Naïve Bayes

In Class Quiz-3

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality ??

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

- Step 1: How many classes in Training Data
  - 2 {+,-}
- Step 2: What are the probability of these classes?
  - Count how many training samples are there: 5
  - Count how many training samples are + : 2
  - Count how many training samples are - : 3

$$P(-) = \frac{3}{5} \qquad P(+) = \frac{2}{5}$$

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

- Step 3: Count how many tokens(=words) in each class
  - N(+) = 9
  - N(-) = 14
- Step 4: Count vocabulary size
  - how many unique tokens in the full training data
  - |V| = 20

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

- Step 5: For each word ‘w’ in Test data find the probability of w appearing in +/-
  [use the training data to find this probability]

$$p(\text{“}predictable\text{”}|-) = \frac{count(\text{“}predictable\text{”}, -) + 1}{N(-) + |V|}$$

  - N(-) = total number of tokens in “-”
  - |V| = vocabulary size = Number of unique tokens in the full training data

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

- Step 5: For each word 'w' in Test data find the probability of w appearing in +/-
  [use the training data to find this probability]

$$p(\text{“}predictable\text{”}|-) = \frac{count(\text{“}predictable\text{”}, -) + 1}{N(-) + |V|}$$

- N(-) = total number of tokens in "-"
- |V| = vocabulary size = Number of unique tokens in the full training data

$$p(\text{“}predictable\text{”}|-) = \frac{1+1}{N(-) + |V|} = \frac{1+1}{14 + |V|} = \frac{1+1}{14 + 20}$$

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

$$P(\text{“predictable”}|-) = \frac{1+1}{14+20}$$
$$P(\text{“with”}|-) = \frac{0+1}{14+20}$$
$$P(\text{“no”}|-) = \frac{1+1}{14+20}$$
$$P(\text{“originality”}|-) = \frac{0+1}{14+20}$$

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

$$P(\text{“predictable”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“with”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“no”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“originality”}|+) = \frac{0+1}{9+20}$$

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

- Step 6: Using the probabilities calculated in step 2, 5, find the most probable class for the test samples
  - S = Test sentence = { "predictable with no originality"}

  $P(-|S), P(+|S)$

  $P(-|S) \propto P(S|-)P(-)$

  $P(+|S) \propto P(S|+)P(+)$

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

- Step 6: Using the probabilities calculated in step 2, 5, find the most probable class for the test samples
  - S = Test sentence = { “predictable with no originality”}

$$P(-|S), P(+|S)$$

$$P(-|S) \propto P(S|-)P(-)$$

$$P(+|S) \propto P(S|+)P(+)$$

$$P(S|-) = P(x_1, ..., x_n|-) = P(w_1, w_2, w_3, w_4|-) = P(w_1|-)P(w_2|-)P(w_3|-)P(w_4|-)$$

$$w_1 = \text{“}predictable\text{”}, w_2 = \text{“}with\text{”}, \text{‘}w_3 = \text{‘}no\text{”}, w_4 = \text{“}orginality\text{”}$$

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

- Step 6: Using the probabilities calculated in step 2, 5, find the most probable class for the test samples
  - S = Test sentence = { "predictable with no originality"}

$$P(-|S), P(+|S)$$

$$P(-|S) \propto P(S|-)P(-)$$

$$P(+|S) \propto P(S|+)P(+)$$

$$P(S|-) = P(x_1, ..., x_n|-) = P(w_1, w_2, w_3, w_4|-) = P(w_1|-)P(w_2|-)P(w_3|-)P(w_4|-)$$

$$P(S|+) = P(x_1, ..., x_n|+) = P(w_1, w_2, w_3, w_4|+) = P(w_1|+)P(w_2|+)P(w_3|+)P(w_4|+)$$

$$w_1 = \text{``}predictable\text{''}, w_2 = \text{``}with\text{''}, \text{`}w_3 = \text{`}no\text{''}, w_4 = \text{``}orginality\text{''}$$

