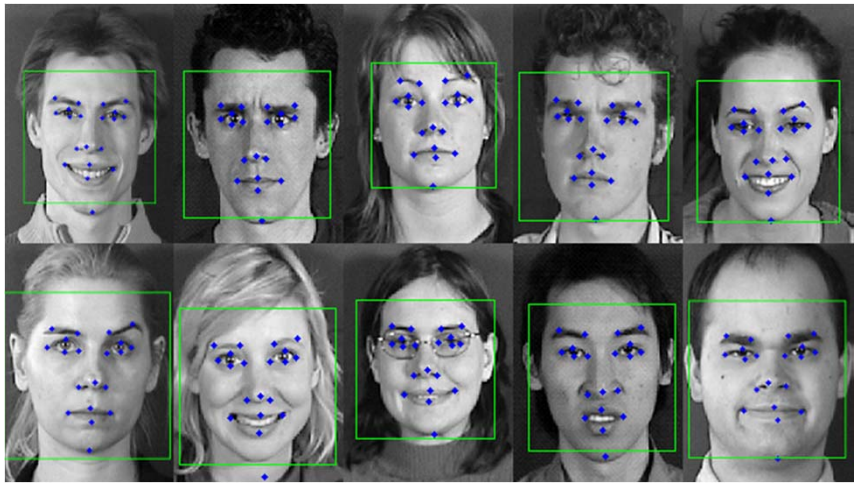
Learning from Data
6. Classification: Naïve Bayes

Spring 2017

Classification

- Supervised (preditive) learning: given examples with labels, predict labels for all unseen examples
  - Classification:
    - label = category,
    - $\mathbf{y} \in \mathcal{Y} = \{1, \ldots, m\}$, m = number of classes
    - $\ell(\mathbf{x}, y, h) = 1(h(\mathbf{x}) \neq y)$, Risk = $P(Y \neq h(\mathbf{X})) = P(\text{Error})$



$\mathbf{x}$ = facial geometry features

y = gender label

Naïve Bayes

- Generative learning
- d features (components) in each feature vector:

$$\mathbf{x} = \left(x_1, \ldots, x_i, \ldots, x_d\right)^\top$$

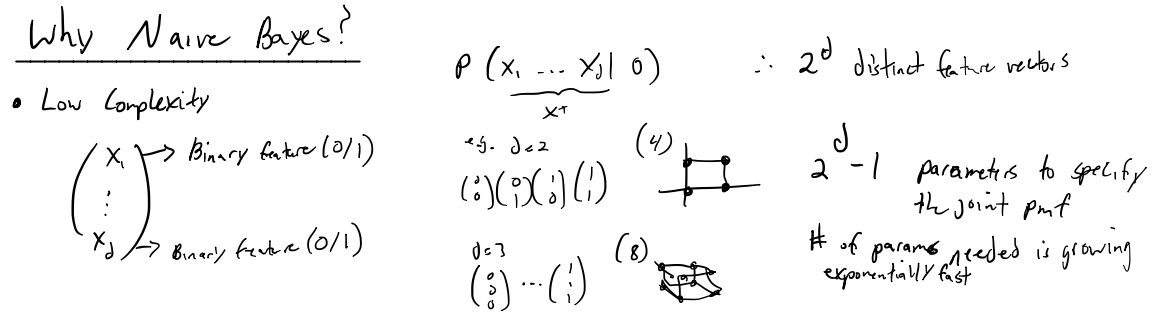
- Naïve Bayes Assumption: all features are conditionally independent given the class label

$$\forall y,\ p(\mathbf{x}|y,\theta) = p(x_1|y,\theta_1)\cdots p(x_d|y,\theta_d) = \prod_{i=1}^{d} p(x_i|y,\theta_i)$$

$$\theta = (\theta_0, \theta_1, \ldots, \theta_d)^\top, \quad \theta_0 = (p(y=1), \ldots, p(y=m))^\top$$

- Resulting MPE decision rule: Naïve Bayes classifier
- Why Naïve? We do not expect all features to be independent even conditional on class label

Naïve Bayes



- Although assumption is false, the resulting classifier works quite well in practice.

- It has only on the order of $O(md)$ parameters to learn ➔ relatively immune to overfitting
  - Example: Suppose that all features are binary, say taking only values +1 or -1, then:
    - In NB model, we need to specify only 1 scalar parameter per feature per class, namely $P(X_i = +1|y), y = 1,\ldots,m$ or md scalar parameters in total
    - But specifying the joint pmf of all d binary features requires $(2^d\text{-}1)$ scalar parameters per class or $m(2^d\text{-}1)$ scalar parameters in total

Naïve Bayes

- Class-conditional likelihood functions:

$$p(x_i|y, \theta_i), i = 1, \ldots, d$$

  - may be either completely parametric or completely non-parametric or some components may be modeled parametrically and others non-parametrically

  - some or all feature components may be discrete, continuous, or mixed

  - Gaussian Naïve Bayes: $\forall i, y,\ p(x_i|y, \theta_i) \sim \mathcal{N}\left(\mu_{iy}, \sigma_{iy}^2\right)$

  - Categorical Naïve Bayes: all features are discrete and take only a finite number of possible values

Naïve Bayes

- Notation summary:

$y =$ class label $\in \{1, \ldots, m\}$

$j =$ sample index $\in \{1, \ldots, n\}$

$i =$ feature (component) index $\in \{1, \ldots, d\}$

$\theta_i =$ parameters for class-conditional pdfs/pmfs of feature i across all classes

