ILLINOIS INSTITUTE OF TECHNOLOGY

Text Categorization and Naïve Bayes

CS-585

Natural Language Processing

Derrick Higgins

(with slides from William W. Cohen and Chris Manning)

TEXT CATEGORIZATION (CLASSIFICATION)

Text Classification: Definition

- The classifier:
  - *Input*: a document x
  - *Output*: a predicted class y from some fixed set of labels $y_1, \ldots, y_k$
- The learner:
  - *Input:* a set of *m* hand-labeled documents $(x_1, y_1), \ldots, (x_m, y_m)$
  - *Output:* a learned classifier $f: x \rightarrow y$

Text Classification: Examples

- Classify news stories as *World, US, Business, SciTech, Sports, Entertainment, Health, Other*
- Add MeSH terms to Medline abstracts (e.g. "Conscious Sedation" [E03.250])
- Classify business names by industry.
- Classify student essays as *A,B,C,D,* or *F.*
- Classify email as *Spam, Other.*
- Classify email to tech staff as *Mac, Windows, ..., Other.*
- Classify pdf files as *ResearchPaper, Other*
- Classify documents as *WrittenByReagan, GhostWritten*
- Classify movie reviews as *Favorable,Unfavorable,Neutral.*
- Classify technical papers as *Interesting, Uninteresting.*
- Classify web sites of companies by Standard Industrial Classification (SIC) code.
- Classify jokes as *Funny, NotFunny.*

Text Classification: Examples

- Best-studied benchmark: *Reuters-21578* newswire stories
  - 9603 train, 3299 test documents, 80-100 words each, 93 classes

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS

BUENOS AIRES, Feb 26

Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

→ Categories: grain, wheat (of 93 binary choices)

Representing text for classification

f(

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
BUENOS AIRES, Feb 26

Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

)=y

?

What is the ~~best~~ simplest useful representation for the document *x* being classified?

Bag of words representation

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
BUENOS AIRES, Feb 26

Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

→ Categories: grain, wheat

Bag of words representation

xxxxxxxxxxxxxxxxxxxx GRAIN/OILSEED xxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxx grain xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx grains, oilseeds xxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxx tonnes, xxxxxxxxxxxxxxxxx shipments
xxxxxxxxxxxx total xxxxxxxxx total xxxxxxxx xxxxxxxxxxxxxxxxxxxxx:

- Xxxxx **wheat** xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, total xxxxxxxxxxxxxxxx
- Maize xxxxxxxxxxxxxxxxx
- Sorghum xxxxxxxxxx
- Oilseed xxxxxxxxxxxxxxxxxxxxx
- Sunflowerseed xxxxxxxxxxxxxx
- Soybean xxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx....

→ Categories: grain, wheat

Bag of words representation

xxxxxxxxxxxxxxxxxxx GRAIN/OILSEED xxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxx grain xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx grains, oilseeds
xxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxx tonnes,
xxxxxxxxxxxxxxxxx shipments xxxxxxxxxxxx total xxxxxxxxx total
xxxxxxxx xxxxxxxxxxxxxxxxxxxx:

- Xxxxx wheat xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, total xxxxxxxxxxxxxxxx
- Maize xxxxxxxxxxxxxxxxx
- Sorghum xxxxxxxxxx
- Oilseed xxxxxxxxxxxxxxxxxxxxx
- Sunflowerseed xxxxxxxxxxxxxx
- Soybean xxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx....

word	*freq*
grain(s)	3
oilseed(s)	2
total	3
wheat	1
maize	1
soybean	1
tonnes	1
...	...

Categories: grain, wheat

NAÏVE BAYES

Text Classification with Naive Bayes

- Represent document x as set of $(w_i, Count(w_i))$ pairs:
  - $x = \{(\text{grain}, 3), (\text{wheat}, 1), \ldots, (\text{the}, 6)\}$
- For each y, build a probabilistic model $\Pr(X|Y = y)$ of "documents" in class y
  - $\Pr(X = \{(\text{grain}, 3), \ldots\}|Y = \text{wheat}) = \ldots$
  - $\Pr(X = \{(\text{grain}, 3), \ldots\}|Y = \text{nonWheat}) = \ldots$
- To classify, find the *y* which was most likely to *generate x—i.e.*, which gives x the best score according to $\Pr(x|y)$
  - $f(x) = \text{argmax}_y \Pr(x|y) \times \Pr(y)$