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

- Step 6: Using the probabilities calculated in step 2, 5, find the most probable class for the test samples

$$P(S|-)P(-) = P(\text{“}predictable\text{”}|-)P(\text{“}with\text{”}|-)P(\text{“}no|-)\text{”}P(\text{“}orginality\text{”}|-)P(-)$$

$$P(S|-)P(-) = \frac{2}{34} \times \frac{1}{34} \times \frac{2}{34} \times \frac{1}{34} \times \frac{3}{5}$$

$$P(\text{“predictable”}|-) = \frac{1+1}{14+20}$$

$$P(\text{“with”}|-) = \frac{0+1}{14+20}$$

$$P(\text{“no”}|-) = \frac{1+1}{14+20}$$

$$P(\text{“originality”}|-) = \frac{0+1}{14+20}$$

$$P(-) = \frac{3}{5}$$

	Category	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprise and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no originality

$$P(S|-)P(-) = \frac{3}{5} \times \frac{2 \times 1 \times 2 \times 1}{34^4} = 1.8 \times 10^{-6}$$

$$P(S|+)P(+) = \frac{2}{5} \times \frac{1 \times 1 \times 1 \times 1}{29^4} = 5.7 \times 10^{-7}$$

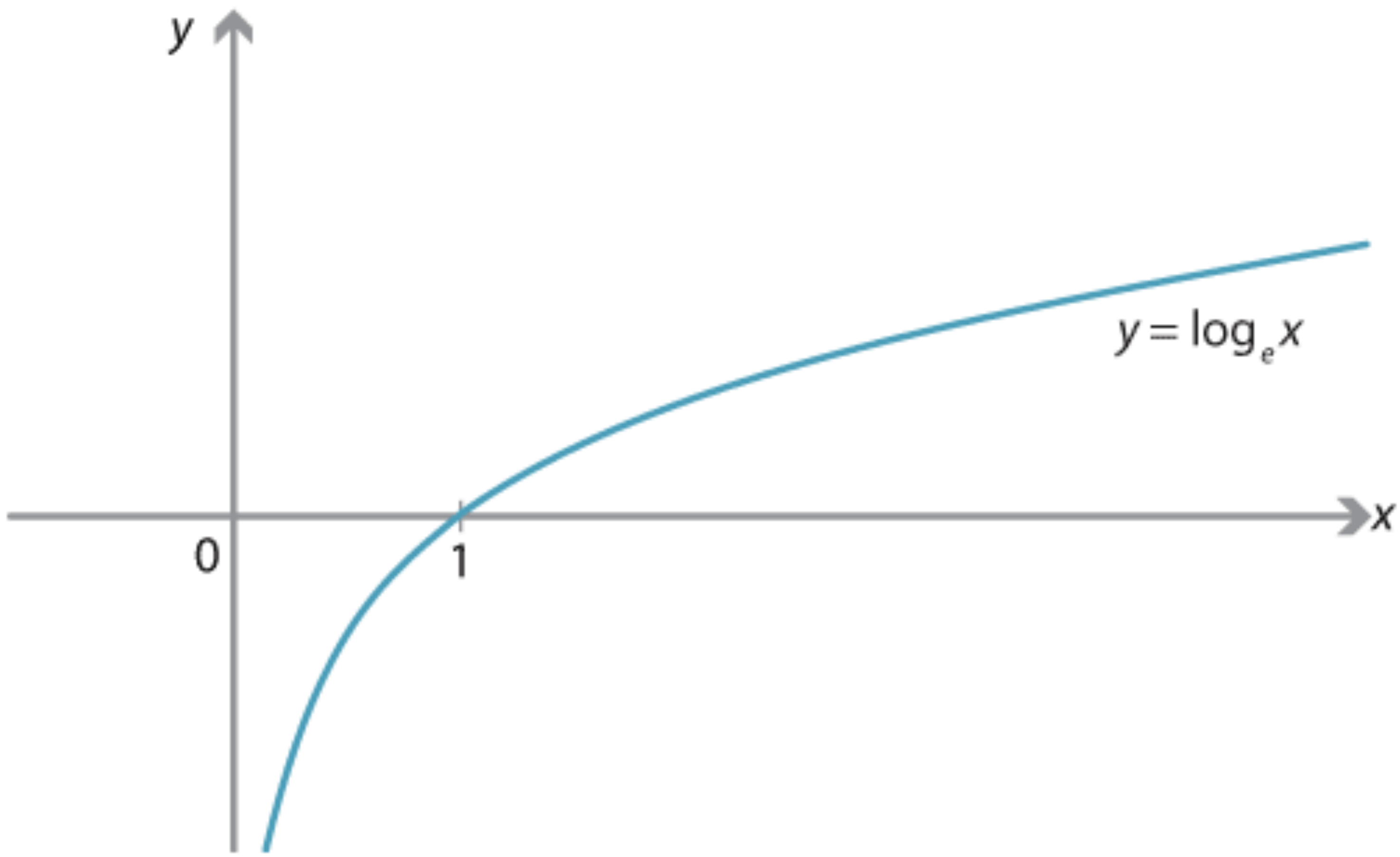
The model thus predicts the class *negative* for the test sentence.

