

COMP4901K/Math4824B

Machine Learning for Natural Language Processing

Lecture 7: Naïve Bayes

Instructor: Yangqiu Song

A General Pipeline of Building a Classifier

Pre-processing data

Tokenization

Stemming/
normalization

N-gram

Stopwords
filtering

Preparing input data for machine learning

Feature construction
(e.g., VSM)

Feature construction
(e.g., word embeddings)

Feature selection
(e.g., DF filtering)

Configuration of Data and Metrics for Test

Training/Dev/Testing

K-Fold Cross Validation

Evaluation Metrics

Model Specification and Selection

Specify models based
on assumptions

Evaluate model based
on C.V. and metrics

Output the model for
future use

Today's lecture

- k nearest neighbors
- Naïve Bayes classifier

Example: Spam Filter

- Input: an email
- Output: spam/not spam
- Setup:
 - Get a large collection of example emails, each labeled “spam” or “not spam”
 - Note: **someone has to hand label all this data!**
 - Want to learn to predict labels of new, future emails
- Features: The attributes used to make the spam/not spam decision
 - Words: FREE!
 - Text Patterns: \$dd, CAPS
 - Non-text: SenderInContacts
 - ...



Dear Sir.

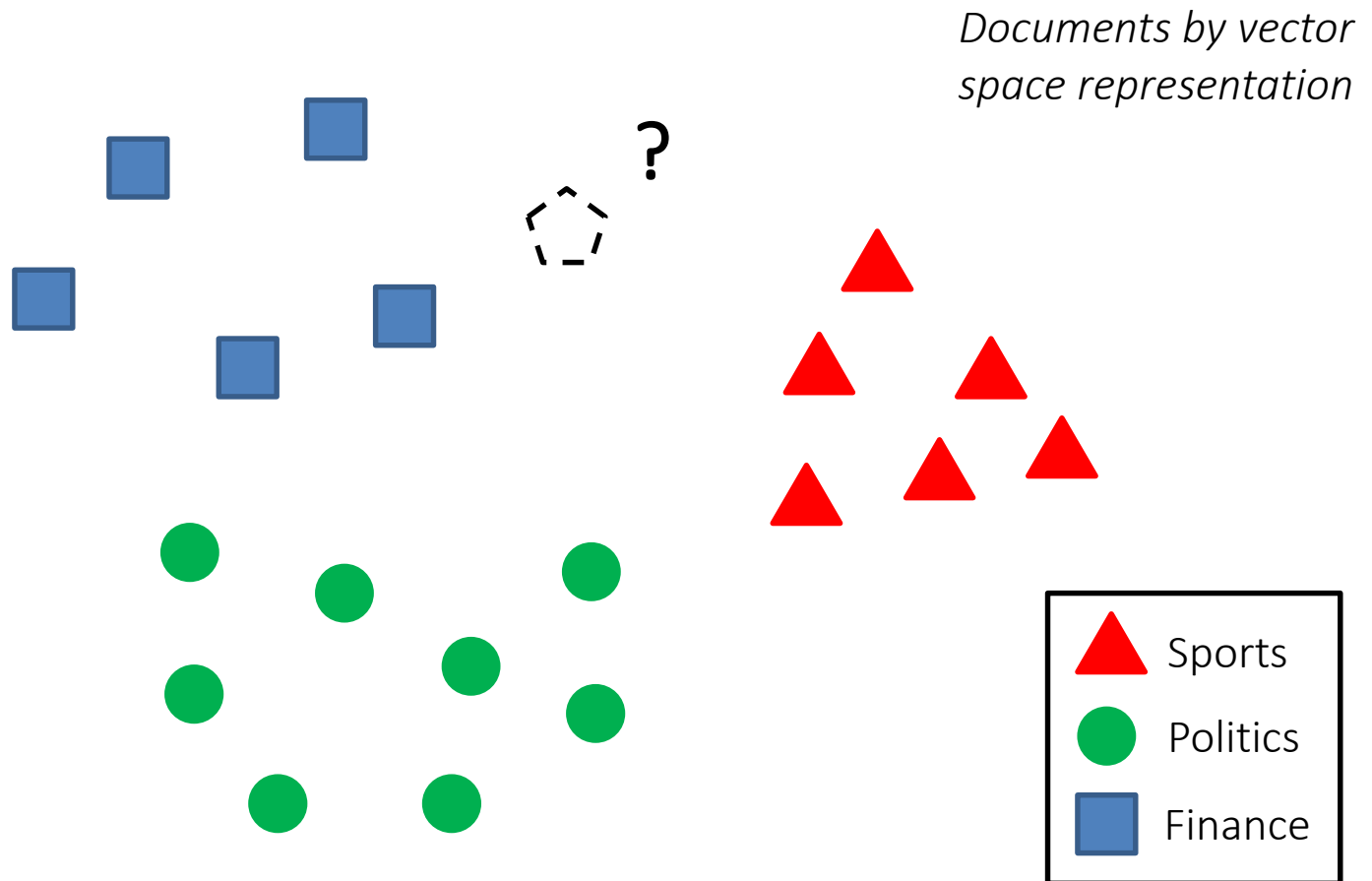
First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret.
...

