Text as Data

Justin Grimmer

Associate Professor
Department of Political Science
University of Chicago

February 14th, 2018

Naive Bayes and General Problem Setup

Suppose we have document i, ($i = 1, \ldots, N$) with J features

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k $(k = 1, \ldots, K)$
$\{C_1, C_2, \ldots, C_K\}$

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with J features $\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k $(k = 1, \ldots, K)$ $\{C_1, C_2, \ldots, C_K\}$
Subset of labeled documents $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_{N_{\text{train}}})$ where $Y_i \in \{C_1, C_2, \ldots, C_K\}$.

Naive Bayes and General Problem Setup

Suppose we have document i, ($i = 1, \ldots, N$) with J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k ($k = 1, \ldots, K$)
$\{C_1, C_2, \ldots, C_K\}$
Subset of labeled documents $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_{N_{\text{train}}})$ where
$Y_i \in \{C_1, C_2, \ldots, C_K\}$.
Goal: classify every document into one category.

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k $(k = 1, \ldots, K)$
$\{C_1, C_2, \ldots, C_K\}$
Subset of labeled documents $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_{N_{\text{train}}})$ where
$Y_i \in \{C_1, C_2, \ldots, C_K\}$.
Goal: classify every document into one category.
Learn a function that maps from space of (possible) documents to categories

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k $(k = 1, \ldots, K)$
$\{C_1, C_2, \ldots, C_K\}$
Subset of labeled documents $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_{N_{\text{train}}})$ where
$Y_i \in \{C_1, C_2, \ldots, C_K\}$.
Goal: classify every document into one category.
Learn a function that maps from space of (possible) documents to categories
To do this: use hand coded observations to estimate (train) regression model

Naive Bayes and General Problem Setup

Suppose we have document i, $(i = 1, \ldots, N)$ with J features
$\boldsymbol{x}_i = (x_{1i}, x_{2i}, \ldots, x_{Ji})$
Set of K categories. Category k $(k = 1, \ldots, K)$
$\{C_1, C_2, \ldots, C_K\}$
Subset of labeled documents $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_{N_{\text{train}}})$ where
$Y_i \in \{C_1, C_2, \ldots, C_K\}$.
Goal: classify every document into one category.
Learn a function that maps from space of (possible) documents to categories
To do this: use hand coded observations to estimate (train) regression model
Apply model to test data, classify those observations

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Goal: For each document $\boldsymbol{x}_i$, we want to infer most likely category

$$(0.1)$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Goal: For each document $\boldsymbol{x}_i$, we want to infer most likely category

$$C_{\text{Max}} = \arg\max_k p(C_k | \boldsymbol{x}_i) \tag{0.1}$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Goal: For each document $\boldsymbol{x}_i$, we want to infer most likely category

$$C_{\text{Max}} = \arg\max_k p(C_k|\boldsymbol{x}_i) \tag{0.1}$$

We're going to use Bayes' rule to estimate $p(C_k|\boldsymbol{x}_i)$.

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Goal: For each document $\boldsymbol{x}_i$, we want to infer most likely category

$$C_{\text{Max}} = \arg\max_k p(C_k|\boldsymbol{x}_i)$$

We're going to use Bayes' rule to estimate $p(C_k|\boldsymbol{x}_i)$.

$$p(C_k|\boldsymbol{x}_i) = \frac{p(C_k, \boldsymbol{x}_i)}{p(\boldsymbol{x}_i)} \quad (0.1)$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Goal: For each document $\boldsymbol{x}_i$, we want to infer most likely category

$$C_{\text{Max}} = \arg\max_k p(C_k|\boldsymbol{x}_i)$$

We're going to use Bayes' rule to estimate $p(C_k|\boldsymbol{x}_i)$.

$$\begin{aligned} p(C_k|\boldsymbol{x}_i) &= \frac{p(C_k, \boldsymbol{x}_i)}{p(\boldsymbol{x}_i)} \\ &= \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \end{aligned} \tag{0.1}$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Goal: For each document $\boldsymbol{x}_i$, we want to infer most likely category

$$C_{\text{Max}} = \arg\max_k p(C_k|\boldsymbol{x}_i)$$

We're going to use Bayes' rule to estimate $p(C_k|\boldsymbol{x}_i)$.