Naïve Bayes Classification: Practical Issues

$$
\begin{aligned}
c_{MAP} &= \mathrm{argmax}_c P(c|x_1,\ldots,x_n) \\
&= \mathrm{argmax}_c P(x_1,\ldots,x_n|c)P(c) \\
&= \mathrm{argmax}_c P(c)\prod_{i=1}^{n} P(x_i|c)
\end{aligned}
$$

- Multiplying together lots of probabilities
- Probabilities are numbers between 0 and 1
- Q: What could go wrong here?

Working with probabilities in log space



$$log_2(1) = 0$$

$$log_2(.00000001) = -26.5754$$

Log Identities (review)

$$\log(a \times b) = \log(a) + \log(b)$$

$$\log(\frac{a}{b}) = \log(a) - \log(b)$$

$$\log(a^n) = n\log(a)$$

$$exp(log(x)) = x$$

Naïve Bayes with Log Probabilities

$$
\begin{aligned}
c_{MAP} &= \text{argmax}_c P(c|x_1, \ldots, x_n) \\
&= \text{argmax}_c P(c) \prod_{i=1}^{n} P(x_i|c) \\
&= \text{argmax}_c \log \left(P(c) \prod_{i=1}^{n} P(x_i|c) \right) \\
&= \text{argmax}_c \log P(c) + \sum_{i=1}^{n} \log P(x_i|c)
\end{aligned}
$$

Naïve Bayes with Log Probabilities

$$c_{MAP} = \operatorname{argmax}_c \log P(c) + \sum_{i=1}^{n} \log P(x_i|c)$$

We do not have to worry about floating point underflow anymore

What if we want to calculate posterior log-probabilities?

$$P(c|x_1,\ldots,x_n) = \frac{P(c)\prod_{i=1}^{n}P(x_i|c)}{\sum_{c'}P(c')\prod_{i=1}^{n}P(x_i|c')}$$

$$\log P(c|x_1,\ldots,x_n) = \log\frac{P(c)\prod_{i=1}^{n}P(x_i|c)}{\sum_{c'}P(c')\prod_{i=1}^{n}P(x_i|c')}$$

$$= \log P(c) + \sum_{i=1}^{n}P(x_i|c) - \boxed{\log\left[\sum_{c'}P(c')\prod_{i=1}^{n}P(x_i|c')\right]}$$

But there is no log identity for summation

What if we want to calculate posterior log-probabilities?

$$log(\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c'))$$

What if we want to calculate posterior log-probabilities?