Bayes Rule

$$\Pr(y \mid x) \cdot \Pr(x) = \Pr(x, y) = \Pr(x \mid y) \cdot \Pr(y)$$

$$\Rightarrow \qquad \Pr(y \mid x) = \frac{\Pr(x \mid y) \cdot \Pr(y)}{\Pr(x)}$$

$$\Rightarrow \arg\max_y \Pr(y \mid x) = \arg\max_y \Pr(x \mid y) \cdot \Pr(y)$$

Text Classification with Naive Bayes

- How to estimate $\Pr(X|Y)$?
- *Simplest useful* process to generate a bag of words:
  - pick word 1 according to $\Pr(W|Y)$
  - repeat for word 2, 3, ...
  - each word is generated *independently* of the others (which is clearly not true) but means

$$\Pr(w_1,\ldots,w_n \mid Y=y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y=y)}_{\text{How to estimate Pr(W|Y)?}}$$

Two Unreasonable Assumptions

- Bag-of-words:

  The order of the words in document *d* makes no difference (but repetitions do)
- Conditional Independence:

  Words appear independently of each other, given the document class

  (e.g., if you see "car", the word "drive" is no more likely to appear than if you saw "dog")

Text Classification with Naive Bayes

- How to estimate Pr(X|Y) ?

$$\Pr(w_1,\ldots,w_n \mid Y=y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y=y)}$$

Estimate *Pr(w|y)* by looking at the data...

$$\Pr(W=w \mid Y=y) = \frac{\text{count}(W=w \text{ and } Y=y)}{\text{count}(Y=y)}$$

Simple Smoothing

- If X contains a vocabulary word that does not occur with class $Y = y$ in the training:

$$P(X|Y = y) = 0, \text{ no matter what else is there!}$$

- Solution:
  - Assign small probability to unseen words,
  - Taking away probability from seen words
  - Every word that occurred N times with class $Y = y$, we will pretend actually occurred $N + \alpha$ times

Text Classification with Naive Bayes

- How to estimate Pr(X|Y) ?

$$\Pr(w_1,\ldots,w_n \mid Y=y) = \prod_{i=1}^{n} \Pr(w_i \mid Y=y)$$

... and also imagine α "pseudo-occurrences" of w_i in class $Y = y$

- $\Pr(w_i|Y = y) = \frac{count(w_i \wedge Y=y)+\alpha}{count(Y=y)+\alpha|V|}$

Text Classification with Naive Bayes

- How to estimate Pr(X|Y) ?

$$\Pr(w_1, \ldots, w_n \mid Y = y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y = y)}$$

For instance, α=3

- $\Pr(w_i|Y = y) = \frac{count(w_i \wedge Y=y)+3}{count(Y=y)+3|V|}$

Avoiding Underflow

- Consider:
  - Many docs have more than 100 words
  - Word probabilities will each be <0.1
  - So, P(X|Y)<10^{-100} for any document X

  ➔UNDERFLOW!!

- Solution: $\log a > \log b$ iff $a > b$

  Use $\log[P(X|Y)P(Y)] = \log P(X|Y) + \log P(Y)$

  $$\log P(X|Y) = \sum_{w_i \varepsilon X} \log P(w_i|Y)$$

Text Classification with Naive Bayes

- Putting this together:

  for each document x_i with label y_i

      d_count[y_i]++

      d_count++

      for each word w_{ij} in x_i

          w_count[w_{ij}][y_i]++

          w_count[y_i]++

  – to classify a new $x=w_1 \ldots w_n$, pick y with top *score*:

$$score(y, w_1, \cdots, w_n) = \log \frac{d\_count[y]}{d\_count} + \sum_{i=1}^{n} \log \frac{w\_count[w_i][y] + \alpha}{w\_count[y] + \alpha|V|}$$

key point: we only need counts for words that actually appear in *x*

Naïve Bayes: Putting it all together

$$\log\big(P(Y=y,X)\big) = \log(P(X|Y=y)) + \log(P(Y=Y))$$

$$\log\big(P(Y=y)\big) = \log\frac{d\_count[y]}{d\_count}$$

$$\log\big(P(X|Y=y)\big) = \sum_{w \in X} \log\frac{w\_count[w][y] + \alpha}{w\_count[y] + \alpha|V|}$$

Some numerical
care required

$$P(Y=y|X) = \frac{P(Y=y,X)}{\sum_{y' \in Y} P(Y=y',X)}$$

WebKB Experiment (1998)

- Classify webpages from CS departments into:
  - student, faculty, course, project
- Train on ~5,000 hand-labeled web pages
  - Cornell, Washington, U.Texas, Wisconsin
- Crawl and classify a new site (CMU) using Naïve Bayes

Results

	Student	Faculty	Person	Project	Course	Departmt
Extracted	180	66	246	99	28	1
Correct	130	28	194	72	25	1
Accuracy:	72%	42%	79%	73%	89%	100%