$\theta_i = \{\theta_{iy}, y = 1, \ldots, m\}$, where

$\theta_{iy} =$ parameters for class-conditional pdf/pmf of feature i in class y, and

feature i for class y

$p(x_i|y, \theta_i) = p(x_i|y, \theta_{iy})$ for all i, y

$\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ (training set)

$$\text{feature index} \downarrow \quad \begin{matrix} i = 1 \\ \vdots \\ i \\ \vdots \\ i = d \end{matrix} \qquad \mathbf{x}_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{i1} \\ \vdots \\ x_{d1} \end{pmatrix}, \quad \ldots \quad \mathbf{x}_j = \begin{pmatrix} x_{1j} \\ \vdots \\ x_{ij} \\ \vdots \\ x_{dj} \end{pmatrix}, \quad \ldots \quad \mathbf{x}_n = \begin{pmatrix} x_{1n} \\ \vdots \\ x_{in} \\ \vdots \\ x_{dn} \end{pmatrix}$$

$$j = 1, \quad \ldots \qquad j, \qquad \ldots \qquad j = n$$

$$\text{sample index} \longrightarrow$$

Naïve Bayes

- Training set: $\mathcal{D} = \{(\mathbf{x}_j, y_j), j = 1, \ldots, n\}$
- Training set for feature i:

$$\mathcal{D}_i = \{(x_{ij}, y_j), j = 1, \ldots, n\}$$

- Training set for feature i and class y:

$$\mathcal{D}_{iy} = \{(x_{ij}, y_j), j : y_j = y\}$$

- Key Result: for each feature i and each class y, the ML estimate of θ_{iy} based on $\mathcal{D}$ is the same as the ML estimate of θ_{iy} based on $\mathcal{D}_{iy}$
- Intuition: features independent in every class ⇒ parameters of feature distribution can be estimated independently for each feature in each class

Naïve Bayes

Proof:

$$p(\mathcal{D}|\theta) = \prod_{j=1}^{n} p(\mathbf{x}_j, y_j|\theta)$$

$$= \prod_{j=1}^{n} \left[p(y_j) \prod_{i=1}^{d} p(x_{ij}|y_j, \theta_i) \right]$$

Naïve Bayes assumption:
$p(\mathbf{x}_j|y_j) = \prod_{i=1}^{d} p(x_{ij}|y_j, \theta_i)$

$$= \left(\prod_{j=1}^{n} p(y_j) \right) \prod_{i=1}^{d} \prod_{j=1}^{n} p(x_{ij}|y_j, \theta_i)$$

$$= \left(\prod_{y=1}^{m} p(y)^{n_y} \right) \prod_{i=1}^{d} \prod_{y=1}^{m} \prod_{j:y_j=y} p(x_{ij}|y, \theta_{iy})$$

n_y = # class y samples
$= \sum_{j=1}^{n} 1(y_j = y)$

$$\Rightarrow \ln p(\mathcal{D}|\theta) = \sum_{y=1}^{m} n_y \ln p(y) + \sum_{i=1}^{d} \sum_{y=1}^{m} \sum_{j:y_j=y} \ln p(x_{ij}|y, \theta_{iy})$$

$$\Rightarrow \frac{1}{n} \ln p(\mathcal{D}|\theta) = \sum_{y=1}^{m} \left(\frac{n_y}{n} \right) \ln \theta_{0y} + \sum_{i=1}^{d} \sum_{y=1}^{m} \left(\frac{n_y}{n} \right) \left[\frac{1}{n_y} \sum_{j:y_j=y} \ln p(x_{ij}|y, \theta_{iy}) \right]$$

maximize

can ignore other features

Naïve Bayes

Thus,

$$\widehat{\theta}_{ML}(\mathcal{D}) = \arg\max_{\theta} \frac{1}{n} \ln p(\mathcal{D}|\theta),$$

$$\theta = (\theta_0, \{\theta_{iy}, i = 1, \ldots, d, y = 1, \ldots, m\})$$

prior

$$\Rightarrow \widehat{\theta}_{0,ML}(\mathcal{D}) = \arg\max_{\theta_0} \left[\sum_{y=1}^{m} \left(\frac{n_y}{n} \right) \ln \theta_{0y} \right],$$

$$\forall i, y, \ \widehat{\theta}_{iy,ML}(\mathcal{D}) = \arg\max_{\theta_{iy}} \left[\frac{1}{n_y} \sum_{j:y_j=y} \ln p(x_{ij}|y, \theta_{iy}) \right]$$

$$= \widehat{\theta}_{iy,ML}(\mathcal{D}_{iy})$$

$$p(x_{ij}|y, \theta_{iy}) \stackrel{?}{=} \frac{(\#\text{ Word ID} == i)_y \, (\text{counts of } i)_y}{\text{number of total words in class } y}$$

Categorical Naïve Bayes

- All features are categorical:
  - Without loss of generality (w.l.o.g.): $\forall i,\ x_i \in \{1, \ldots, W\}$
- Notation summary:

$$y = \text{class label } \in \{1, \ldots, m\}$$
$$j = \text{sample index } \in \{1, \ldots, n\}$$
$$i = \text{feature (component) index } \in \{1, \ldots, d\}$$
$$w = \text{feature value } \in \{1, \ldots, W\}$$

- Model:

$$\forall y, j, i, w,\ P(X_{ij} = w | Y_j = y) = \beta_{w,y,i}, \text{ where}$$
$$\forall y, i, w,\ \beta_{w,y,i} \geq 0, \text{ and } \forall y, i,\ \sum_{w=1}^{W} \beta_{w,y,i} = 1$$

- Parameters:

$$\theta = \{\underbrace{p(y)}_{\theta_{0y}}, \underbrace{\beta_{w,y,i}, w = 1, \ldots, W}_{\theta_{iy} = \{\beta_{w,y,i}, w = 1, \ldots, W\}}, y = 1, \ldots, m, i = 1, \ldots, d\}$$