$$\begin{aligned} p(C_k|\boldsymbol{x}_i) &= \frac{p(C_k, \boldsymbol{x}_i)}{p(\boldsymbol{x}_i)} \\ &= \frac{\overbrace{p(C_k)}^{\text{Proportion in } C_k} \underbrace{p(\boldsymbol{x}_i|C_k)}_{\text{Language model}}}{p(\boldsymbol{x}_i)} \end{aligned}$$

Naive Bayes and Optimization (Jurafsky Inspired Slide)

Naive Bayes and Optimization (Jurafsky Inspired Slide)

$$C_{\text{Max}} = \arg\max_k \; p(C_k|\boldsymbol{x}_i)$$

Naive Bayes and Optimization (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_k \; p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \arg\max_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)}
\end{aligned}
$$

Naive Bayes and Optimization (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_k \ p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \arg\max_k \ \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\
C_{\text{Max}} &= \arg\max_k \ p(C_k)p(\boldsymbol{x}_i|C_k)
\end{aligned}
$$

Naive Bayes and Optimization (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_k \ p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \arg\max_k \ \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\
C_{\text{Max}} &= \arg\max_k \ p(C_k)p(\boldsymbol{x}_i|C_k)
\end{aligned}
$$

Two probabilities to estimate:

Naive Bayes and Optimization (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \text{arg max}_k \; p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \text{arg max}_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\
C_{\text{Max}} &= \text{arg max}_k \; p(C_k)p(\boldsymbol{x}_i|C_k)
\end{aligned}
$$

Two probabilities to estimate:

$p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)

Naive Bayes and Optimization (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \text{arg max}_k \; p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \text{arg max}_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\
C_{\text{Max}} &= \text{arg max}_k \; p(C_k)p(\boldsymbol{x}_i|C_k)
\end{aligned}
$$

Two probabilities to estimate:

$p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)

$p(\boldsymbol{x}_i|C_k)$ complicated without assumptions

Naive Bayes and Optimization (Jurafsky Inspired Slide)

$$
\begin{aligned}
C_{\text{Max}} &= \arg\max_k \; p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \arg\max_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\
C_{\text{Max}} &= \arg\max_k \; p(C_k)p(\boldsymbol{x}_i|C_k)
\end{aligned}
$$

Two probabilities to estimate:

$p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)

$p(\boldsymbol{x}_i|C_k)$ complicated without assumptions

- Imagine each x_{ij} just binary indicator. Then 2^J possible $\boldsymbol{x}_i$ documents

Naive Bayes and Optimization (Jurafsky Inspired Slide)

$$\begin{aligned}
C_{\text{Max}} &= \arg\max_k \; p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \arg\max_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\
C_{\text{Max}} &= \arg\max_k \; p(C_k)p(\boldsymbol{x}_i|C_k)
\end{aligned}$$

Two probabilities to estimate:

$p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)

$p(\boldsymbol{x}_i|C_k)$ complicated without assumptions

- Imagine each x_{ij} just binary indicator. Then 2^J possible $\boldsymbol{x}_i$ documents
- Simplify: assume each feature is independent

Naive Bayes and Optimization (Jurafsky Inspired Slide)

$$\begin{aligned}
C_{\text{Max}} &= \arg\max_k \; p(C_k|\boldsymbol{x}_i) \\
C_{\text{Max}} &= \arg\max_k \; \frac{p(C_k)p(\boldsymbol{x}_i|C_k)}{p(\boldsymbol{x}_i)} \\
C_{\text{Max}} &= \arg\max_k \; p(C_k)p(\boldsymbol{x}_i|C_k)
\end{aligned}$$

Two probabilities to estimate:

$p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)

$p(\boldsymbol{x}_i|C_k)$ complicated without assumptions

- Imagine each x_{ij} just binary indicator. Then 2^J possible $\boldsymbol{x}_i$ documents
- Simplify: assume each feature is independent

$$p(\boldsymbol{x}_i|C_k) = \prod_{j=1}^{J} p(x_{ij}|C_k)$$

Naive Bayes and Optimization (Jurafsky Inspired Slide)

Two components to estimation:

- $p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)
- $p(\boldsymbol{x}_i|C_k) = \prod_{j=1}^{J} p(x_{ij}|C_k)$

Naive Bayes and Optimization (Jurafsky Inspired Slide)

Two components to estimation:

- $p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)
- $p(\boldsymbol{x}_i|C_k) = \prod_{j=1}^{J} p(x_{ij}|C_k)$

Maximum likelihood estimation (training set):

Naive Bayes and Optimization (Jurafsky Inspired Slide)

Two components to estimation:

- $p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)
- $p(\boldsymbol{x}_i|C_k) = \prod_{j=1}^{J} p(x_{ij}|C_k)$

Maximum likelihood estimation (training set):

$$p(x_{im} = z|C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)}}{\text{No(C} = \text{C}_k\text{)}}$$

Naive Bayes and Optimization (Jurafsky Inspired Slide)

Two components to estimation:

- $p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)
- $p(\boldsymbol{x}_i|C_k) = \prod_{j=1}^{J} p(x_{ij}|C_k)$

Maximum likelihood estimation (training set):

$$p(x_{im} = z|C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)}}{\text{No(C} = \text{C}_k\text{)}}$$

Problem: What if No(Docs$_{ij}$ = z and C = C$_k$) = 0 ?

Naive Bayes and Optimization (Jurafsky Inspired Slide)

Two components to estimation:

- $p(C_k) = \frac{\text{No. Documents in } k}{\text{No. Documents}}$ (training set)
- $p(\boldsymbol{x}_i|C_k) = \prod_{j=1}^{J} p(x_{ij}|C_k)$

Maximum likelihood estimation (training set):

$$p(x_{im} = z|C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)}}{\text{No(C} = \text{C}_k\text{)}}$$

Problem: What if No(Docs_{ij} = z and C = C_k) = 0 ?
$\prod_{j=1}^{J} p(x_{ij}|C_k) = 0$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z|C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)} + 1}{\text{No(C} = \text{C}_k) + k}$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z|C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k\text{)} + 1}{\text{No(C} = \text{C}_k) + k}$$

Algorithm steps:

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z | C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)} + 1}{\text{No(C= C}_k) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\boldsymbol{x}_i | C_k)$ on training data

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z|C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k\text{)} \; + 1}{\text{No(C} = \text{C}_k) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\boldsymbol{x}_i|C_k)$ on training data
2) Use this to identify most likely C_k for each document i in test set

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z | C_k) = \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k\text{)} + 1}{\text{No(C} = \text{C}_k\text{)} + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\boldsymbol{x}_i | C_k)$ on training data
2) Use this to identify most likely C_k for each document i in test set

$$C_i = \text{arg max }_k \hat{p}(C_k) \hat{p}(\boldsymbol{x}_i | C_k)$$

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z|C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)} + 1}{\text{No(C} = \text{C}_k) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\boldsymbol{x}_i|C_k)$ on training data
2) Use this to identify most likely C_k for each document i in test set

$$C_i \quad = \quad \text{arg max }_k \hat{p}(C_k)\hat{p}(\boldsymbol{x}_i|C_k)$$

Simple intuition about Naive Bayes:

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z|C_k) \;=\; \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k\text{)} + 1}{\text{No(C}= \text{C}_k) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\boldsymbol{x}_i|C_k)$ on training data
2) Use this to identify most likely C_k for each document i in test set

$$C_i \;=\; \text{arg max }_k \hat{p}(C_k)\hat{p}(\boldsymbol{x}_i|C_k)$$

Simple intuition about Naive Bayes:

- Learn what documents in class j look like

Naive Bayes and General Problem Setup (Jurafsky Inspired Slide)

Solution: smoothing (Bayesian estimation)

$$p(x_{ij} = z | C_k) \quad = \quad \frac{\text{No(Docs}_{ij} = \text{z and C} = \text{C}_k \text{)} + 1}{\text{No(C= C}_k) + k}$$

Algorithm steps:

1) Learn $\hat{p}(C)$ and $\hat{p}(\boldsymbol{x}_i | C_k)$ on training data
2) Use this to identify most likely C_k for each document i in test set

$$C_i \quad = \quad \text{arg max }_k \hat{p}(C_k)\hat{p}(\boldsymbol{x}_i | C_k)$$