Faculty

associate	0.00417
chair	0.00303
member	0.00288
ph	0.00287
director	0.00282
fax	0.00279
journal	0.00271
recent	0.00260
received	0.00258
award	0.00250

Students

resume	0.00516
advisor	0.00456
student	0.00387
working	0.00361
stuff	0.00359
links	0.00355
homepage	0.00345
interests	0.00332
personal	0.00332
favorite	0.00310

Courses

homework	0.00413
syllabus	0.00399
assignments	0.00388
exam	0.00385
grading	0.00381
midterm	0.00374
pm	0.00371
instructor	0.00370
due	0.00364
final	0.00355

Departments

departmental	0.01246
colloquia	0.01076
epartment	0.01045
seminars	0.00997
schedules	0.00879
webmaster	0.00879
events	0.00826
facilities	0.00807
eople	0.00772
postgraduate	0.00764

Research Projects

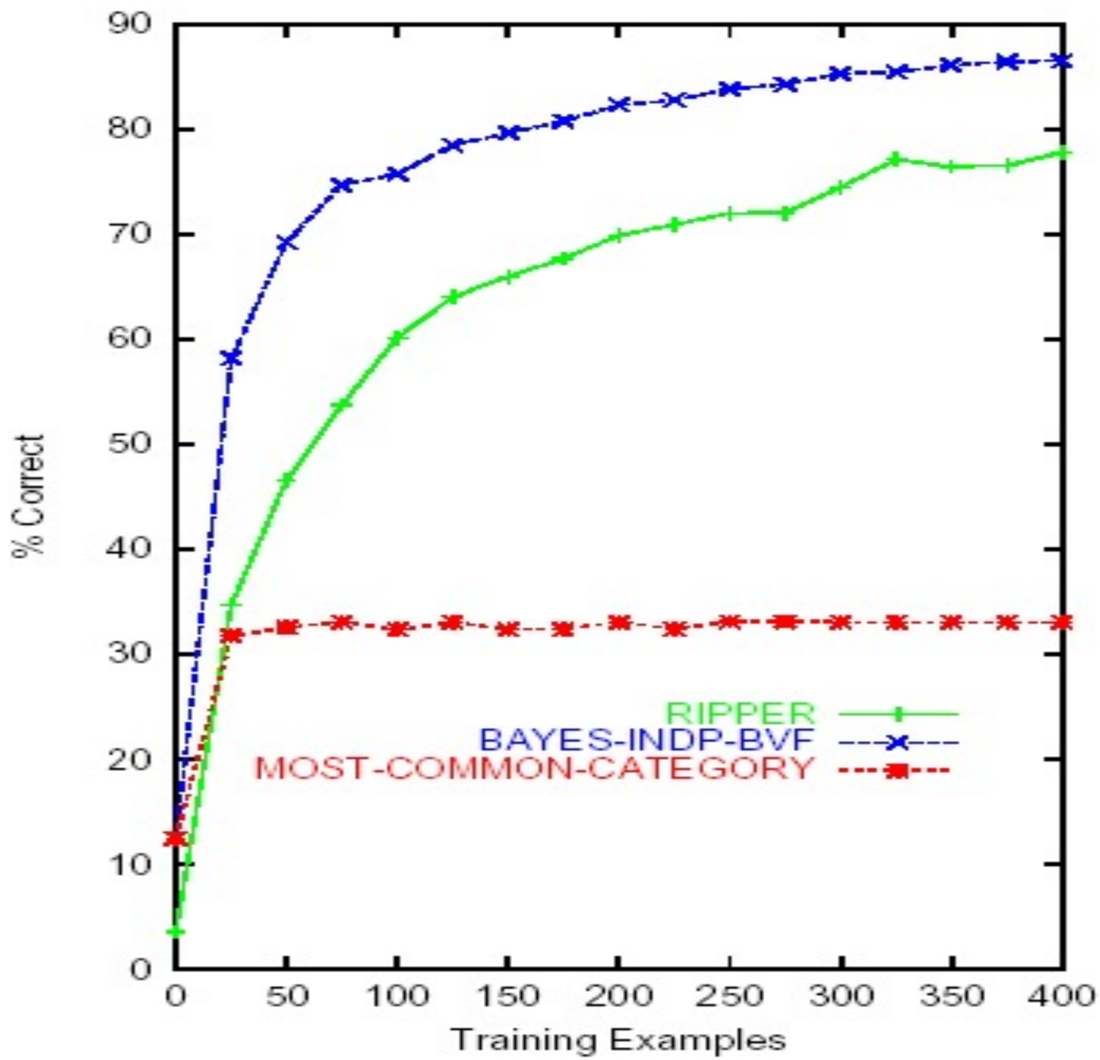
investigators	0.00256
group	0.00250
members	0.00242
researchers	0.00241
laboratory	0.00238
develop	0.00201
related	0.00200
arpa	0.00187
affiliated	0.00184
project	0.00183

Others

type	0.00164
jan	0.00148
enter	0.00145
random	0.00142
program	0.00136
net	0.00128
time	0.00128
format	0.00124
access	0.00117
begin	0.00116

Naïve Bayes vs Rules (Provost 1999)

More experiments: rules (concise boolean queries based on keywords) *vs* Naïve Bayes for content-based foldering showed Naive Bayes is better and faster.