- Total number of scalar parameters = $m + mdW$

Categorical Naïve Bayes

- Bayes classifier for 0-1 loss (MPE rule = MAP rule):

$$h_{\text{MPE}}(\mathbf{x}) = \arg\max_{y=1,\ldots,m} p(y)p(\mathbf{x}|y,\theta)$$

$$= \arg\max_{y=1,\ldots,m} p(y)\prod_{i=1}^{d}\beta_{x_i,y,i} \quad = \sum_{i=1}^{d} \ln \beta_{x_i}$$

- ML (frequentist) estimate of θ: Let
  - n_y = number of class y training examples
  - $n_{w,y,i}$ = number of class y training examples in which the i-th feature value = w
- then, $\sum_{y=1}^{m} n_y = n$ and for all $i, \sum_{w=1}^{W} n_{w,y,i} = n_y$
- Also, let $n_{w,y} = \sum_{i=1}^{d} n_{w,y,i}$ = number of times w occurs across all d components in class y examples then, $\sum_{w=1}^{W} n_{w,y} = dn_y$

Categorical Naïve Bayes

- ML (frequentist) estimate of θ:

$$\widehat{\theta}_{ML}(\mathcal{D}) = \arg\max_{\theta} \frac{1}{n} \ln p(\mathcal{D}|\theta),$$

$$\theta = (\theta_0, \{\theta_{iy}, i = 1, \ldots, d, y = 1, \ldots, m\})$$

$$\widehat{\theta}_{0,ML}(\mathcal{D}) = \arg\max_{\theta_0} \left[\sum_{y=1}^{m} \left(\frac{n_y}{n} \right) \ln \theta_{0y} \right],$$

$$\forall i, y, \ \widehat{\theta}_{iy,ML}(\mathcal{D}) = \arg\max_{\theta_{iy}} \left[\frac{1}{n_y} \sum_{j:y_j=y} \ln p(x_{ij}|y, \theta_{iy}) \right]$$

$$= \arg\max_{\theta_{iy}} \left[\frac{1}{n_y} \sum_{j:y_j=y} \ln \beta_{x_{ij},y,i} \right]$$

$$= \arg\max_{\theta_{iy}} \left[\sum_{w=1}^{W} \left(\frac{n_{w,y,i}}{n_y} \right) \ln \beta_{w,y,i} \right]$$

Categorical Naïve Bayes

- ML (frequentist) estimate of θ solution:

$$\forall y,\ \widehat{\theta}_{0y,ML} = \widehat{p}(y) = \frac{n_y}{n}$$
$$\forall w, y, i,\ \widehat{\beta}_{w,y,i} = \frac{n_{w,y,i}}{n_y}$$

- If for all w,y,i, $\beta_{w,y,i} = \beta_{w,y}$ then,

$$\forall y,\ \widehat{\theta}_{0y,ML} = \widehat{p}(y) = \frac{n_y}{n}$$
$$\forall w, y, i,\ \widehat{\beta}_{w,y,i} = \frac{\sum_{i=1}^{d} n_{w,y,i}}{\sum_{i=1}^{d} n_y}$$
$$= \frac{n_{w,y}}{d \cdot n_y}$$

Categorical Naïve Bayes

- Solution based on following result
- Result: Let $p_1, \ldots, p_L$, and $q_1, \ldots, q_L$, denote two pmfs over L items, i.e., they are non-negative and sum to one. Then,

$$\arg \max_{q_1,\ldots,q_L} \sum_{l=1}^{L} p_l \ln q_l = \{p_l, l = 1, \ldots, L\}$$

- *Proof*: $\forall t > 0, \ln t \leq t - 1$ with equality, if, and only if (iff), $t = 1$. Replacing t with q_l/p_l , multiplying by p_l , and summing over all l, we get

$$\sum_{l=1}^{L} p_l \ln\left(\frac{q_l}{p_l}\right) \leq \sum_{l=1}^{L} p_l\left(\frac{q_l}{p_l} - 1\right) = \sum_{l=1}^{L}\left(q_l - p_l\right) = 1 - 1 = 0.$$

Alternatively, this follows from the fact that the KL-divergence $D(p||q)$ is non-negative and is zero if, and only if, the two pmfs p, q are identical

Document Classification (2 classes; m=2)

Vocabulary size = $W = 3$ e.g. $\{a, b, c\}$

Training set

class 1 examples

x_1	x_2
a	b
b	c
b	a
c	b
a	c
c	
c	

doc 1 length = 7 doc 2 length = 5

class 2 examples

c	c	c
c	c	b
a	c	c
b	c	a
c		

$L = 5$ $L = 3$ $L = 4$

likelihoods $P(x|y)$

$$\hat{\beta}_{a,1} = \frac{2+1}{7+5} = \frac{3}{12}$$ ← # a's ← total words

$$\hat{\beta}_{b,1} = \frac{2+2}{12} = \frac{4}{12}$$

$$\hat{\beta}_{c,1} = \frac{3+2}{12} = \frac{5}{12}$$

$$\hat{\beta}_{a,2} = \frac{2}{12}$$

$$\hat{\beta}_{b,2} = \frac{2}{12}$$

$$\hat{\beta}_{c,2} = \frac{8}{12}$$

Priors

$$\hat{p}(y=1) = \frac{2}{2+3} = \frac{2}{5}, \quad \hat{p}(y=2) = \frac{3}{2+3} = \frac{3}{5}$$

Test document:

possibly take log probabilities avoid underflow

NB assumption

b
b
a
c
c
a
b
a
c
b d
c
c

$p(b) \mid_1$
$p(b) \mid_1$
$p(a) \mid_1$

$\cdot \mid_2$
$\cdot \mid_2$
$\cdot \mid_2$

$d = 12$

$p(\text{class } 1)$ $\quad$ $p(\text{class } 2)$

$\not{d}$ so don't multiply by 0 if d not in training
OR can do a Bayesian regularization

