Lecture 6: text classification, feature selection
Topics

• Text classification example
  – model specification
  – model estimation with regularization

• Feature selection
  – filter methods
  – wrapper methods
Example problem

- Text classification (information retrieval)
  - a large number of documents $x$ in a database
  - only a few labeled documents $\{(x_1, y_1), \ldots, (x_n, y_n)\}$

- We wish to build a classifier on the basis of the few labeled training examples (documents).
  - we assume that the labels are binary (1/0)

- Several steps:
  1. Feature transformation (why?)
  2. Model/classifier specification
  3. Model/classifier estimation with regularization
Feature transformation

• The presence/absence of specific words in a document carries information about what the document is about

• We can construct $m$ (about 10,000) indicator features $\{\phi_k(x)\}$ for whether a word appears in the document

  $\phi_k(x) = 1$, if word $k$ appears in document $x$; zero otherwise

  $\Phi(x) = [\phi_1(x), \ldots, \phi_m(x)]^T$ is the resulting feature vector

• Are there better features?
Model specification: “Naive Bayes” model

- We can treat each word detector $\phi_i(x)$ as an independent expert.
- We combine these “expert opinions” by modeling their decisions given the labels:

$$P(\Phi(x)|y, \theta) = \prod_{k=1}^{m} P(\phi_k(x)|y, \theta_k)$$

where $P(\phi_k(x)|y, \theta_k)$ is the conditional probability that the $k^{th}$ word appears in a document labeled $y$. $\theta_k$ are the parameters associated with this conditional probability.

- Classification via Bayes rule:

$$P(y|\Phi(x), \theta) = \frac{P(\Phi(x)|y, \theta)P(y)}{\sum_{y'=0,1} P(\Phi(x)|y', \theta)P(y')}$$
Naive Bayes estimation

- We can write the conditional probabilities of a single feature as
  \[ P(\phi_k(x)|y, \theta_k) = \theta_{k|y}^{\phi_k(x)} (1 - \theta_{k|y})^{1-\phi_k(x)} \]
  where \( \theta_{k|y} \) is the probability that the word \( k \) appears in a document labeled \( y \) and \( \theta_k = \{\theta_{k|1}, \theta_{k|0}\} \).

- Maximum likelihood estimation (here for a single feature)
  \[
  J_n(\theta_k) = \sum_{i=1}^{n} \log P(\phi_k(x_i)|y_i, \theta_k) \\
  = \sum_{i=1}^{n} \left[ \phi_k(x_i) \log(\theta_{k|y_i}) + (1 - \phi_k(x_i)) \log(1 - \theta_{k|y_i}) \right] \\
  = \sum_{y=0,1} N_{ky} \log(\theta_{k|y}) + (N_y - N_{ky}) \log(1 - \theta_{k|y})
  \]
  \( N_{ky} = \# \) of documents containing word \( k \) and labeled \( y \)
  \( N_y = \# \) of documents with label \( y \)
Naive Bayes estimation cont’d

- We get closed form maximum likelihood estimates

\[ J_n(\theta_k) = \sum_{y=0,1} \left[ N_{ky} \log(\theta_{k|y}) + (N_y - N_{ky}) \log(1 - \theta_{k|y}) \right] \]

\[ \frac{\partial}{\partial \theta_{k|y}} J_n(\theta_k) = \frac{N_{ky}}{\theta_{k|y}} - \frac{N_y - N_{ky}}{1 - \theta_{k|y}} = 0 \]

\[ \hat{\theta}_{k|y} = \frac{N_{ky}}{N_y} \]

(interpretation?)

- **BUT**: we have very few documents and some words are rare; these estimates are unlikely to be good

- We need regularization but what prior should we use?
Prior over the parameters

- Suppose we are dealing with simple coin flips (0/1), where parameter $\theta$ determines the probability of “1”.
- We can construct a prior over $\theta$ on the basis of
  1. a default parameter choice $p$ (in the absence of any data)
  2. how much we believe in the default choice (parameter $n'$)