Naive Bayes Summary

- Pros:
  - Very fast and easy-to-implement
  - Well-understood formally & experimentally
    - see "Naive (Bayes) at Forty", Lewis, ECML98
- Cons:
  - Seldom gives the very best performance
  - "Probabilities" $\Pr(y|x)$ are not accurate
    - Probabilities tend to be close to zero or one

LINEAR SEPARATORS

Linear Separators

Consider a 2-class problem; we can classify by asking:

$$\frac{P(X|Y=y_1)P(Y=y_1)}{P(X|Y=y_2)P(Y=y_2)} > 1\ ?$$

In other words:

$$\log P(X|Y=y_1) + \log P(Y=y_1) - \log P(X|Y=y_2) - \log P(Y=y_2) > 0\ \ ?$$

$$\log P(X|Y=y_1) - \log P(X|Y=y_2) > \log P(Y=y_2) - \log P(Y=y_1)\ \ ?$$

$$\log P(X|Y=y_1) - \log P(X|Y=y_2) > \boldsymbol{\theta}\ \ ?$$

Linear Separators

$$\log P(X|Y = y_1) - \log P(X|Y = y_2) > \theta \quad ?$$

$$\sum_{w_i \in X} (\log P(w_i|Y = y_1) - \log P(w_i|Y = y_2)) > \theta \quad ?$$

$$\sum_{w_i} (\log P(w_i|Y = y_1) - \log P(w_i|Y = y_2)) \times count(w_i, X) > \theta \quad ?$$

$$\sum_{i} \omega_i x_i > \theta \quad ?$$

$$\mathbf{w}^T \mathbf{x} > \theta \quad ?$$

Linear Separators

Bag of words and conditional independence assumptions

$$\log P(X|Y = y_1) - \log P(X|Y = y_2) > \theta \quad ?$$

$$\sum_{w_i \in X} (\log P(w_i|Y = y_1) - \log P(w_i|Y = y_2)) > \theta \quad ?$$

$$\sum_{w_i} (\log P(w_i|Y = y_1) - \log P(w_i|Y = y_2)) \times count(w_i, X) > \theta \quad ?$$

$$\sum_{i} \omega_i x_i > \theta \quad ?$$

$$\mathbf{w}^T \mathbf{x} > \theta \quad ?$$

Linear Separators

Sum over all word types instead of tokens; factor out document count into a separate term

$$\log P(X|Y=y_1) - \log P(X|Y=y_2) > \theta \quad ?$$

$$\sum_{w_i \in X} (\log P(w_i|Y=y_1) - \log P(w_i|Y=y_2)) > \theta \quad ?$$

$$\sum_{w_i} (\log P(w_i|Y=y_1) - \log P(w_i|Y=y_2)) \times count(w_i, X) > \theta \quad ?$$

$$\sum_i \omega_i x_i > \theta \quad ?$$

$$\mathbf{w}^T \mathbf{x} > \theta \quad ?$$

Linear Separators

$$\log P(X|Y=y_1) - \log P(X|Y=y_2) > \theta \quad ?$$

$$\sum_{w_i \in X} (\log P(w_i|Y=y_1) - \log P(w_i|Y=y_2)) > \theta \quad ?$$

$$\sum_{w_i} (\log P(w_i|Y=y_1) - \log P(w_i|Y=y_2)) \times count(w_i, X) > \theta \quad ?$$

$$\sum_{i} \omega_i x_i > \theta \quad ?$$

$$\mathbf{w}^T \mathbf{x} > \theta \quad ?$$

Define

$$\omega_i = (\log P(w_i|Y=y_1) - \log P(w_i|Y=y_2))$$
$$x_i = count(w_i, X)$$

Linear Separators

$$\log P(X|Y=y_1) - \log P(X|Y=y_2) > \theta \quad ?$$

$$\sum_{w_i \in X} (\log P(w_i|Y=y_1) - \log P(w_i|Y=y_2)) > \theta \quad ?$$

$$\sum_{w_i} (\log P(w_i|Y=y_1) - \log P(w_i|Y=y_2)) \times count(w_i, X) > \theta \quad ?$$