Laplace Smoothing (default counts)

$$\underset{y}{\text{argmax}} \; \frac{1}{n} \ln p(x, y)$$

where $p(x|y) = \prod_{i=1}^{\text{words}} p(x_i | y)$

$$\hat{p}(y=1) \cdot \hat{p}(\underset{\text{naive bayes}}{x} \mid y=1)$$

$$= \frac{2}{5} \cdot \hat{\beta}_{a,1} \cdot \hat{\beta}_{b,1} \cdot \hat{\beta}_{c,1}$$

$$= \frac{2}{5} \cdot (\hat{\beta}_{a,1})^3 \cdot (\hat{\beta}_{b,1})^4 \cdot (\hat{\beta}_{c,1})^3$$

ML's not $p(x|y)$

just $\max p(x|y)p(y)$

then compare with

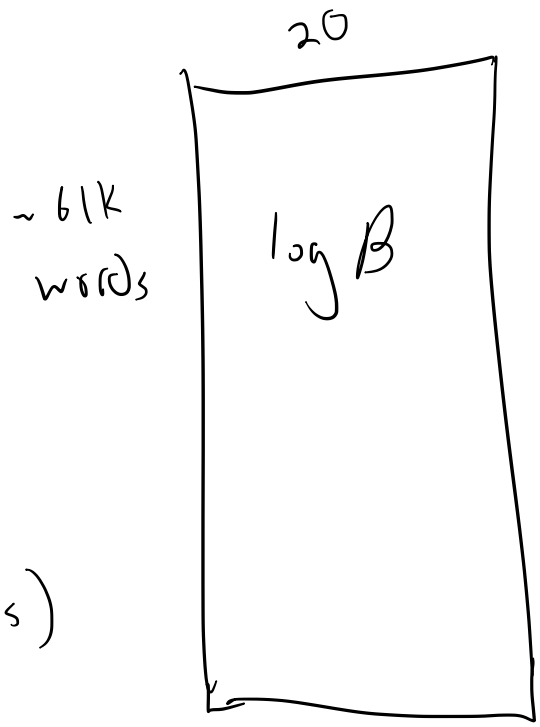
$$\hat{p}(y=2) \cdot \ldots$$

$$= \frac{3}{5} \cdot (\hat{\beta}_{a,2})^3 \cdot (\hat{\beta}_{b,2})^4 (\hat{\beta}_{c,2})^5$$

$p(x_{ij} = w | y_j = y) = \beta_{w,y,j}$

$p(x|y) = \prod_{i=1}^{d} \beta_{x_i,y,i}$

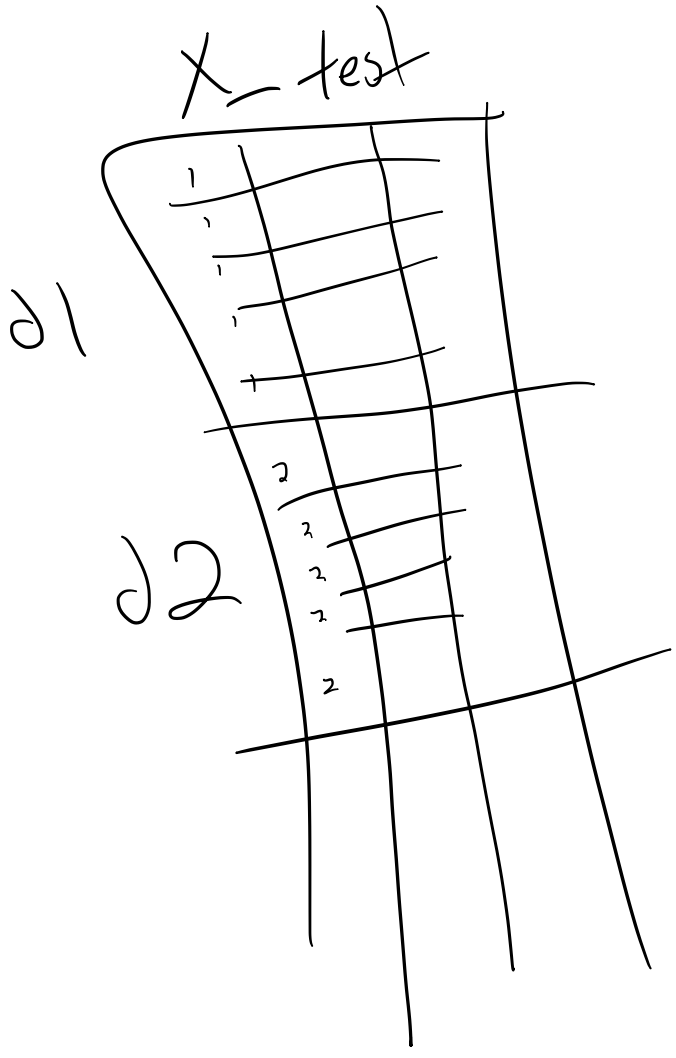
$\log B\left(X\_test(d1, 2), :\right)$.



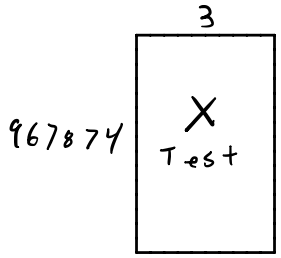
$\cdot \log(\text{likelihoods})$

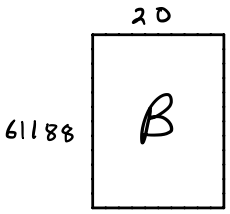
X_test(doc1, 2)

