Topics

- Generative models and text classification
  - problem formulation
  - model specification
  - model estimation with regularization
  - feature selection
Example problem

- Text classification (information retrieval)
  - only a few labeled documents $D^l = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
  - many unlabeled documents $D^u = \{x_i\}_{i=1,...,N}$ in a database

- Two possible problem formulations:
  1. Superviser learning problem
     - train with $D^l$
     - classify all the unlabeled examples in $D^u$
  2. Semi-supervised learning problem
     - train with $D^l \cup D^u$
     - classify all the unlabeled examples in $D^u$
Example problem

- We wish to build a classifier on the basis of the few labeled training examples (documents).

- Several steps:
  1. feature transformation
  2. model/classifier specification
  3. model/classifier estimation with regularization
  4. feature selection
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 (basis functions) $\{\phi_i(x)\}$ for whether a word appears in the document:

  \[ \phi_i(x) = 1, \text{ if word } i \text{ appears in document } x; \text{ zero otherwise} \]

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

- For notational simplicity we will replace each document $x$ with a fixed length vector $\Phi = [\phi_1, \ldots, \phi_m]^T$, where $\phi_i$ is set to $\phi_i(x)$. 
Classifiers

- Discriminative (e.g., support vector machine)
  - need to choose a kernel function

- Generative
  - need to define class conditional distributions
Possible document models

- We may assume, for example, that within each class of documents, the presence/absence of each word is independent of other words

\[
P(\Phi|y) = \prod_{i=1}^{m} P(\phi_i|y, i)
\]

This model still allows us to choose the frequency at which we expect to see each word in each class, i.e., we can choose \(P(\phi_i|y, i)\), where \(\phi_i, y \in \{0, 1\}\).

- This gives rise to a “Naive Bayes” model over documents and labels

\[
P(\Phi, y) = P(\Phi|y)P(y) \overset{def}{=} \left[ \prod_{i=1}^{m} P(\phi_i|y, i) \right] P(y)
\]
“Naive Bayes” model: parameters

- We parameterize the Naive Bayes model by allowing different parameter settings for each component of the class conditional distribution

\[
P(\Phi|y, \theta) = \prod_{i=1}^{m} P(\phi_i|y, \theta_i)
\]

where \(\theta = [\theta_1, \ldots, \theta_m]^T\) and \(\theta_i = [\theta_i|1, \theta_i|0]^T\) so that

\[
\theta_i|1 = P(\phi_i = 1|y = 1, i), \quad \theta_i|0 = P(\phi_i = 1|y = 0, i)
\]
“Naive Bayes” model: parameters

- We parameterize the Naive Bayes model by allowing different parameter settings for each component of the class conditional distribution

\[
P(\Phi|y, \theta) = \prod_{i=1}^{m} P(\phi_i|y, \theta_i)
\]

where \( \theta = [\theta_1, \ldots, \theta_m]^T \) and \( \theta_i = [\theta_{i|1}, \theta_{i|0}]^T \) so that

\[
\theta_{i|1} = P(\phi_i = 1|y = 1, i), \quad \theta_{i|0} = P(\phi_i = 1|y = 0, i)
\]

- Since \( \phi_i \in \{0, 1\} \) we can write the individual conditional probabilities compactly as

\[
P(\phi_i|y, \theta_i) = \theta_{i|y}^{\phi_i} (1 - \theta_{i|y})^{1-\phi_i}
\]

where the value of \( \phi_i \) selects the right probability.
Naive Bayes: parameter estimation

\[ P(\phi_i | y, \theta_i) = \theta_i^{\phi_i} (1 - \theta_i | y)^{1-\phi_i} \]

- Maximum log-likelihood criterion for a single feature

\[ J_n(\theta_i) = \sum_{t=1}^{n} \log P(\phi_{ti} | y_t, \theta_i) = \sum_{\phi_i, y} N_i(\phi_i, y) \log P(\phi_i | y, \theta_i) \]

\[ = \sum_{\phi_i, y} N_i(\phi_i, y) \left[ \phi_i \log(\theta_i | y) + (1 - \phi_i) \log(1 - \theta_i | y) \right] \]

\[ = \sum_y \left[ N_i(1, y) \log(\theta_i | y) + N_i(0, y) \log(1 - \theta_i | y) \right] \]

\[ N_i(1, y) = \# \text{ of documents containing word } i \text{ and labeled } y \]
\[ N_i(0, y) = \# \text{ of documents without word } i \text{ and labeled } y \]
Parameter estimation cont’d