$$\sum_i \omega_i x_i > \theta \quad ?$$

Vector product
notation

$$\mathbf{w}^T\mathbf{x} > \theta \quad ?$$

Linear Separators

$$\log P(X|Y=y_1) - \log P(X|Y=y_2) > \theta \quad ?$$

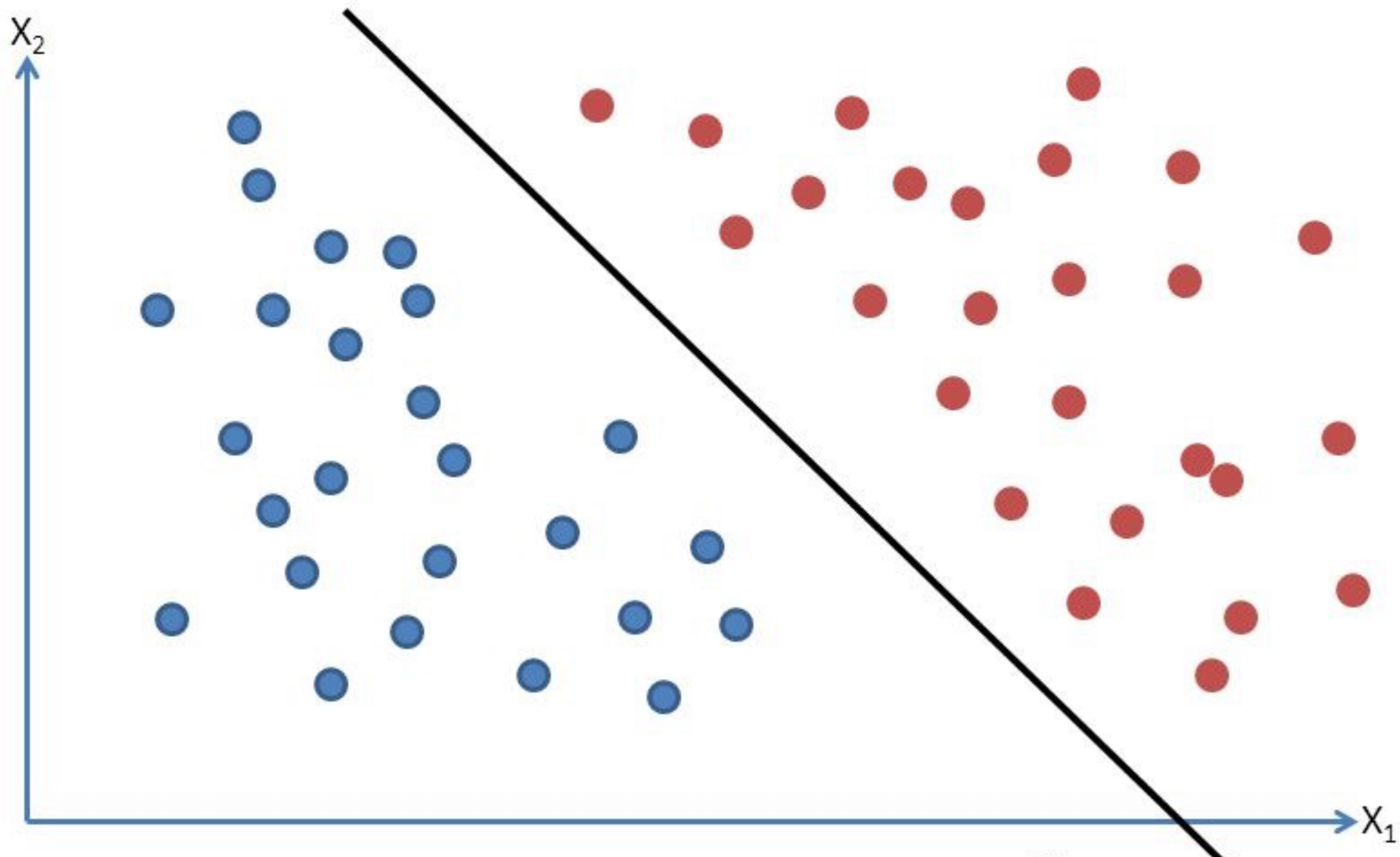
$$\sum_{w_i \in X} (\log P(w_i|Y=y_1) - \log P(w_i|Y=y_2)) > \theta \quad ?$$

$$\sum_{w_i} (\log P(w_i|Y=y_1) - \log P(w_i|Y=y_2)) \times count(w_i, X) > \theta \quad ?$$