loglikelihood(

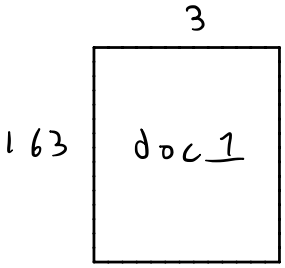


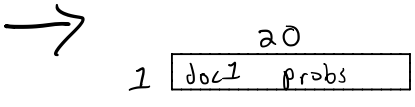
nDocsTest = 7505

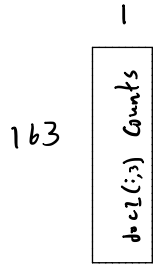




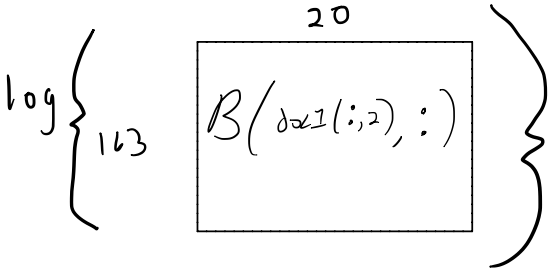
doc1(:,3) = multipliers





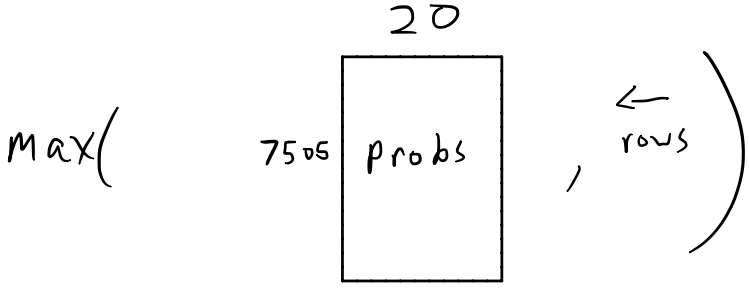


•*



classes

words



Overfitting problem

- If $W >> n$, it is quite likely that there is a value w_0 which never occurs in the training set, but occurs in a test example $\mathbf{x}_{\text{test}}$
- For such a w_0, $n_{w_0,y,i} = \beta_{w_0,y,i} = 0$ for all y, i, and $p\left(\mathbf{x}_{\text{test}} \middle| y, \hat{\theta}_{\text{ML}}\right) = 0, \forall y$, and the decision reduces to random guessing. This ignores information from values that were seen in both training and test sets
- Solution 1: remove words that were not seen during training and proceed as before. Better than random guessing, but still ignores information in new words
- Solution 2: Regularize estimation of β by incorporating prior beliefs via a pdf $\pi(\beta)$.

Bayesian Naïve Bayes with Dirichlet prior

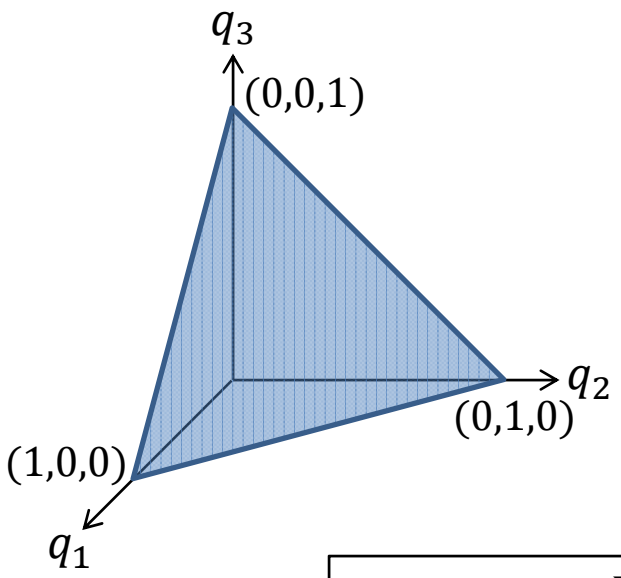
- Dirichlet distribution $\mathrm{Dir}(\mathbf{q}|\boldsymbol{\alpha})$: is a family of continuous multivariate probability distributions parameterized by a W-dimensional vector $\boldsymbol{\alpha}$ of positive reals called the concentration parameters.
- It is a pdf over all pmfs over W values, i.e., a pdf over the $(W\text{-}1)$-dimensional probability simplex:

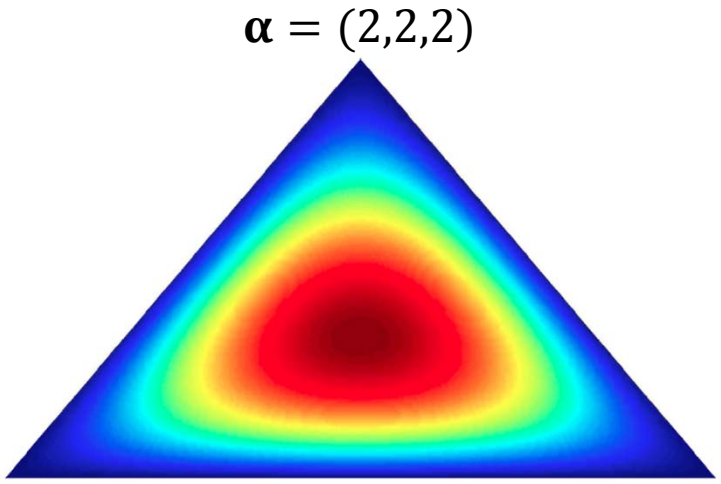
$$\mathcal{S}_W := \left\{ \mathbf{q} : 0 \le q_i \le 1, i = 1, \ldots, W, \sum_{i=1}^{W} q_i = 1 \right\}$$

$$\mathrm{Dir}(\mathbf{q}|\boldsymbol{\alpha}) := \frac{1(\mathbf{q} \in \mathcal{S}_W)}{Z(\boldsymbol{\alpha})} \prod_{i=1}^{W} q_i^{\alpha_i - 1}$$