Simple intuition about Naive Bayes:

- Learn what documents in class j look like
- Find class k that document i is most similar to

Naive Bayes and Unigram Language Models

Assume the following data generating process (should look familiar)

$$
\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\
\boldsymbol{\theta} &\sim \text{Dirichlet}(\boldsymbol{\lambda}) \\
\boldsymbol{\tau}_i &\sim \text{Multinomial}(1, \boldsymbol{\pi}) \\
\boldsymbol{x}_i | \tau_{ik} = 1, \boldsymbol{\theta} &\sim \text{Multinomial}(n_i, \boldsymbol{\theta}_k)
\end{aligned}
$$

Naive Bayes and Unigram Language Models

Assume the following data generating process (should look familiar)

$$\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\
\boldsymbol{\theta} &\sim \text{Dirichlet}(\boldsymbol{\lambda}) \\
\boldsymbol{\tau}_i &\sim \text{Multinomial}(1, \boldsymbol{\pi}) \\
\boldsymbol{x}_i | \tau_{ik} = 1, \boldsymbol{\theta} &\sim \text{Multinomial}(n_i, \boldsymbol{\theta}_k)
\end{aligned}$$

If we randomly sample documents N_{train} and label them ($\boldsymbol{Y}$), then we can estimate

Naive Bayes and Unigram Language Models

Assume the following data generating process (should look familiar)

$$
\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\
\boldsymbol{\theta} &\sim \text{Dirichlet}(\boldsymbol{\lambda}) \\
\boldsymbol{\tau}_i &\sim \text{Multinomial}(1, \boldsymbol{\pi}) \\
\boldsymbol{x}_i | \tau_{ik} = 1, \boldsymbol{\theta} &\sim \text{Multinomial}(n_i, \boldsymbol{\theta}_k)
\end{aligned}
$$

If we randomly sample documents N_{train} and label them $(\boldsymbol{Y})$, then we can estimate

$$
\widehat{\pi}_k = \frac{\sum_{i=1}^{N} I(Y_i = k) + \alpha_k}{N_{\text{train}} + \sum_{k=1}^{K} \alpha_k}
$$

Naive Bayes and Unigram Language Models

Assume the following data generating process (should look familiar)

$$
\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\
\boldsymbol{\theta} &\sim \text{Dirichlet}(\boldsymbol{\lambda}) \\
\boldsymbol{\tau}_i &\sim \text{Multinomial}(1, \boldsymbol{\pi}) \\
\boldsymbol{x}_i | \tau_{ik} = 1, \boldsymbol{\theta} &\sim \text{Multinomial}(n_i, \boldsymbol{\theta}_k)
\end{aligned}
$$

If we randomly sample documents N_{train} and label them $(\boldsymbol{Y})$, then we can estimate

$$
\begin{aligned}
\widehat{\pi}_k &= \frac{\sum_{i=1}^{N} I(Y_i = k) + \alpha_k}{N_{\text{train}} + \sum_{k=1}^{K} \alpha_k} \\
\widehat{\theta}_{jk} &= \frac{\sum_{i=1}^{N} I(Y_i = k) x_{ij} + \lambda_j}{\sum_{j=1}^{J} \sum_{i=1}^{N} I(Y_i = k) x_{ij} + \sum_{j=1}^{J} \lambda_j}
\end{aligned}
$$

Naive Bayes and Unigram Language Models

The probability a new document has $\tau_{ik} = 1$ is then

Naive Bayes and Unigram Language Models

The probability a new document has $\tau_{ik} = 1$ is then

$$p(\tau_{ik} = 1 | \boldsymbol{x}_i, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\theta}}) \quad \propto \quad p(\tau_{ik} = 1) p(\boldsymbol{x}_i | \boldsymbol{\theta}, \tau_{ik} = 1)$$

Naive Bayes and Unigram Language Models

The probability a new document has $\tau_{ik} = 1$ is then

$$
\begin{aligned}
p(\tau_{ik} = 1 | \boldsymbol{x}_i, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\theta}}) &\propto p(\tau_{ik} = 1) p(\boldsymbol{x}_i | \boldsymbol{\theta}, \tau_{ik} = 1) \\
&\propto \widehat{\pi_k} \prod_{j=1}^{J} \left(\widehat{\theta}_{jk} \right)^{x_{ij}}
\end{aligned}
$$