$$\sum_{i} \omega_i x_i > \theta \quad ?$$

$$\mathbf{w}^T \mathbf{x} > \theta \quad ?$$

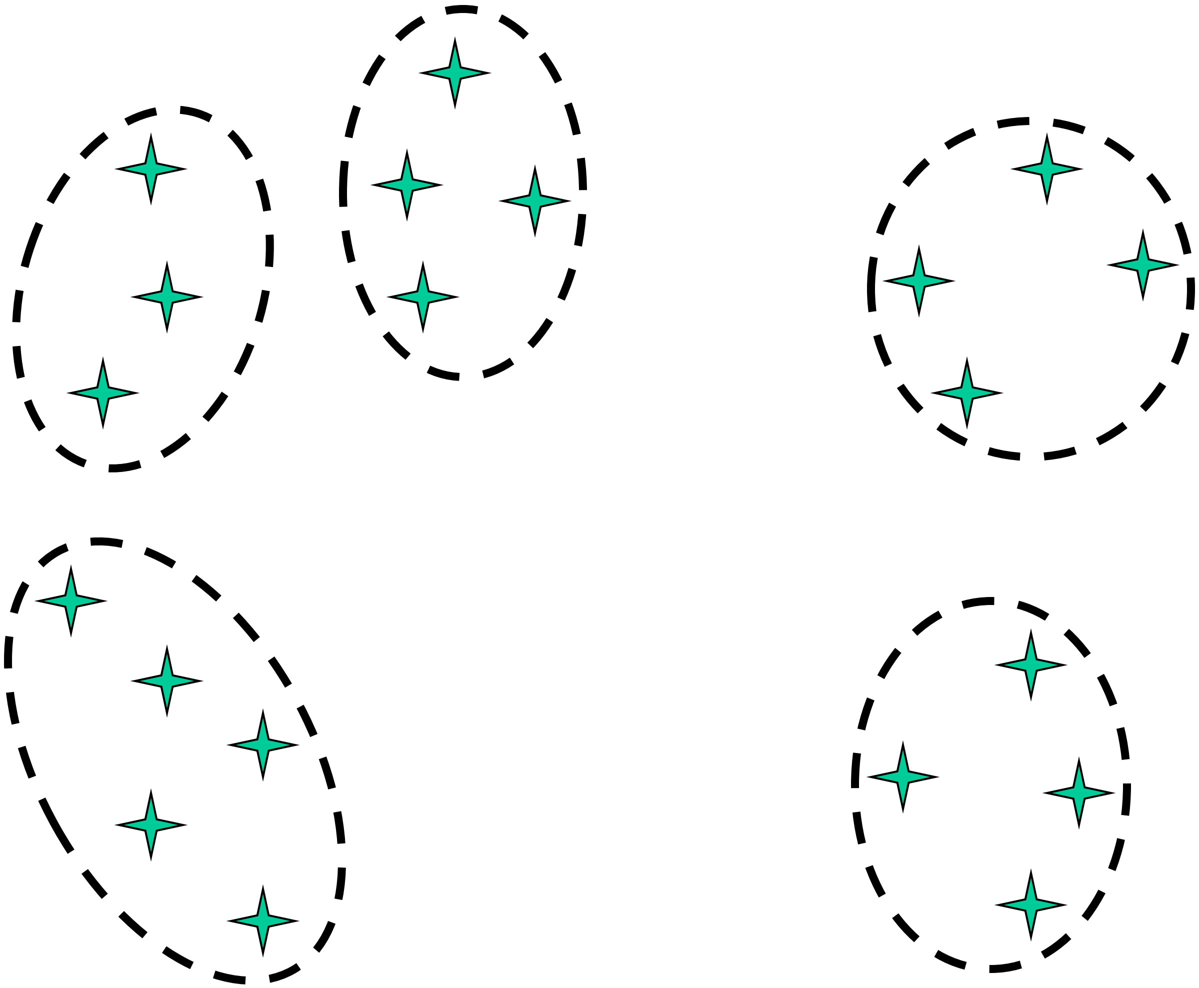
Linear Separators



UNSUPERVISED CLASSIFICATION

Unsupervised Classification

- Text classification *without* labeled training or other information sources
- Cannot label to predefined categories (there are none), so try to find "natural" ones
- Use **clustering** methods to find "sensible" categories of documents