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

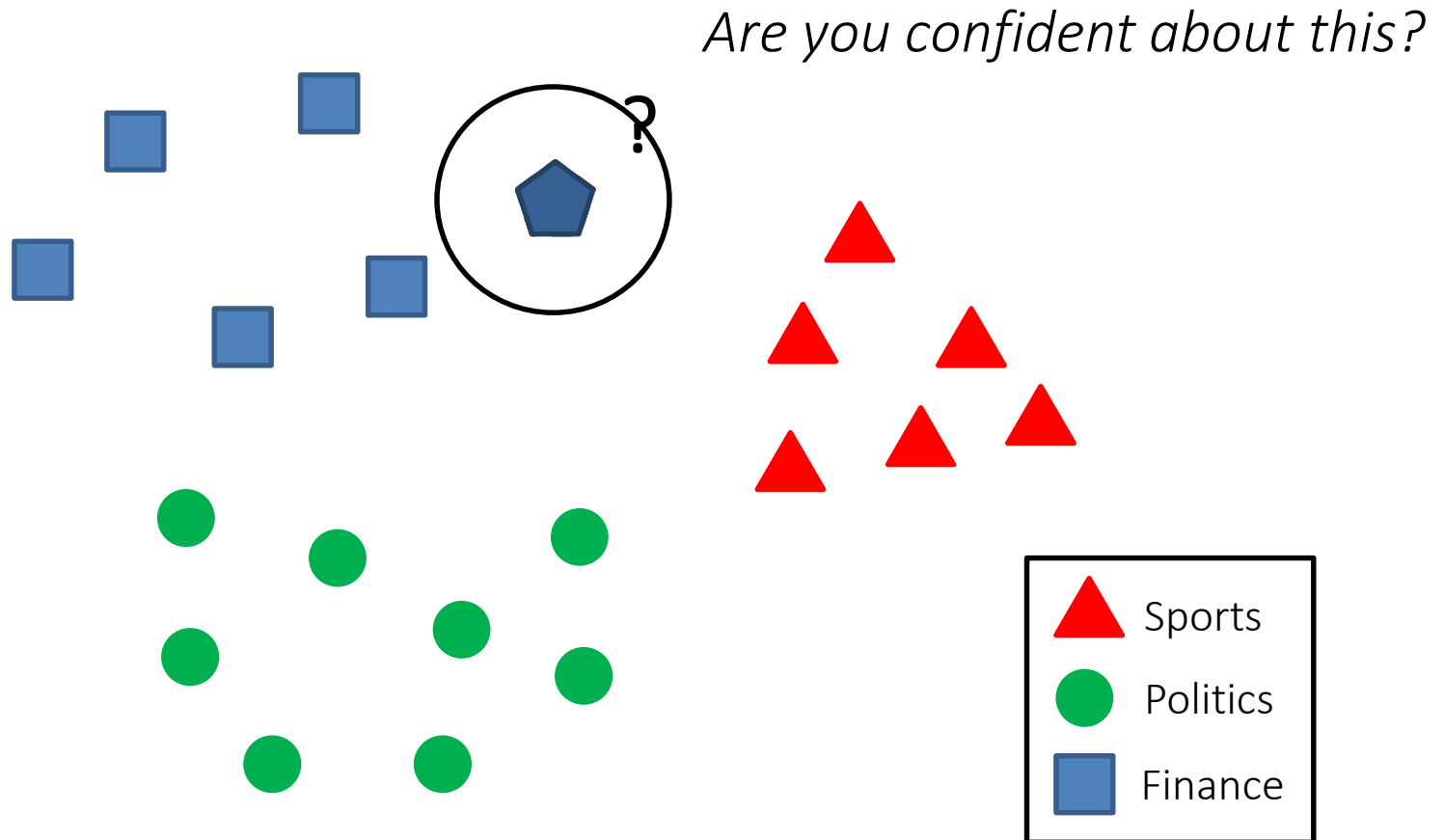
99 MILLION EMAIL ADDRESSES
FOR ONLY \$99

Ok, I know this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

How to classify this document?