- Such a prior is known as the beta distribution:

$$P(\theta) \propto \theta^{n'p} (1 - \theta)^{n'(1-p)}$$

$p = 0.5, \; n' = 0, 1, 2, 3$
Regularized Naive Bayes estimation

- In a maximum penalized likelihood estimation with Beta prior

\[ P(\theta_{k|y}) \propto \theta_{k|y}^{n'p} (1 - \theta_{k|y})^{n'(1-p)} \]

for both \( \theta_{k|y}, y = 0, 1 \), we maximize

\[
J_n(\theta_k) = \sum_{y=0,1} \left[ N_{ky} \log(\theta_{k|y}) + (N_y - N_{ky}) \log(1 - \theta_{k|y}) \right] \\
+ \sum_{y=0,1} \log P(\theta_{k|y}) \\
= \sum_{y=0,1} \left[ N_{ky} \log(\theta_{k|y}) + (N_y - N_{ky}) \log(1 - \theta_{k|y}) \right] \\
+ \sum_{y=0,1} \left[ n'p \log(\theta_{k|y}) + n'(1 - p) \log(1 - \theta_{k|y}) \right]
\]

- The resulting parameter estimates are

\[
\hat{\theta}_{k|y} = \frac{N_{ky} + n'p}{N_y + n'}
\]

Interpretation?
Feature selection

- Various objectives
  - Noise reduction
  - Regularization
  - Relevance detection
  - Reduction of computational effort

- There are roughly two main types of feature selection methods
  1. Filter method
  2. Wrapper method

- We can also do feature *weighting* rather than *selection*

  We’ll often have to resort to approximations...
Feature selection: example

- Our goal here is to reduce the number of useless word detectors
  \[ \phi_k = 0, 1 \] whether \( k^{th} \) word is present in a document
  \[ y = 0, 1 \] document label

- Suppose we have \( \hat{P}(\phi_k|y), \hat{P}(y) \), and \( \hat{P}(\phi_k) = \sum_{y=0,1} \hat{P}(\phi_k|y)\hat{P}(y) \), which we get from our (regularized) parameter estimation algorithm

- We should pick only features that provide substantial information about the labels, i.e., those with high \textit{mutual information} with the labels:

  \[
  I(\phi_k; y) = \sum_{\phi_k=0,1} \sum_{y=0,1} \hat{P}(\phi_k, y) \log_2 \left[ \frac{\hat{P}(\phi_k, y)}{\hat{P}(\phi_k)\hat{P}(y)} \right]
  \]
• Entropy (uncertainty) of a binary random variable $y$

$$H(y) = - \sum_{y=0,1} P(y) \log_2 P(y)$$

Why Shannon entropy?
101011010101000111011010011010101...
Background cont’d

- Properties of mutual information:

\[ I(\phi_k; y) = \sum_{\phi_k=0,1} \sum_{y=0,1} P(\phi_k, y) \log_2 \frac{P(\phi_k, y)}{P(\phi_k)P(y)} \]

1. \( I(\phi_k; y) = I(y; \phi_k) \) (symmetry)
2. If \( \phi_k \) and \( y \) are independent, \( I(\phi_k; y) = 0 \)
3. \( I(\phi_k; y) \leq H(y), I(\phi_k; y) \leq H(\phi_k) \)
4. \( I(\phi_k; y) = H(y) - H(y|\phi_k) = H(\phi_k) - H(\phi_k|y) \)

where the conditional entropy \( H(y|\phi_k) \) is defined as

\[ H(y|\phi_k) = \sum_{\phi_k=0,1} P(\phi_k) \left[ -\sum_{y=0,1} P(y|\phi_k) \log_2 P(y|\phi_k) \right] \]
• Venn diagram

\[ I(\phi_k; y) = H(y) - H(y | \phi_k) = H(\phi_k) - H(\phi_k | y) \]
Feature selection: example

- Reducing the number of useless word detectors
  \[ \phi_k = 0, 1 \quad \text{whether } k^{th} \text{ word is present in a document} \]
  \[ y = 0, 1 \quad \text{document label} \]

- We pick only features that provide substantial information about the labels, i.e., those with high \textit{mutual information} with the labels:
  \[
  I(\phi_k; y) = \sum_{\phi_k=0,1} \sum_{y=0,1} \hat{P}(\phi_k, y) \log_2 \left[ \frac{\hat{P}(\phi_k, y)}{\hat{P}(\phi_k) \hat{P}(y)} \right]
  \]

- What approximations are we making here?
A bit more general view

- A filtering approach
  - is generic, i.e., not optimized for any specific classifier
  - may sacrifice classification accuracy
  - modular

- A wrapper approach
  - is always tailored to a specific classifier
  - may lead to better accuracy as a result