$$log(\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c'))$$

$$= log(\sum_{c'} (exp(log(P(c') \prod_{i=1}^{n} P(x_i|c'))))$$

$$\boxed{exp(log(x)) = x}$$

What if we want to calculate posterior log-probabilities?

$$log(\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c'))$$

$$= log(\sum_{c'} (exp(log(P(c') \prod_{i=1}^{n} P(x_i|c'))))$$

$$= log(\sum_{c'} exp(b_{c'}))$$

$$\boxed{b_{c'} = log(P(c') \prod_{i=1}^{n} P(x_i|c'))}$$

What if we want to calculate posterior log-probabilities?

$$log(\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c'))$$

$$= log(\sum_{c'} (exp(log(P(c') \prod_{i=1}^{n} P(x_i|c'))))$$

$$= log(\sum_{c'} exp(b_{c'}))$$

$$= log[\sum_{c'} exp(b_{c'})(exp(B - B)]$$

$$\boxed{exp(B - B) = exp(0) = 1}$$

What if we want to calculate posterior log-probabilities?

$$log(\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c'))$$

$$= log(\sum_{c'} (exp(log(P(c') \prod_{i=1}^{n} P(x_i|c'))))$$

$$= log(\sum_{c'} exp(b_{c'}))$$

$$= log[\sum_{c'} exp(b_{c'})(exp(B - B)]$$

$$= log[\sum_{c'} exp(b_{c'} - B) exp(B)]$$

What if we want to calculate posterior log-probabilities?

$$log(\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c'))$$

$$= log(\sum_{c'}(exp(log(P(c') \prod_{i=1}^{n} P(x_i|c'))))$$

$$= log(\sum_{c'} exp(b_{c'}))$$

$$= log[\sum_{c'} exp(b_{c'})(exp(B - B)]$$

$$= log[\sum_{c'} exp(b_{c'} - B)exp(B)]$$

$$= log[(\sum_{c'} expb_{c'})(exp(B))]$$

What if we want to calculate posterior log-probabilities?

$$log(\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c'))$$

$$= log(\sum_{c'}(exp(log(P(c') \prod_{i=1}^{n} P(x_i|c'))))$$

$$= log(\sum_{c'} exp(b_{c'}))$$

$$= log[\sum_{c'} exp(b_{c'})(exp(B - B)]$$

$$= log[\sum_{c'} exp(b_{c'} - B)exp(B)]$$

$$= log[(\sum_{c'} expb_{c'})(exp(B))]$$

$$= log[(\sum_{c'} expb_{c'})] + log[(exp(B))]$$

What if we want to calculate posterior log-probabilities?

$$log(\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c'))$$

$$= log(\sum_{c'} (exp(log(P(c') \prod_{i=1}^{n} P(x_i|c'))))$$

$$= log(\sum_{c'} exp(b_{c'}))$$

$$= log[\sum_{c'} exp(b_{c'})(exp(B - B)]$$

$$= log[\sum_{c'} exp(b_{c'} - B) exp(B)]$$

$$= log[(\sum_{c'} exp b_{c'})(exp(B))]$$

$$= log[(\sum_{c'} exp b_{c'})] + log[(exp(B))]$$

$$= log[(\sum_{c'} exp b_{c'})] + B$$

What if we want to calculate posterior log-probabilities?

$$log(\sum_{c'} P(c') \prod_{i=1}^{n} P(x_i|c'))$$

$$= log(\sum_{c'} (exp(log(P(c') \prod_{i=1}^{n} P(x_i|c'))))$$

$$= log(\sum_{c'} exp(b_{c'}))$$

$$= log[\sum_{c'} exp(b_{c'})(exp(B - B)]$$

$$= log[\sum_{c'} exp(b_{c'} - B)exp(B)]$$

$$= log[(\sum_{c'} expb_{c'})(exp(B))]$$

$$= log[(\sum_{c'} expb_{c'})] + log[(exp(B))]$$

$$\boxed{B = max_{c'} b_{c'}} \quad = log[(\sum_{c'} expb_{c'})] + B$$

Log Exp Sum Trick:

$$\log[\sum_i \exp(x_i)] = x_{max} + \log[\sum_i \exp(x_i - x_{max})]$$

Another issue: Smoothing

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

Another issue: Smoothing

Alpha doesn't necessarily need to be 1 (hyperparmeter)

$$\hat{P}(w_i|c) = \frac{\mathrm{count}(w,c) + \alpha}{\sum_{w' \in V} \mathrm{count}(w',c) + \alpha|V|}$$