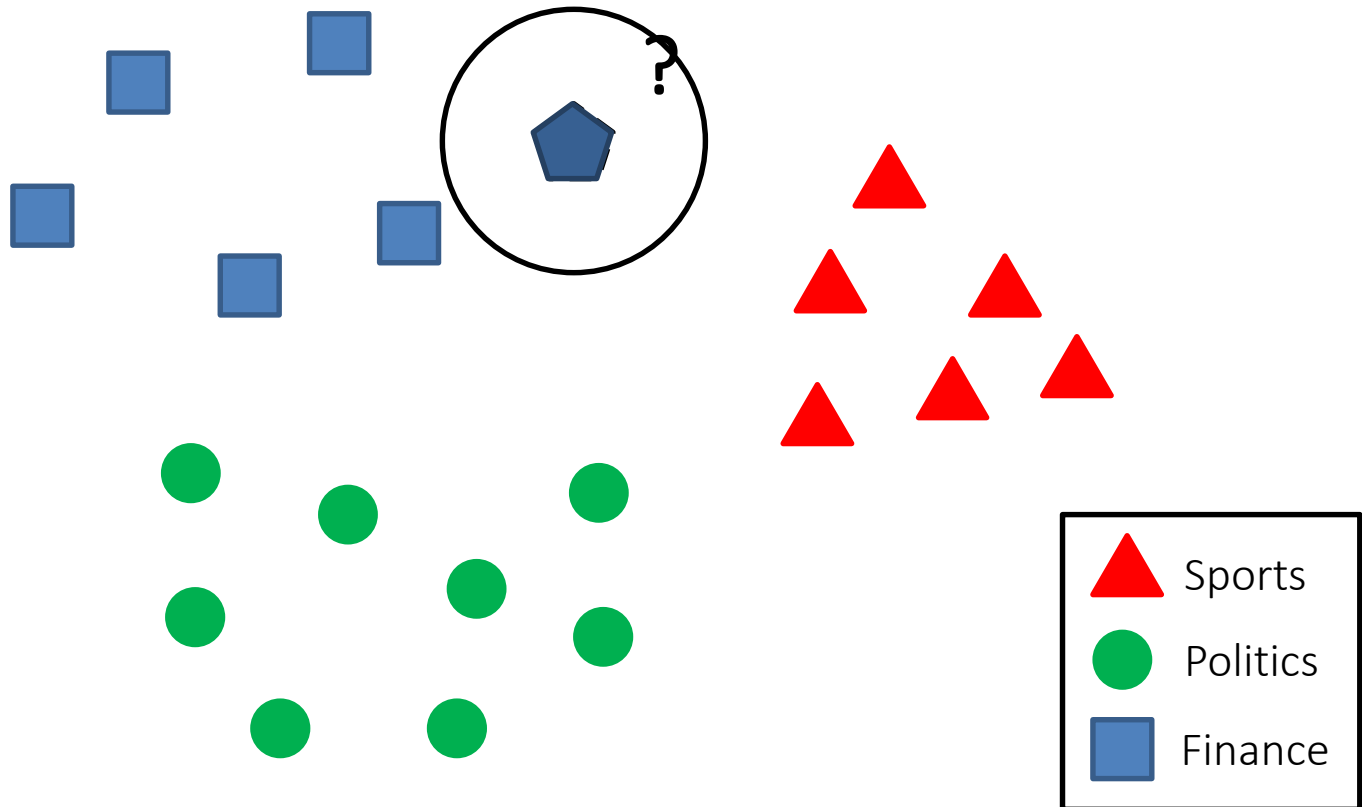


Let's check the nearest neighbor



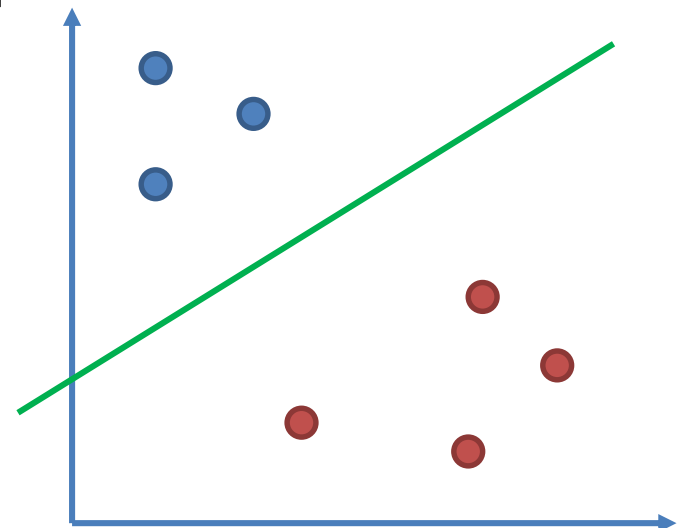
Let's check more nearest neighbors

- Ask k nearest neighbors
 - Let them vote



Problems with k nearest neighbors

- Memorize all instances (documents here)
- Perform search over large database
 - High-dimensional data
- Can we do it in a simpler way?
 - Summarizing the classification using a simpler function