Naive Bayes and Unigram Language Models

The probability a new document has $\tau_{ik} = 1$ is then

$$
\begin{aligned}
p(\tau_{ik} = 1 | \boldsymbol{x}_i, \widehat{\boldsymbol{\pi}}, \widehat{\boldsymbol{\theta}}) & \propto p(\tau_{ik} = 1) p(\boldsymbol{x}_i | \boldsymbol{\theta}, \tau_{ik} = 1) \\
& \propto \widehat{\pi_k} \prod_{j=1}^{J} \left(\widehat{\theta}_{jk} \right)^{x_{ij}} \\
& \propto \overbrace{\widehat{\pi_k}}^{p(C_k)} \underbrace{\prod_{j=1}^{J} \left(\widehat{\theta}_{jk} \right)^{x_{ij}}}_{\text{Unigram model}}
\end{aligned}
$$

Some R Code

```
library(e1071)
dep<- c(labels, rep(NA, no.testSet))
dep<- as.factor(dep)
out<- naiveBayes(dep~., as.data.frame(tdm))
predicts<- predict(out, as.data.frame(tdm[-training.set,]))
```

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, LASSO, . . .: focused on individual document classification.

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, LASSO, ...: focused on individual document classification.
But what if we're focused on proportions only?

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, LASSO, ...: focused on individual document classification.

But what if we're focused on proportions only?

Hopkins and King (2010): method for characterizing distribution of classes

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, LASSO, ...: focused on individual document classification.
But what if we're focused on proportions only?
Hopkins and King (2010): method for characterizing distribution of classes
Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, LASSO, . . .: focused on individual document classification.
But what if we're focused on proportions only?
Hopkins and King (2010): method for characterizing distribution of classes
Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, LASSO, ...: focused on individual document classification.
But what if we're focused on proportions only?
Hopkins and King (2010): method for characterizing distribution of classes
Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies
- Hopkins and King (2010): extend the method to text documents

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, LASSO, ...: focused on individual document classification.

But what if we're focused on proportions only?

Hopkins and King (2010): method for characterizing distribution of classes

Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies
- Hopkins and King (2010): extend the method to text documents

Basic intuition:

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, LASSO, ...: focused on individual document classification.
But what if we're focused on proportions only?
Hopkins and King (2010): method for characterizing distribution of classes
Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies
- Hopkins and King (2010): extend the method to text documents

Basic intuition:

- Examine joint distribution of characteristics (without making Naive Bayes like assumption)
- Focus on distributions (only) makes this analysis possible

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(J x 1)$ vector]

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$P(\boldsymbol{x}) = \text{probability of observing } \boldsymbol{x}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(J x 1)$ vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$\begin{aligned} P(\boldsymbol{x}) &= \text{probability of observing } \boldsymbol{x} \\ P(\boldsymbol{x}|C_j) &= \text{Probability of observing } \boldsymbol{x} \text{ conditional on category } C_j \end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(J x 1)$ vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$\begin{aligned} P(\boldsymbol{x}) &= \text{probability of observing } \boldsymbol{x} \\ P(\boldsymbol{x}|C_j) &= \text{Probability of observing } \boldsymbol{x} \text{ conditional on category } C_j \\ P(\boldsymbol{X}|C) &= \text{Matrix collecting vectors} \end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$\begin{aligned}
P(\boldsymbol{x}) &= \text{probability of observing } \boldsymbol{x} \\
P(\boldsymbol{x}|C_j) &= \text{Probability of observing } \boldsymbol{x} \text{ conditional on category } C_j \\
P(\boldsymbol{X}|C) &= \text{Matrix collecting vectors} \\
P(C) &= P(C_1, C_2, \ldots, C_K) \text{ target quantity of interest}
\end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$\begin{aligned}
P(\boldsymbol{x}) &= \text{probability of observing } \boldsymbol{x} \\
P(\boldsymbol{x}|C_j) &= \text{Probability of observing } \boldsymbol{x} \text{ conditional on category } C_j \\
P(\boldsymbol{X}|C) &= \text{Matrix collecting vectors} \\
P(C) &= P(C_1, C_2, \ldots, C_K) \text{ target quantity of interest}
\end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

$$\underbrace{P(\boldsymbol{x})}_{2^J x 1} = \underbrace{P(\boldsymbol{x}|C)}_{2^J x K}\underbrace{P(C)}_{K x 1}$$