- We can solve for the parameters directly

\[ J_n(\theta_i) = \sum_y \left[ N_i(1, y) \log(\theta_i | y) + N_i(0, y) \log(1 - \theta_i | y) \right] \]

\[ \frac{\partial}{\partial \theta_i | y} J_n(\theta_i) = \frac{N_i(1, y)}{\theta_i | y} - \frac{N_i(0, y)}{1 - \theta_i | y} = 0 \]

\[ \Rightarrow \hat{\theta}_i | y = \frac{N_i(1, y)}{N_i(1, y) + N_i(0, y)} \quad \text{(empirical fraction)} \]

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

- We need regularization...
Regularization

- We can instead find the parameter setting according to a maximum penalized log-likelihood criterion:

  \[ J_n(\theta_i) = \sum_{t=1}^{n} \log P(\phi_{ti}|y_t, \theta_i) + \log P(\theta_i) \]

- The prior probability \( P(\theta_i) \) should
  - prevent us from choosing extreme values for the parameters
  - permit us to express a bias towards some clear and interpretable default answer
  - allow us to specify how much we wish to bias the solution towards the default answer
Prior over the parameters

- Suppose for simplicity that we are dealing with 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 probability value in the absence of any data (parameter $p$)
  2. how strongly we believe in the default choice (equivalent sample size parameter $n'$)
Prior over the parameters

- Suppose for simplicity that we are dealing with 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 probability value in the absence of any data (parameter $p$)
  2. how strongly we believe in the default choice (equivalent sample size parameter $n'$)
- The *beta* distribution has these properties
  \[ P(\theta) \propto \theta^{n'p} (1 - \theta)^{n'(1-p)} \]
  where $n'$ and $p$ are known as hyper-parameters.

\[ p = 0.5, \ n' = 0, 1, 2, 3 \]
Regularized parameter estimation

- The effect of using the Beta (or more generally a Dirichlet) prior

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

in the penalized log-likelihood criterion

\[
J_n(\theta_i) = \sum_{t=1}^{n} \log P(\phi_{ti}|y_t, \theta_i) + \log P(\theta_i)
\]

is merely to add a few additional counts (pseudo-counts):

\[
\hat{\theta}_{i|y} = \frac{N_i(1, y) + n'p}{N_i(1, y) + N_i(0, y) + n'} \quad \text{(biased empirical fraction)}
\]
Interpretation of the regularized estimate

Let $N(y) = N_i(1, y) + N_i(0, y)$ be the number of documents in class $y$. Then

$$\hat{\theta}_i|_y = \frac{N_i(1, y) + n'p}{N_i(1, y) + N_i(0, y) + n'} = \frac{N_i(1, y) + n'p}{N(y) + n'}$$
Interpretation of the regularized estimate

Let $N(y) = N_i(1, y) + N_i(0, y)$ be the number of documents in class $y$. Then

$$
\hat{\theta}_i | y = \frac{N_i(1, y) + n'p}{N_i(1, y) + N_i(0, y) + n'} = \frac{N_i(1, y) + n'p}{N(y) + n'}
$$

$$
= \left( \frac{N_i(1, y)}{N(y) + n'} \right) + \left( \frac{n'p}{N(y) + n'} \right)
$$
Interpretation of the regularized estimate

Let \( N(y) = N_i(1, y) + N_i(0, y) \) be the number of documents in class \( y \). Then

\[
\hat{\theta}_i|y = \frac{N_i(1, y) + n'p}{N_i(1, y) + N_i(0, y) + n'} = \frac{N_i(1, y) + n'p}{N(y) + n'}
\]

\[
= \left( \frac{N_i(1, y)}{N(y) + n'} \right) + \left( \frac{n'p}{N(y) + n'} \right)
\]

\[
= \left( \frac{N(y)}{N(y) + n'} \right) \cdot \frac{N_i(1, y)}{N(y)} + \left( \frac{n'}{N(y) + n'} \right) \cdot p
\]
Interpretation of the regularized estimate