Another issue: Smoothing

Can think of alpha as a “pseudocount”.
Imaginary number of times this word has been seen.

$$\hat{P}(w_i|c) = \frac{\text{count}(w,c) + \alpha}{\sum_{w' \in V} \text{count}(w',c) + \alpha|V|}$$

Another issue: Smoothing

$$\hat{P}(w_i|c) = \frac{\text{count}(w,c) + \alpha}{\sum_{w' \in V} \text{count}(w',c) + \alpha|V|}$$

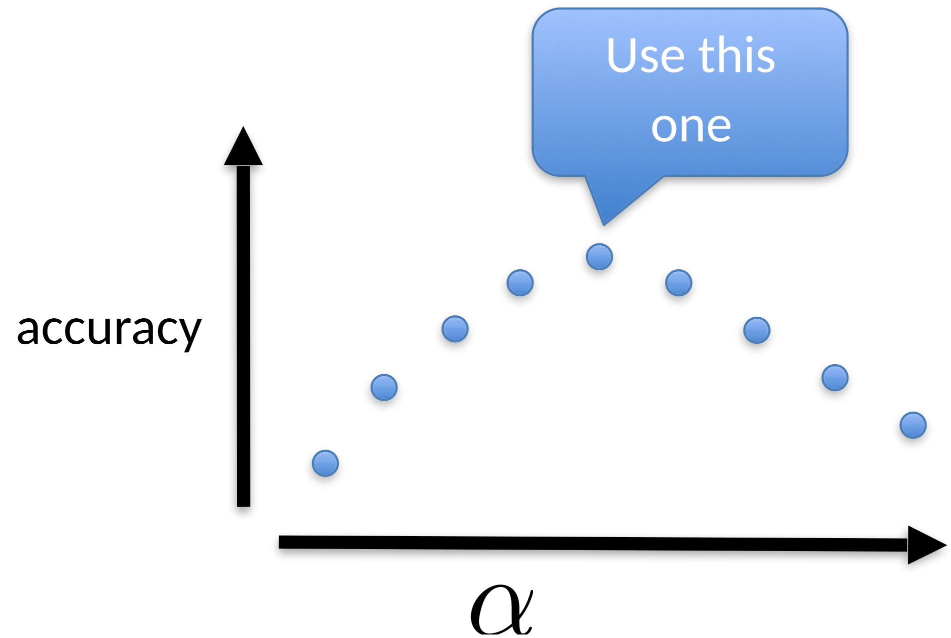
- Q: What if alpha = 0?
- Q: what if alpha = 0.000001?
- Q: what happens as alpha gets very large?

Overfitting

- Model cares too much about the training data
- How to check for overfitting?
  - Training vs. test accuracy
- Pseudocount parameter combats overfitting

Q: how to pick Alpha?

- Split train vs. Test
- Try a bunch of different values
- Pick the value of alpha that performs best
- What values to try? Grid search
  - (10^-2,10^-1,...,10^2)



Data Splitting

- Train vs. Test

- Better:
  - Train (used for fitting model **parameters**)
  - Dev (used for tuning **hyperparameters**)
  - Test (reserve for final evaluation)

- Cross-validation

Feature Engineering

- What is your word / feature representation
  - Tokenization rules: splitting on whitespace?
  - Uppercase is the same as lowercase?
  - Numbers?
  - Punctuation?
  - Stemming?