Matrix algebra problem to solve, for $P(C)$

Like Naive Bayes, requires two pieces to estimate

Complication $2^J >> $ no. documents

Kernel Smoothing Methods (without a formal model)

- $P(\boldsymbol{x})$ = estimate directly from test set
- $P(\boldsymbol{x}|C)$ = estimate from training set
    - Key assumption: $P(\boldsymbol{x}|C)$ in training set is equivalent to $P(\boldsymbol{x}|C)$ in test set
- If true, can perform biased sampling of documents, worry less about drift...

Algorithm Summarized

- Estimate $\hat{p}(\boldsymbol{x})$ from test set
- Estimate $\hat{p}(\boldsymbol{x}|C)$ from training set
- Use $\hat{p}(\boldsymbol{x})$ and $\hat{p}(\boldsymbol{x}|C)$ to solve for $p(C)$

Assessing Model Performance

Not classifying individual documents $\rightarrow$ different standards
Mean Square Error :

$$\mathrm{E}[(\hat{\theta} - \theta)^2] = \mathrm{var}(\hat{\theta}) + \mathrm{Bias}(\hat{\theta}, \theta)^2$$

Suppose we have true proportions $P(C)^{\text{true}}$. Then, we'll estimate Root Mean Square Error

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{\sum_{j=1}^{J} (P(C_j)^{\text{true}} - P(C_j))}{J}} \\ \text{Mean Abs. Prediction Error} &= |\frac{\sum_{j=1}^{J} (P(C_j)^{\text{true}} - P(C_j))}{J}| \end{aligned}$$

Visualize: plot true and estimated proportions

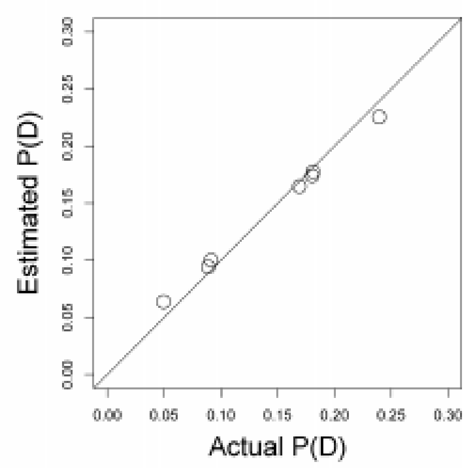
Estimated P(D)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Actual P(D)
0.00
0.05
0.10
0.15
0.20
0.25
0.30

Table 1 Performance of Our Nonparametric Approach and Four Support Vector Machine Analyses

	Percent of Blog Posts Correctly Classified			
	In-Sample Fit	In-Sample Cross-Validation	Out-of-Sample Prediction	Mean Absolute Proportion Error
Nonparametric	—	—	—	1.2
Linear	67.6	55.2	49.3	7.7
Radial	67.6	54.2	49.1	7.7
Polynomial	99.7	48.9	47.8	5.3
Sigmoid	15.6	15.6	18.2	23.2

Notes: Each row is the optimal choice over numerous individual runs given a specific kernel. Leaving aside the sigmoid kernel, individual classification performance in the first three columns does not correlate with mean absolute error in the document category proportions in the last column.

Using the House Press Release Data

Method	RMSE	APSE
ReadMe	0.036	0.056
NaiveBayes	0.096	0.14
SVM	0.052	0.084

Code to Run in R

Control file:

filename	truth	trainingset
20July2009LEWIS53.txt	4	1
26July2006LEWIS249.txt	2	0

```
tdm<- undergrad(control=control, fullfreq=F)
process<- preprocess(tdm)
output<- undergrad(process)
output$est.CSMF ## proportion in each category
output$true.CSMF ## if labeled for validation set (but not
used in training set)
```