Basic notions about a classifier

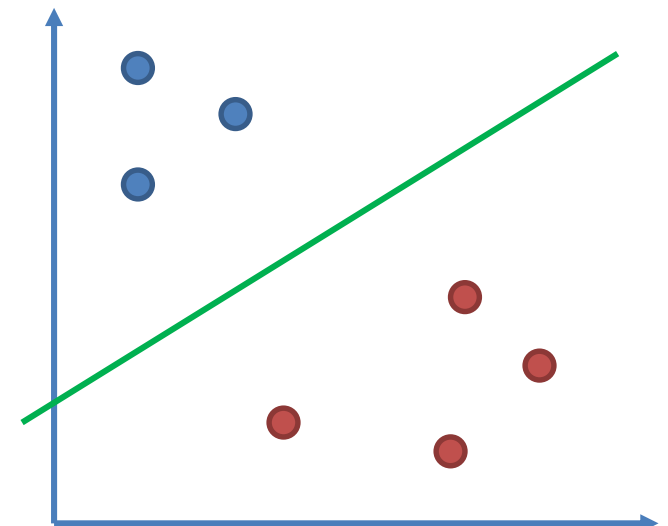
- Data points/Instances
 - A document $d = \{w_1, w_2, \dots, w_{M_d}\}$
 - A data instance $\mathbf{x} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(V)}]$: a V -dimensional feature vector
 - A vocabulary $\mathbf{x} = \{x_1, x_2, \dots, x_V\}$
- Labels
 - y : a categorical value from $\{1, \dots, K\}$
- Classification hyper-plane
 - $f(\mathbf{x}) \rightarrow y$

Key question: how to find such a mapping?

Sequence of tokens

Vector space representation

Index of features



Bayes' Rule

- Two ways to factor a joint distribution over two variables:

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x)$$

That's my rule!

- Dividing, we get:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

- Why is this at all helpful?
 - Lets us build **one conditional** from **its reverse**
 - Often one conditional is tricky but the other one is simple
- In the running for most important AI equation!



Bayes, Thomas (1763) An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370-418

Naïve Bayes Assumption and Classifier

- Naïve Bayes: Assume all features are independent effects of the label

$$\begin{aligned} & P\left(y = k, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(V)}\right) \\ &= P(y = k)P\left(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(V)} | Y = k\right) \\ &= P(y = k) \prod_j^V P\left(\mathbf{x}^{(j)} | y = k\right) \end{aligned}$$

- For prediction, we use

$$P\left(y = k | \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(V)}\right) = \frac{P\left(y = k, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(V)}\right)}{P\left(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(V)}\right)}$$

Number of Parameters

- A general Naive Bayes model:

$$P\left(y = k, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(V)}\right) = P(Y = k) \prod_j^V P\left(\mathbf{x}^{(j)} | y = k\right)$$

$|Y|$ parameters

$|Y| * |X|^V$ values

$V * |X| * |Y|$ parameters

- We only have to specify how each feature depends on the class
- Total number of parameters is *linear* in V
- Model is very simplistic, but often works anyway

Number of Parameters

- A general Naive Bayes model:

$$P\left(y = k, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(V)}\right) = P(Y = k) \prod_j^V P\left(\mathbf{x}^{(j)} | y = k\right)$$

$|Y|$ parameters

$|Y| * |X|^V$ values

$V * |X| * |Y|$ parameters

General Naïve Bayes

- What do we need in order to use Naïve Bayes?
 - **Inference method** (we just saw this part)
 - Start with a bunch of probabilities: $P(y = k)$ and the $P(\mathbf{x}^{(j)} | y = k)$ tables
 - Use standard inference to compute $P(y = k | \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(V)})$
 - Nothing new here
 - **Estimates** of local conditional probability tables
 - $P(y)$, the prior over labels
 - $P(\mathbf{x}^{(j)} | y = k)$ for each feature (evidence variable)
 - These probabilities are collectively called the **parameters** of the model and denoted by θ
 - Up until now, we assumed these appeared by magic, but...
 - ...they typically **come from training data counts**: we'll look at this soon (learning)