Non-Hierarchical Clustering

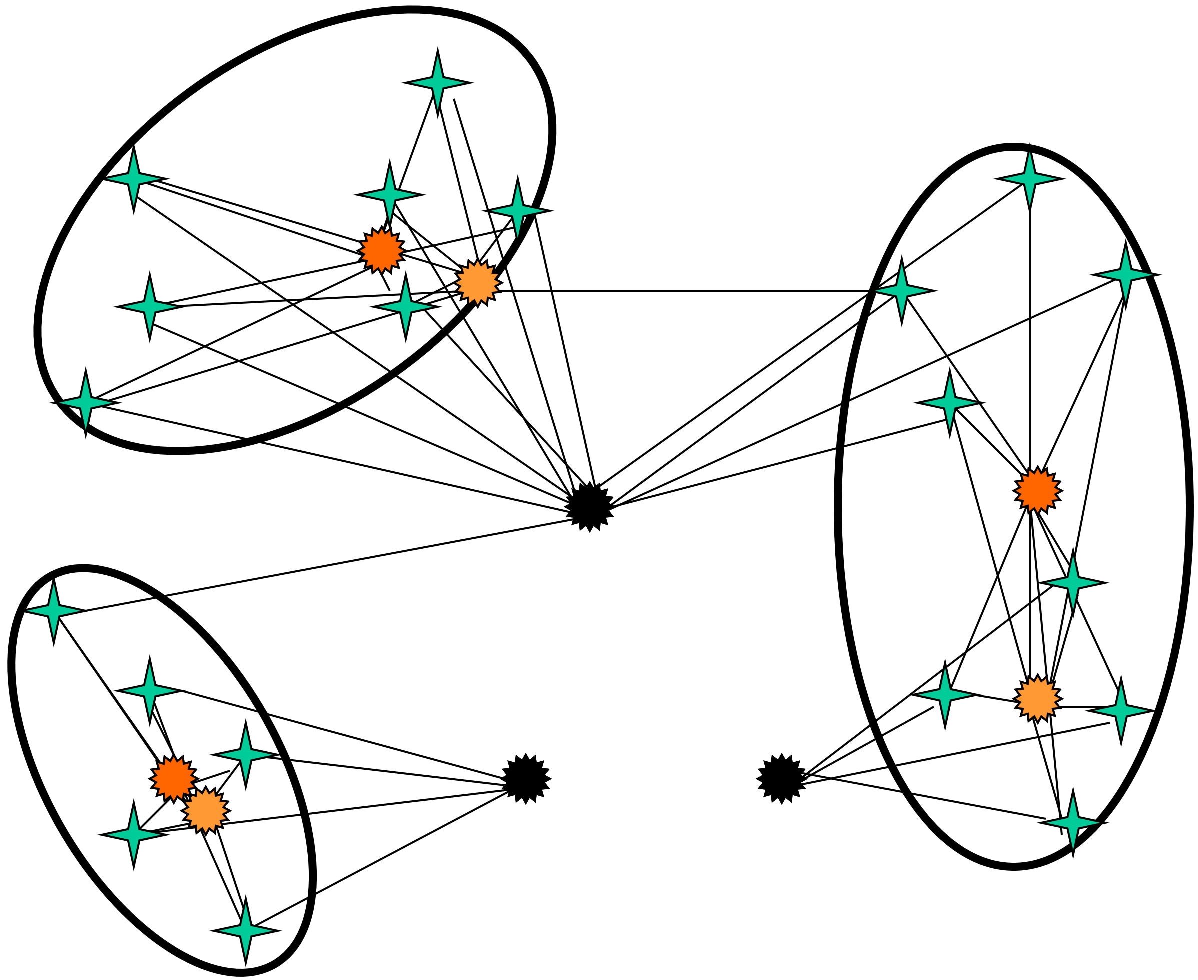
- **Iterative clustering:**
  - Start with initial (random) set of clusters
  - Assign each object to a cluster (or clusters)
  - Recompute cluster parameters
  - Stop when clustering is “good”

- Q: How many clusters?

  A: Who knows??

But there are some principled methods...

K-means Clustering



K-means Algorithm

Input:

- Set $X = \{x_1, \ldots, x_n\}$ of objects
- Distance measure d: $X \times X \rightarrow \mathbb{R}$
- Mean function μ

Select k initial cluster centers $f_1, \ldots, f_k$

while not finished **do:**

  for all clusters c_j **do:**

    $c_j \leftarrow \{ x_i \mid f_j = \mathbf{argmin}_f \, d(x_i, f) \}$

  for all means f_j **do:**

    $f_j \leftarrow \mu(c_j)$

K-means as EM (ish)

E: Calculate cluster assignments given current centroid locations

Data point	Location	Closest cluster centroid
1	(-1,1)	2
2	(-1,-1)	3
3	(1,2)	1
4	(2,2)	1
5	(-2,1)	2
6	(-2,-2)	3
7	(-3,-1)	3
8	(4,2)	1
9	(-1,0)	2