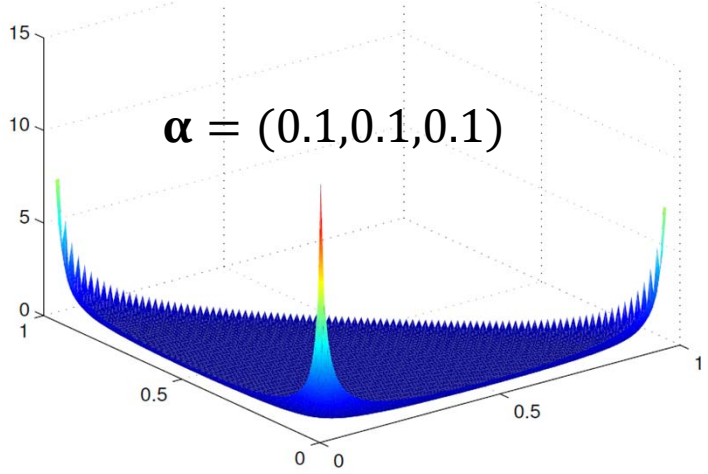
$$Z(\boldsymbol{\alpha}) = \text{normalization constant} = \frac{1}{\Gamma\left(\sum_{i=1}^{W} \alpha_i\right)} \prod_{i=1}^{W} \Gamma(\alpha_i)$$

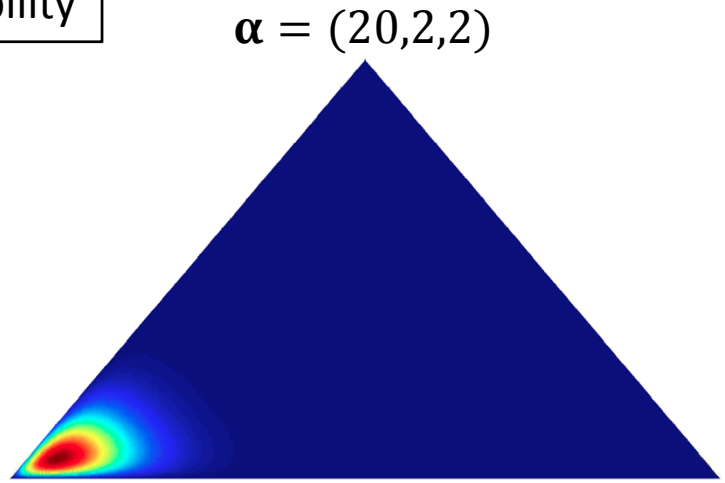
Dirichlet distribution





Red (hotter) ➔ larger probability
Blue (cooler)➔ lower probability





Dirichlet distribution

- $\alpha_0 := \sum_{i=1}^{W} \alpha_i$ controls how peaked the distribution is (larger ⇒ more peaked)
- Dir(1,1,1) is uniform over the probability simplex
- Dir(2,2,2) is a broad distribution centered at (1/3,1/3,1/3)
- Dir(20,20,20) is a narrow distribution centered at (1/3,1/3,1/3)
- If $\alpha_i < 0$ for all i, we get “spikes” at the corners of the probability simplex

Bayesian Naïve Bayes with Dirichlet prior

- Dirichlet Prior for β:

$$\pi(\beta) = \prod_{y=1}^{m} \prod_{i=1}^{d} \frac{1}{Z(\boldsymbol{\alpha}_{yi})} \prod_{w=1}^{W} \left(\beta_{w,y,i}\right)^{\alpha_{w,y,i}-1}$$

- MAP (Bayesian) estimate of θ:

$$\begin{aligned}
\widehat{\theta}_{\mathrm{MAP}} &= \arg\max_{\theta} p(\theta|\mathcal{D}) \\
&= \arg\max_{\theta} p(\mathcal{D}|\theta)\pi(\beta) \\
&= \arg\max_{\theta} \left[\prod_{y=1}^{m} (p(y))^{n_y}\right] \cdot \left[\prod_{w=1}^{W} \prod_{y=1}^{m} \prod_{i=1}^{d} (\beta_{w,y,i})^{n_{w,y,i}+\alpha_{w,y,i}-1}\right]
\end{aligned}$$

Bayesian Naïve Bayes with Dirichlet prior

- MAP (Bayesian) estimate of θ solution:

$$\forall y,\ \widehat{p}(y) = \frac{n_y}{n}$$
$$\forall w, y, i,\ \widehat{\beta}_{w,y,i} = \frac{n_{w,y,i} + \alpha_{w,y,i} - 1}{n_y + \sum_{w=1}^{W} \left(\alpha_{w,y,i} - 1\right)}$$

- If for all w,y,i, $\beta_{w,y,i} = \beta_{w,y}$ then taking $\alpha_{w,y,i} = \alpha_{w,y}$ for all w,y,i, we get

$$\forall y,\ \widehat{p}(y) = \frac{n_y}{n}$$
$$\forall w, y, i,\ \widehat{\beta}_{w,y,i} = \frac{n_{w,y} + \alpha_{w,y} - 1}{d n_y + \sum_{w=1}^{W} \left(\alpha_{w,y} - 1\right)}$$

Remarks

- alphas can be interpreted as “prior” counts and the MAP solution as updating these prior counts with empirical counts from the likelihood
- If all alphas are equal to 2, then the prior counts are equal to one. This is referred to as add-one smoothing of the ML estimate or Laplace’s rule of succession.
- Can also incorporate a separate Dirichlet prior for $p(y)$ in a similar way
- The ML estimates of θ are asymptotically consistent

Example: Binary classification

Y = (lifestyle) Peaceful / Stressed Binary (0, 1)

Feature Vector

Values

$\underline{X} = \begin{pmatrix} 1 & \text{Income} \\ 2 & \text{charitable giving} \\ 3 & \text{Avg \# hrs sleep per day} \end{pmatrix}$ → Low / middle / high ... Ternary, say Categorical over (0, 1, 2)