Naïve Bayes for Text

- Bag-of-words Naïve Bayes:

- Features: w_i is the word at position i
- As before: predict label conditioned on feature variables (spam vs. not spam)
- As before: assume features are conditionally independent given label
- New: each w_i is identically distributed

$$\begin{aligned} & P(y = k, w_1, \dots, w_M) \\ &= P(y = k)P(w_1, \dots, w_M | y = k) \\ &= P(y = k) \prod_i^M P(w_i | y = k) \\ &= P(y = k) \prod_j^V P(x_j | y = k)^{c(x_j)} \\ &= P(y = k) \prod_j^V P(x_j | y = k)^{x^{(j)}} \end{aligned}$$

- Data points/Instances

- A document $d = \{w_1, w_2, \dots, w_{M_d}\}$
- A data instance $\mathbf{x} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(V)}]$: a V -dimensional feature vector
- A vocabulary $\mathbf{x} = \{x_1, x_2, \dots, x_V\}$

Change the index
from word tokens
to word types

$$\begin{aligned}
& P(y = k, w_1, \dots, w_M) \\
&= P(y = k)P(w_1, \dots, w_M | y = k) \\
&= P(y = k) \prod_i^M P(w_i | y = k) \\
&= P(y = k) \prod_j^V P(x_j | y = k)^{c(x_j)} \\
&= P(y = k) \prod_j^V P(x_j | y = k)^{x_j}
\end{aligned}$$

- Data points/Instances
 - A document $d = \{w_1, w_2, \dots, w_{M_d}\}$
 - A data instance $\mathbf{x} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(V)}]$: a V -dimensional feature vector
 - A vocabulary $\mathbf{x} = \{x_1, x_2, \dots, x_V\}$

Example: Spam Filtering

- Model: $P(y = k, w_1, \dots, w_M) = P(y = k) \prod_j^M P(w_j | y = k)$
- What are the parameters?

$P(y)$

Not spam : 0.66 Spam: 0.33

$P(w|y = \text{Spam})$

the : 0.0156
to : 0.0153
and : 0.0115
of : 0.0095
you : 0.0093
a : 0.0086
with: 0.0080
from: 0.0075
...

$P(w|y = \text{Not spam})$

the : 0.0210
to : 0.0133
of : 0.0119
2002: 0.0110
with: 0.0108
from: 0.0107
and : 0.0105
a : 0.0100
...

- Where do these tables come from?

Spam Example

$$P(Y, f_1 \dots f_n) = \frac{\begin{bmatrix} P(y_1) \prod_i P(f_i|y_1) \\ P(y_2) \prod_i P(f_i|y_2) \\ \vdots \\ P(y_k) \prod_i P(f_i|y_k) \end{bmatrix}}{P(f_1 \dots f_n)} \xrightarrow{+} P(Y|f_1 \dots f_n)$$

Word	P(w spam)	P(w not)	Total Spam	Total Not
(prior)	0.33333	0.66666	$\ln(0.3333)=-1.1$	-0.4
Gary	0.00002	0.00021	-11.8	-8.9
would	0.00069	0.00084	-19.1	-16.0
you	0.00881	0.00304	-23.8	-21.8
like	0.00086	0.00083	-30.9	-28.9
to	0.01517	0.01339	-35.1	-33.2
lose	0.00008	0.00002	-44.5	-44.0
weight	0.00016	0.00002	-53.3	-55.0
while	0.00027	0.00027	-61.5	-63.2
you	0.00881	0.00304	-66.2	-69.0
sleep	0.00006	0.00001	-76.0	-80.5


$$P(\text{spam} | W) = \exp(-76.0) / (\exp(-76.0) + \exp(-80.5))$$

$$= 98.9$$

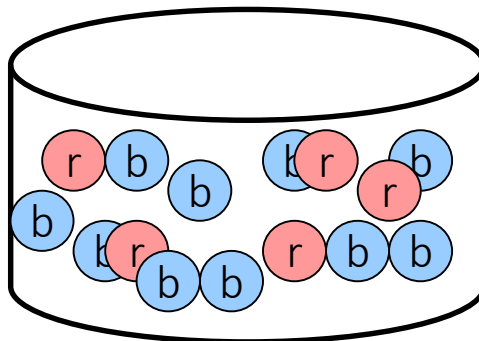
Parameter Estimation

- Estimating the distribution of a random variable
- *Empirically*: use training data (learning!)
 - E.g.: for each outcome x , look at the *empirical rate* of that value:

$$P_{\text{ML}}(x) = \frac{\text{count}(x)}{\text{total samples}}$$


 $P_{\text{ML}}(\textcolor{red}{r}) = 2/3$

- This is the estimate that maximizes the *likelihood of the data*



Parameter Estimation for Naïve Bayes

- Maximum likelihood estimation: Simply use the frequencies in the data

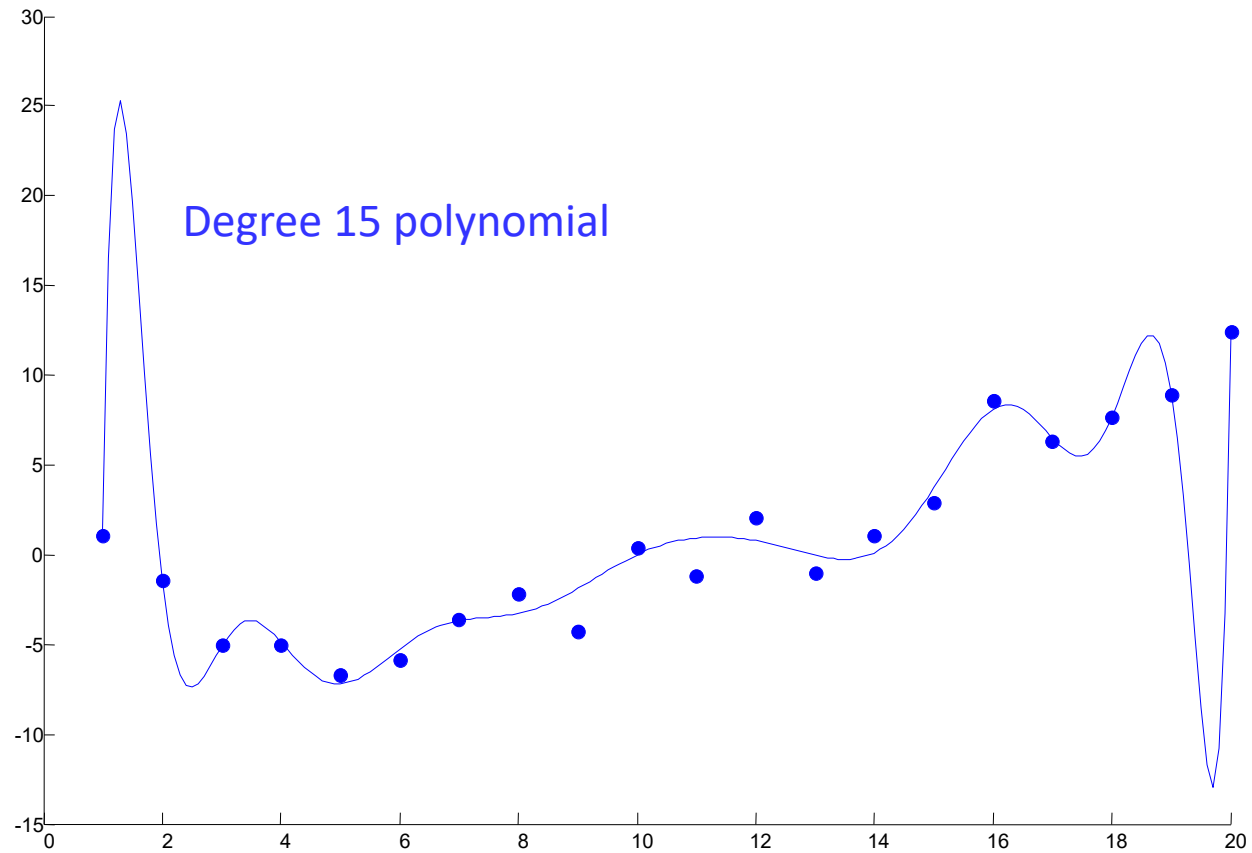
- For each label
$$P(y = k) = \frac{\#document(y_d = k)}{\#document}$$

- For each word type
$$P(x_j|k) = \frac{\sum_d \#count(y_d = k, x_j)}{\sum_x \sum_d count(y_d = k, x_j)}$$

fraction of times word type x_j appears
among all words in documents of label k

- Create mega-document for label k by concatenating all docs in this topic
 - Use frequency of words in mega-document

Overfitting



Example: Overfitting

- Posterior determined by *relative* probabilities (odds ratios):

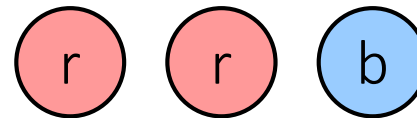
$\frac{P(w \text{Not spam})}{P(w \text{Spam})}$	$\frac{P(w \text{Spam})}{P(w \text{Not spam})}$
south-west : inf	screens : inf
nation : inf	minute : inf
morally : inf	guaranteed : inf
nicely : inf	\$205.00 : inf
extent : inf	delivery : inf
seriously : inf	signature : inf
...	...