Let \( N(y) = N_i(1, y) + N_i(0, y) \) be the number of documents in class \( y \). Then

\[
\hat{\theta}_{i|y} = \frac{N_i(1, y) + n'p}{N_i(1, y) + N_i(0, y) + n'} = \frac{N_i(1, y) + n'p}{N(y) + n'}
\]

\[
= \left( \frac{N_i(1, y)}{N(y) + n'} \right) + \left( \frac{n'p}{N(y) + n'} \right)
\]

\[
= \left( \frac{N(y)}{N(y) + n'} \right) \cdot \frac{N_i(1, y)}{N(y)} + \left( \frac{n'}{N(y) + n'} \right) \cdot p
\]

\[
= \left( \frac{N(y)}{N(y) + n'} \right) \cdot \hat{\theta}_{i|y} + \left( \frac{n'}{N(y) + n'} \right) \cdot p
\]
Example problem cont’d

• We wish to build a classifier on the basis of the few labeled training examples (documents).

• Several steps:
  1. feature transformation
  2. model/classifier specification
  3. model/classifier estimation with regularization
  4. feature selection
Feature selection

- In a classification setting there are many possible reasons for us to select only a relevant subset of the input features, not all of them:
  - noise reduction
  - additional regularization
  - reduction of computational effort
  etc.

- We need a criterion for finding features that might be useful for the classification task
Feature selection cont’d

• Suppose we have already estimated $P(\phi_i|y, \hat{\theta}_i)$ for each word $i$ and label $y$ based on the available data.

For notational simplicity, we will remove any explicit reference to the maximum likelihood parameters and instead use “hats” for estimated probabilities

$$\hat{P}(y) \quad \text{(estimated class freq.)}$$

$$\hat{P}(\phi_i, y) = P(\phi_i|y, \hat{\theta}_i) \hat{P}(y)$$

$$\hat{P}(\phi_i) = \sum_{y=0,1} \hat{P}(\phi_i, y) \quad \text{(estimated word freq.)}$$

• Our goal is to use these probabilities somehow to guide the selection of useful word features.
Feature selection cont’d

• We can select features which by themselves would provide substantial amount of information about the label

• More formally, we choose features that have a high value of \textit{mutual information} with the labels:

\[
\hat{I}(\phi_i; y) = \sum_{\phi_i=0,1} \sum_{y=0,1} \hat{P}(\phi_i, y) \log_2 \left[ \frac{\hat{P}(\phi_i, y)}{\hat{P}(\phi_i)\hat{P}(y)} \right]
\]

Mutual information can be viewed as a measure of distance between $\hat{P}(\phi_i, y)$ and $\hat{P}(\phi_i)\hat{P}(y)$, where

- $\hat{P}(\phi_i, y)$ is our best estimate of the relation between the single feature and the label

- $\hat{P}(\phi_k)\hat{P}(y)$ would be our estimate if we assumed that the feature and the label are \textit{independent}
A bit of background

- 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:

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

1. \( I(\phi_i; y) = I(y; \phi_i) \) (symmetry)
Properties of mutual information:

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

1. \( I(\phi_i; y) = I(y; \phi_i) \) (symmetry)
2. If \( \phi_i \) and \( y \) are independent, \( I(\phi_i; y) = 0 \)
Properties of mutual information:

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

1. \( I(\phi_i; y) = I(y; \phi_i) \) (symmetry)
2. If \( \phi_i \) and \( y \) are independent, \( I(\phi_i; y) = 0 \)
3. \( I(\phi_i; y) \leq H(y), I(\phi_i; y) \leq H(\phi_i) \)
Background cont’d

• Properties of mutual information:

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

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

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

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

\[ I(\phi_i; y) = H(y) - H(y|\phi_i) = H(\phi_i) - H(\phi_i|y) \]
Back to feature selection

- We choose features that have a high value of \textit{mutual information} with the labels:

\[
\hat{I}(\phi_i; y) = \sum_{\phi_i=0,1} \sum_{y=0,1} \hat{P}(\phi_i, y) \log_2 \left[ \frac{\hat{P}(\phi_i, y)}{\hat{P}(\phi_i) \hat{P}(y)} \right]
\]

- There are many unanswered questions:
  - how many features do we include?
  - what about redundant features?
  - coordination among features?
  - which classifier does this type of selection benefit?