→ No / Yes ... Binary, say Bernoulli over (0, 1)

→ 2 – 10 ... Real value, say Gaussian (scalar)

$d = 3$

Training data

$Y = 0$ (peaceful) (7)

1	2	2	2	1	0	0	1
2	1	1	1	1	0	0	1
3	9	8	6.5	5	4	3.5	7

$Y = 1$ (stressed) (7)

1	0	2	2	2	1	1	0
2	0	0	0	1	0	0	0
3	4	5	7	4	8	6.5	3

Naive Bayes

$\theta_{0y} = p(y)$ $\quad \hat{p}_{ML}(0) = \frac{7}{14} \leftarrow \frac{n_0}{n} = \hat{p}_{ML}(1)$ (sample mean)

First feature

θ_{1y} = pmf over (0, 1, 2)

For class '0' q_0, q_1, q_2

For class '1' r_0, r_1, r_2

$y=0$ $\quad \hat{q}_{0\,ML} \stackrel{?}{=} \frac{2}{7}$ $\quad \hat{q}_{1\,ML} \stackrel{?}{=} \frac{2}{7}$ $\quad \hat{q}_{2\,ML} \stackrel{?}{=} \frac{3}{7}$

$y=1$ $\quad \hat{r}_{0\,ML} \stackrel{?}{=} \frac{2}{7}$ $\quad \hat{r}_{1\,ML} \stackrel{?}{=} \frac{2}{7}$ $\quad \hat{r}_{2\,ML} \stackrel{?}{=} \frac{3}{7}$

Second feature

θ_{2y} (charity) = pmf over (0, 1)

$\hat{\theta}_{2\,0\,ML} = \left(\frac{2}{7}, \frac{5}{7}\right)$ ('0', '1')
(class 0) peaceful

$\hat{\theta}_{2\,1\,ML} = \left(\frac{6}{7}, \frac{1}{7}\right)$ ('0', '1')
(class 1) Attacked

Feature 3

θ_{3y} (sleep) = for $y=0$, $\theta_{30} = (\mu_0, \sigma_0^2)$

for $y=1$, $\theta_{31} = (\mu_1, \sigma_1^2)$

estimated params.

$\hat{\mu}_{0ML} = \frac{9+8+6.5+\cdots}{7}$

$\hat{\sigma}^2_{0ML} = \cdots$

$\hat{\mu}_{1ML} = \cdots$

$\hat{\sigma}^2_{1ML} = \cdots$

Test individual

$\underline{X}_{test} = \begin{pmatrix} 1 \\ 1 \\ 3 \end{pmatrix}$

$y = ?$

Compare

MAP → ML b.c. $\frac{1}{2}, \frac{1}{2}$ in this case

$y=0$ (p(y))

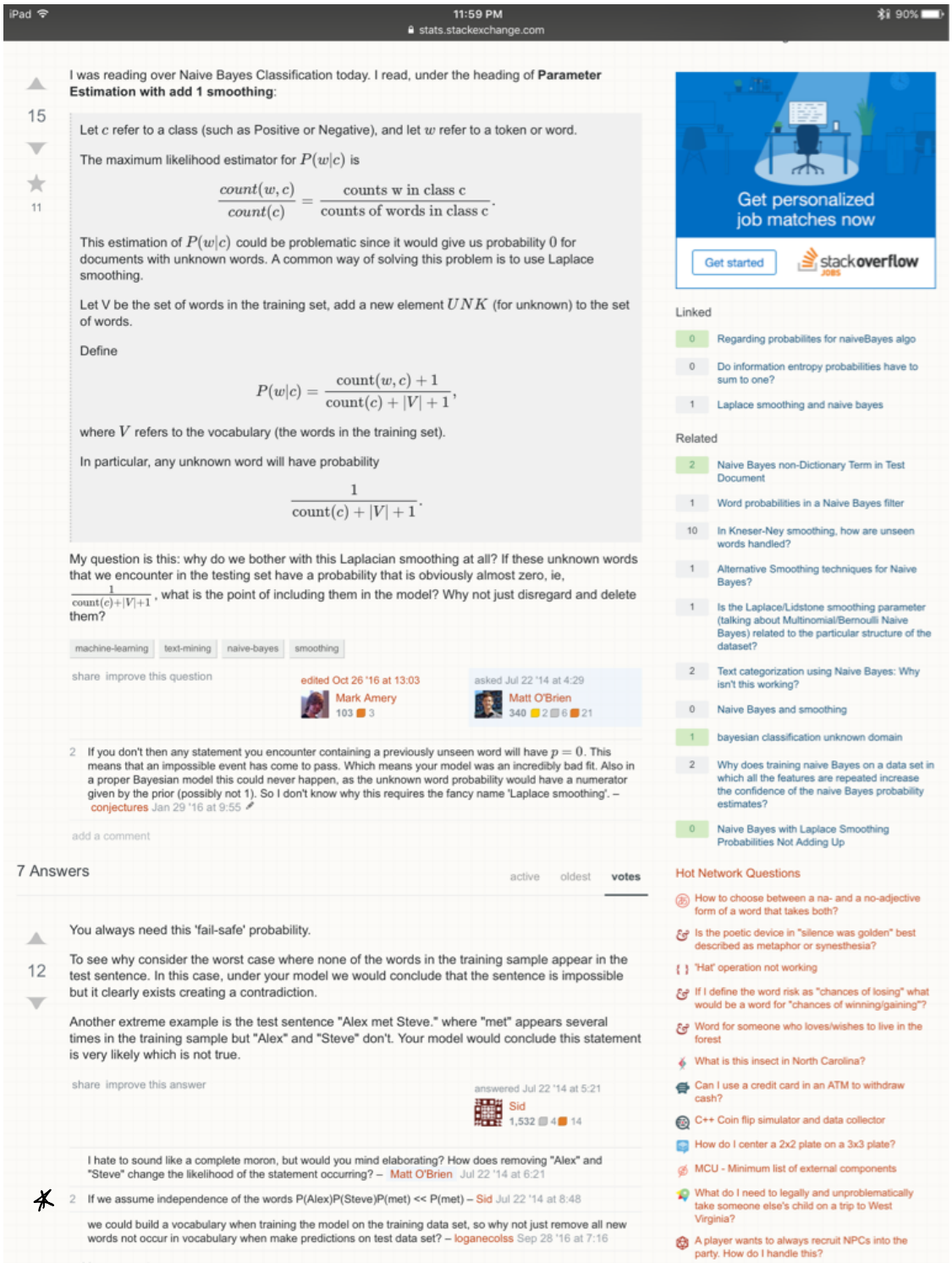
$\frac{1}{2} \cdot \frac{2}{7} \cdot \frac{5}{7} \cdot \mathcal{N}(\hat{\mu}_{0ML}, \hat{\sigma}^2_{0ML})$ evaluated at 3

