$ in *Vocabulary*
    \[
    n_k \leftarrow \text{# of occurrences of } w_k \text{ in } Text_j
    \]
  \[
  P(w_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |\text{Vocabulary}|}
  \]

[Jurafsky and Martin]
Example

- **Features** = \{I hate love this book\}
- **Training**
  - I hate this book
  - Love this book
- **What is** \( P(Y|X) \)?
- **Prior** \( p(Y) \)
- **Testing**
  - hate book
- **Different conditions**
  - \( a = 0 \) (no smoothing)
  - \( a = 1 \) (smoothing)

\[
P(Y) = [1/2 \quad 1/2]
\]
\[
M = \begin{bmatrix}
1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1
\end{bmatrix}
\]
\[
P(X|Y) = \begin{bmatrix}
1/4 & 1/4 & 0 & 1/4 & 1/4 \\
0 & 0 & 1/3 & 1/3 & 1/3
\end{bmatrix}
\]
\[
P(Y|X) \propto [1/2 \times 1/4 \times 1/4 \quad 1/2 \times 0 \times 1/3] = [1 \quad 0]
\]
\[
P(X|Y) = \begin{bmatrix}
2/9 & 2/9 & 1/9 & 2/9 & 2/9 \\
1/8 & 1/8 & 2/8 & 2/8 & 2/8
\end{bmatrix}
\]
\[
P(Y|X) \propto [1/2 \times 2/9 \times 2/9 \quad 1/2 \times 1/8 \times 2/8] = [0.613 \quad 0.387]
\]
Example

\[ P(Y) = \begin{bmatrix} 1/2 & 1/2 \end{bmatrix} \]

\[ P(X|Y) = \begin{bmatrix} 2/9 & 2/9 & 1/9 & 2/9 & 2/9 \\ 1/8 & 1/8 & 2/8 & 2/8 & 2/8 \end{bmatrix} \]

\[ P(Y|X) \propto [1/2 \times 2/9 \times 2/9 \quad 1/2 \times 1/8 \times 2/8] = [0.613 \ 0.387] \]
Ways Naive Bayes Is Not So Naive

• Very fast, low storage requirements
• Robust to irrelevant features
• Irrelevant features cancel each other without affecting results
• Very good in domains with many equally important features
  – Decision trees suffer from *fragmentation* in such cases – especially if little data
• Optimal if the independence assumptions hold
  – If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
• A good, dependable baseline for text classification
  – But other classifiers give better accuracy

[Jurafsky and Martin]
NLP