Should actually be a "soft" assignment

K-means as EM (ish)

M: Move the cluster centroids to the center of their associated data points (making the data more "likely")

Data point	Location	Closest cluster centroid
1	(-1,1)	2
2	(-1,-1)	3
3	(1,2)	1
4	(2,2)	1
5	(-2,1)	2
6	(-2,-2)	3
7	(-3,-1)	3
8	(4,2)	1
9	(-1,0)	2

Cluster	New centroid
1	(2.33,2)
2	(-1.33,0.67)
3	(-2,-1.33)

$\text{mean}([1,2],[2,2],[4,2])$

$\text{mean}([-1,1],[-2,1],[-1,0])$

$\text{mean}([-1,-1],[-2,-2],[-3,-1])$

The EM Algorithm

Soft clustering method to solve

$$\theta^* = \arg\max_\theta P_{model}(X \mid \theta)$$

Note: Any occurrence of the data consists of:

- **Observable variables:** The objects we see
  - *Bags of words*
  - *Word sequences in tagging tasks*
- **Hidden variables:** Which cluster generated which object
  - *Document categories*
  - *Underlying tag sequences*

Two Principles

Expectation: If we knew θ we could compute the expected values of the hidden variables (e.g, probability of x belonging to some cluster)

Maximization: If we knew the hidden structure, we could compute the maximum likelihood value of θ

Iterative Solution

Initialize: Choose an initial θ_0

Then iterate until convergence:

- E-step: Compute $(X, Z_i) \; = \; \mathrm{Exp}[X, Z \mid \theta_i]$
- M-step: Choose θ_{i+1} to maximize $P(X, Z_i, \theta_{i+1})$

M-step sometimes cannot be computed, but moving along its gradient also works

EM for Naive Bayes Text Classification

E-step: Compute P(c_k | d_i) for each document d_i and category c_k given current model

M-step: Re-estimate the model parameters $P(w_j \mid c_k)$ and $P(c_k)$

Continue as long as log-likelihood of corpus increases:

$$\log \prod_i \sum_k P(d_i \mid c_k) P(c_k) = \sum_i \log \sum_k P(d_i \mid c_k) P(c_k)$$

E-Step

- For each document d_i and each category c_k, estimate the posterior probability $h_{ik} = P(c_k \mid d_i)$:

$$h_{ik} = \frac{P(d_i \mid c_k)P(c_k)}{\sum_{k'} P(d_i \mid c_{k'})P(c_{k'})}$$

- To compute $P(d_i \mid c_k)$, use naive Bayes:

$$P(d_i \mid c_k) = \prod_{w_j \in d_k} P(w_j \mid c_k)$$

M-Step

Re-estimate parameters using maximum-likelihood estimation:

$$P(w_j|c_k) = \frac{\sum_{d_i : w_j \in d_i} h_{ik}}{\sum_{d_i, \forall w_{j'} \in d_i} h_{ik}}$$

$$P(c_k) = \frac{\sum_i h_{ik}}{\sum_k \sum_i h_{ik}}$$

EM for Naïve Bayes

E: Calculate probabilistic assignments of documents to categories

Document	P(Y=sports|X)	P(Y=news|X)
1	0.3	0.7
2	0.01	0.99
3	0.2	0.8
4	0.7	0.3
5	0.9	0.1
6	0.99	0.01
7	0.6	0.4
8	0.5	0.5
…	…	…

$$\frac{P(Y=y,X)}{\sum_{y'\in Y} P(Y=y',X)}$$

EM for Naïve Bayes

M: Recalculate P(Y) and P(W|Y) to maximize the likelihood of the data under soft category assignments

Document	P(Y=sports|X)	P(Y=news|X)
1	0.3	0.7
2	0.01	0.99
3	0.2	0.8
4	0.7	0.3
5	0.9	0.1
6	0.99	0.01
7	0.6	0.4
8	0.5	0.5
…	…	…

$$\frac{\sum_i h_{ik}}{\sum_k \sum_i h_{ik}}$$

$$\frac{\sum_{d_i : w_j \in d_i} h_{ik}}{\sum_{d_i, \forall w_{j'} \in d_i} h_{ik}}$$

y	P(Y=y)
sports	0.525
news	0.475

w	P(w|Y=sports)	P(w|Y=news)
ball		
senate		
frog		
…	…	…

Decision Procedure

Assign categories by:

$$cat(d_i) = \arg\max_{c_k} \left[\log P(c_k) + \sum_{w_j \in d_i} \log P(w_j \mid c_k) \right]$$

- Can adjust number of categories *k* to get finer or coarser distinctions
- If adding more categories doesn't increase log-likelihood of data much, then stop