$y=1$

$\frac{1}{2} \cdot \frac{2}{7} \cdot \frac{1}{7} \cdot \mathcal{N}(\hat{\mu}_{1ML}, \sigma^2_{1ML})(3)$

$\gtrless$ ($y=0$ / $y=1$)

Why Laplace smoothing?



iPad 11:59 PM 90%
stats.stackexchange.com

15 11

I was reading over Naive Bayes Classification today. I read, under the heading of **Parameter Estimation with add 1 smoothing**:

> Let c refer to a class (such as Positive or Negative), and let w refer to a token or word.
>
> The maximum likelihood estimator for $P(w|c)$ is
>
> $$\frac{count(w,c)}{count(c)} = \frac{\text{counts w in class c}}{\text{counts of words in class c}}.$$
>
> This estimation of $P(w|c)$ could be problematic since it would give us probability 0 for documents with unknown words. A common way of solving this problem is to use Laplace smoothing.
>
> Let V be the set of words in the training set, add a new element UNK (for unknown) to the set of words.
>
> Define
>
> $$P(w|c) = \frac{\text{count}(w,c)+1}{\text{count}(c)+|V|+1},$$
>
> where V refers to the vocabulary (the words in the training set).
>
> In particular, any unknown word will have probability
>
> $$\frac{1}{\text{count}(c)+|V|+1}.$$

My question is this: why do we bother with this Laplacian smoothing at all? If these unknown words that we encounter in the testing set have a probability that is obviously almost zero, ie, $\frac{1}{\text{count}(c)+|V|+1}$, what is the point of including them in the model? Why not just disregard and delete them?

machine-learning text-mining naive-bayes smoothing

share improve this question

edited Oct 26 '16 at 13:03 Mark Amery 103 3

asked Jul 22 '14 at 4:29 Matt O'Brien 340 2 6 21

2 If you don't then any statement you encounter containing a previously unseen word will have $p = 0$. This means that an impossible event has come to pass. Which means your model was an incredibly bad fit. Also in a proper Bayesian model this could never happen, as the unknown word probability would have a numerator given by the prior (possibly not 1). So I don't know why this requires the fancy name 'Laplace smoothing'. – conjectures Jan 29 '16 at 9:55

add a comment

7 Answers

active oldest votes

12

You always need this 'fail-safe' probability.

To see why consider the worst case where none of the words in the training sample appear in the test sentence. In this case, under your model we would conclude that the sentence is impossible but it clearly exists creating a contradiction.

Another extreme example is the test sentence "Alex met Steve." where "met" appears several times in the training sample but "Alex" and "Steve" don't. Your model would conclude this statement is very likely which is not true.

share improve this answer

answered Jul 22 '14 at 5:21 Sid 1,532 4 14

I hate to sound like a complete moron, but would you mind elaborating? How does removing "Alex" and "Steve" change the likelihood of the statement occurring? – Matt O'Brien Jul 22 '14 at 6:21

2 If we assume independence of the words P(Alex)P(Steve)P(met) << P(met) – Sid Jul 22 '14 at 8:48

we could build a vocabulary when training the model on the training data set, so why not just remove all new words not occur in vocabulary when make predictions on test data set? – loganecolss Sep 28 '16 at 7:16

add a comment

Linked

0 Regarding probabilites for naiveBayes algo

0 Do information entropy probabilities have to sum to one?

1 Laplace smoothing and naive bayes

Related

2 Naive Bayes non-Dictionary Term in Test Document

1 Word probabilities in a Naive Bayes filter

10 In Kneser-Ney smoothing, how are unseen words handled?

1 Alternative Smoothing techniques for Naive Bayes?

1 Is the Laplace/Lidstone smoothing parameter (talking about Multinomial/Bernoulli Naive Bayes) related to the particular structure of the dataset?

2 Text categorization using Naive Bayes: Why isn't this working?

0 Naive Bayes and smoothing

1 bayesian classification unknown domain

2 Why does training naive Bayes on a data set in which all the features are repeated increase the confidence of the naive Bayes probability estimates?

0 Naive Bayes with Laplace Smoothing Probabilities Not Adding Up

Hot Network Questions

- How to choose between a na- and a no-adjective form of a word that takes both?
- Is the poetic device in "silence was golden" best described as metaphor or synesthesia?
- 'Hat' operation not working
- If I define the word risk as "chances of losing" what would be a word for "chances of winning/gaining"?
- Word for someone who loves/wishes to live in the forest
- What is this insect in North Carolina?
- Can I use a credit card in an ATM to withdraw cash?
- C++ Coin flip simulator and data collector
- How do I center a 2x2 plate on a 3x3 plate?
- MCU - Minimum list of external components
- What do I need to legally and unproblematically take someone else's child on a trip to West Virginia?
- A player wants to always recruit NPCs into the party. How do I handle this?