What went wrong here?

Unlikely that every occurrence of “minute” is 100% spam
Unlikely that every occurrence of “seriously” is 100% not spam
What about all the words that don’t occur in the training set at all?

Laplace Smoothing

- Laplace's estimate:
 - Pretend you saw every outcome once more than you actually did



$$P_{LAP}(x) = \frac{c(x) + 1}{\sum_x [c(x) + 1]}$$
$$= \frac{c(x) + 1}{N + |X|}$$

$$P_{ML}(X) =$$

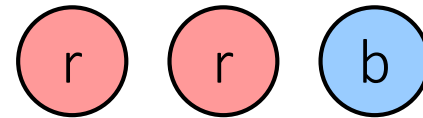
$$P_{LAP}(X) =$$

- Can derive this estimate with *Dirichlet priors*

Laplace Smoothing

- Laplace's estimate (extended):
 - Pretend you saw every outcome k extra times

$$P_{LAP,k}(x) = \frac{c(x) + k}{N + k|X|}$$



- What's Laplace with $k = 0$?
 - k is the **strength** of the prior

$$P_{LAP,0}(X) =$$

$$P_{LAP,1}(X) =$$

- Laplace for conditionals:
 - Smooth each condition independently:

$$P_{LAP,100}(X) =$$

$$P_{LAP,k}(x|y) = \frac{c(x, y) + k}{c(y) + k|X|}$$

Real NB: Smoothing

- For real classification problems, smoothing is critical

$$P(x_j|k) = \frac{\sum_d \#count(y_d = k, x_j) + \alpha}{\sum_x \sum_d count(y_d = k, x_j) + \alpha V}$$

- New odds ratios:

$$\frac{P(w|\text{Not spam})}{P(w|\text{Spam})}$$

helvetica	: 11.4
seems	: 10.8
group	: 10.2
ago	: 8.4
areas	: 8.3
...	

$$\frac{P(w|\text{Spam})}{P(w|\text{Not spam})}$$

verdana	: 28.8
Credit	: 28.4
ORDER	: 27.2
	: 26.9
money	: 26.5
...	

Do these make more sense?

A General Pipeline of Building a Classifier

Pre-processing data

Tokenization

Stemming/
normalization

N-gram

Stopwords
filtering

Preparing input data for machine learning

Feature construction
(e.g., VSM)

Feature construction
(e.g., word embeddings)

Feature selection
(e.g., DF filtering)

Configuration of Data and Metrics for Test

Training/Dev/Testing

K-Fold Cross Validation

Evaluation Metrics

Model Specification and Selection

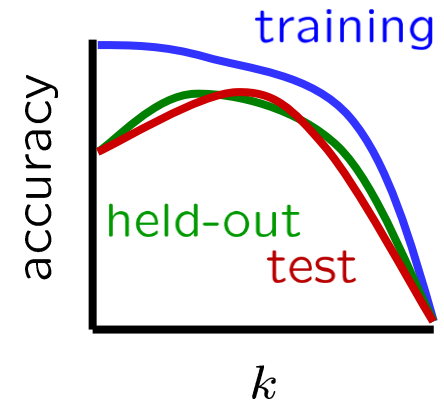
Specify models based
on assumptions

Evaluate model based
on C.V. and metrics

Output the model for
future use

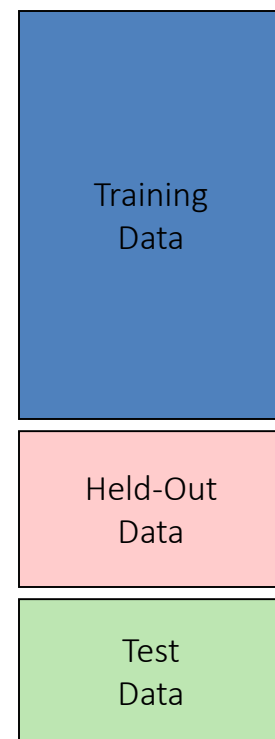
Tuning on Held-Out Data

- Now we've got two kinds of unknowns
 - **Parameters**: the probabilities $P(X|Y)$, $P(Y)$
 - **Hyperparameters**: e.g. the amount / type of smoothing to do, k , α
- What should we learn where?
 - Learn **parameters** from **training data**
 - Tune **hyperparameters** on different data
 - Why?
 - For each value of the hyperparameters, train and test on the **held-out data**
 - Choose the best value and do a final test on the test data



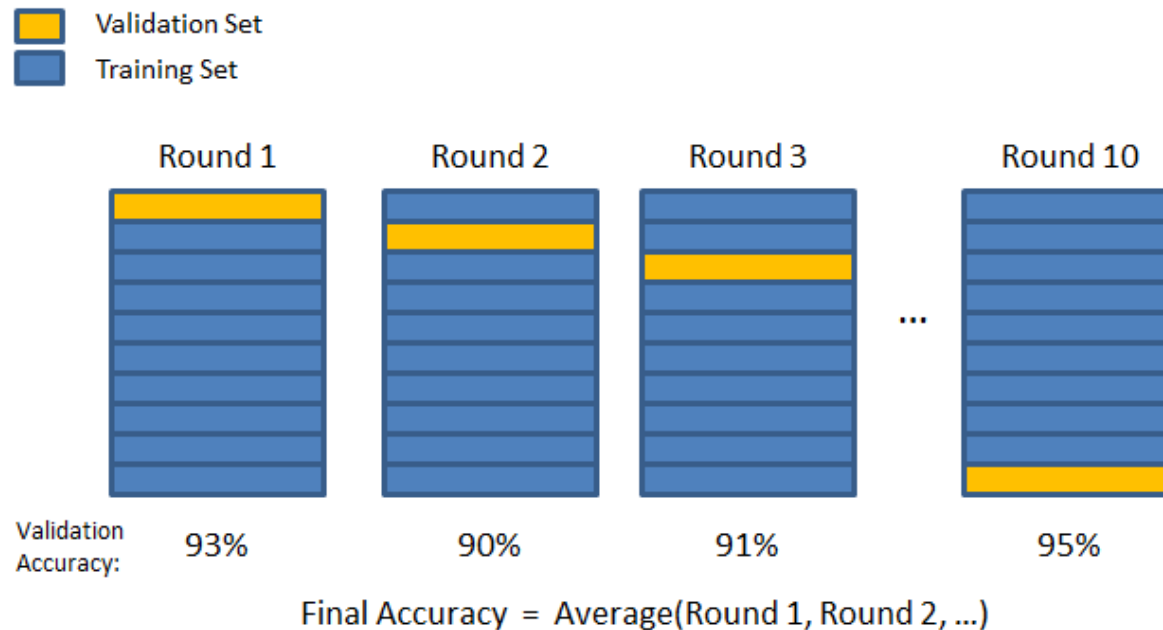
Important Concepts

- Data: labeled instances, e.g. emails marked spam/not spam
 - Training set
 - Held out set (development set/validation set)
 - Test set
- Features: attribute-value pairs which characterize each x
- Experimentation cycle
 - Learn parameters (e.g. model probabilities) on training set
 - (Tune hyperparameters on held-out set)
 - Compute accuracy of test set
 - Very important: never “peek” at the test set!



Cross validation

- Cross validation
 - Avoid noise in train/test separation
 - k -fold cross-validation



Errors, and What to Do

- Examples of errors

Dear GlobalSCAPE Customer,

GlobalSCAPE has partnered with ScanSoft to offer you the latest version of OmniPage Pro, for just \$99.99* - the regular list price is \$499! The most common question we've received about this offer is - Is this genuine? We would like to assure you that this offer is authorized by ScanSoft, is genuine and valid. You can get the . . .

. . . To receive your \$30 Amazon.com promotional certificate, click through to

<http://www.amazon.com/apparel>

and see the prominent link for the \$30 offer. All details are there. We hope you enjoyed receiving this message. However, if you'd rather not receive future e-mails announcing new store launches, please click . . .

What to Do About Errors?

- Need more features— words aren't enough!
 - Have you emailed the **sender** before?
 - Have **1K other people** just gotten the same email?
 - Is the **sending information consistent**?
 - Is the email in **ALL CAPS**?
 - Do **inline URLs** point where they say they point?
 - Does the **email address you by (your) name**?
- Can add these information sources as new variables in the NB model

Summary

- Bayes rule lets us do diagnostic queries with causal probabilities
- The naïve Bayes assumption takes all features to be independent given the class label
- We can build classifiers out of a naïve Bayes model using training data
- Smoothing estimates is important in real systems
- A note about Naïve Bayes with more math explanations
 - See in Canvas