

Text Analytics Course – 2 Day

Cohan Sujay Carlos
Aiaioo Labs
Bangalore

Is text analytics important?

I shall chuck three buzz-words at you:

- One type of **big data** is **unstructured data**
- One type of **unstructured data** is **text**
- The analysis of **text** is **text analysis**

You didn't answer my question.
Is text analytics important?

Text is an encoding of language.

*Humans have transmitted knowledge using
language for 100s of 1000s of years!*

And still do.

**So, all of human knowledge exists somewhere
as language, which could be encoded as text.**

What we're going to cover

- One Machine Learning Tool
- How to reduce Text Analytics Tasks to steps which you can solve with merely this one Machine Learning Tool
- A number of Text Analytics Design Patterns

The One ML Tool

**Get ready for the one ML tool you'll need to
hack text analytics**

DRUM ROLL !!!!!!!

The One

**Get ready for the one ML tool you'll need to
hack text analytics**

The Classifier 😊

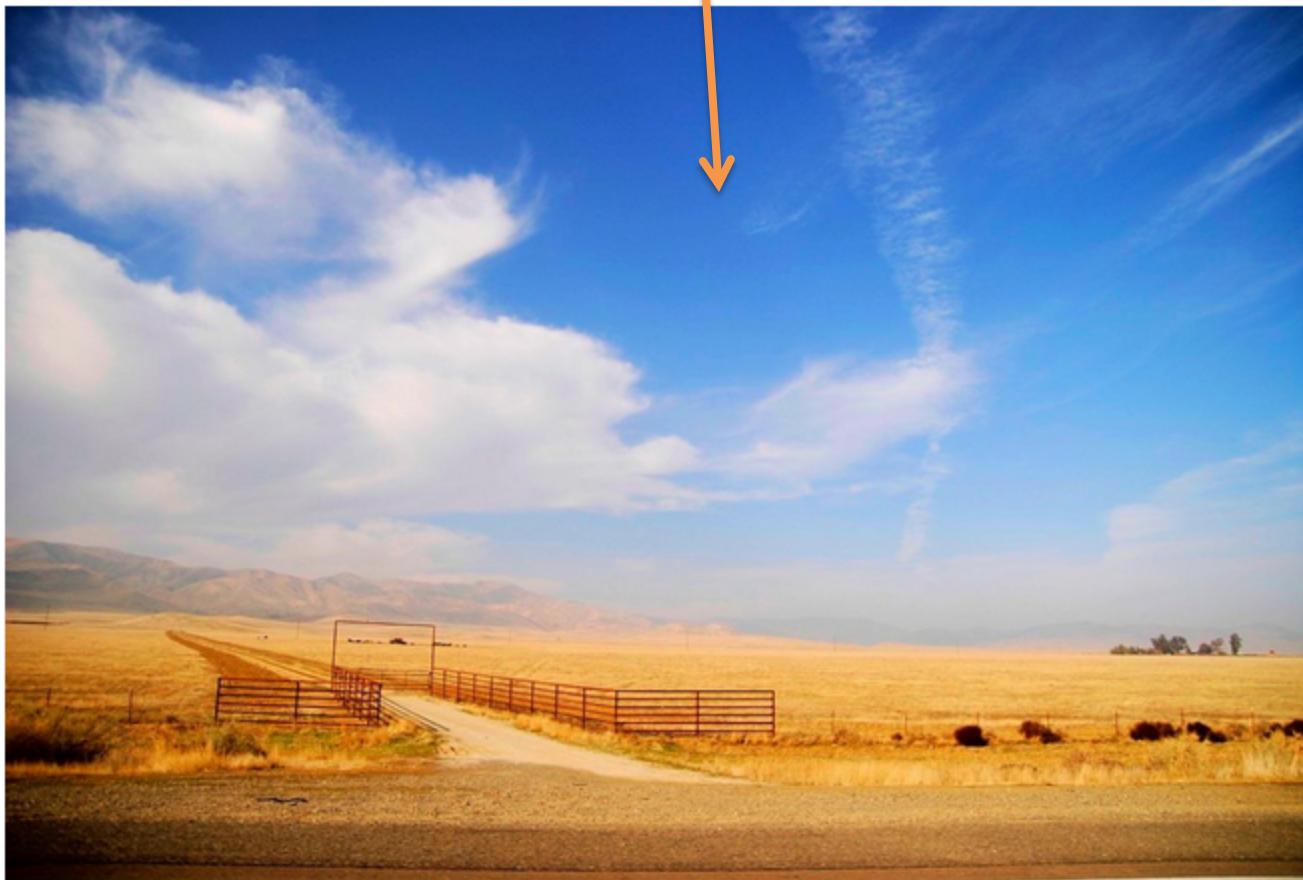
That ML Tool is the Classifier

What is a Classifier?

Something that performs classification.

What is Classification?

What colour do you see here?



Russell Mondy/Flickr

What is Classification?

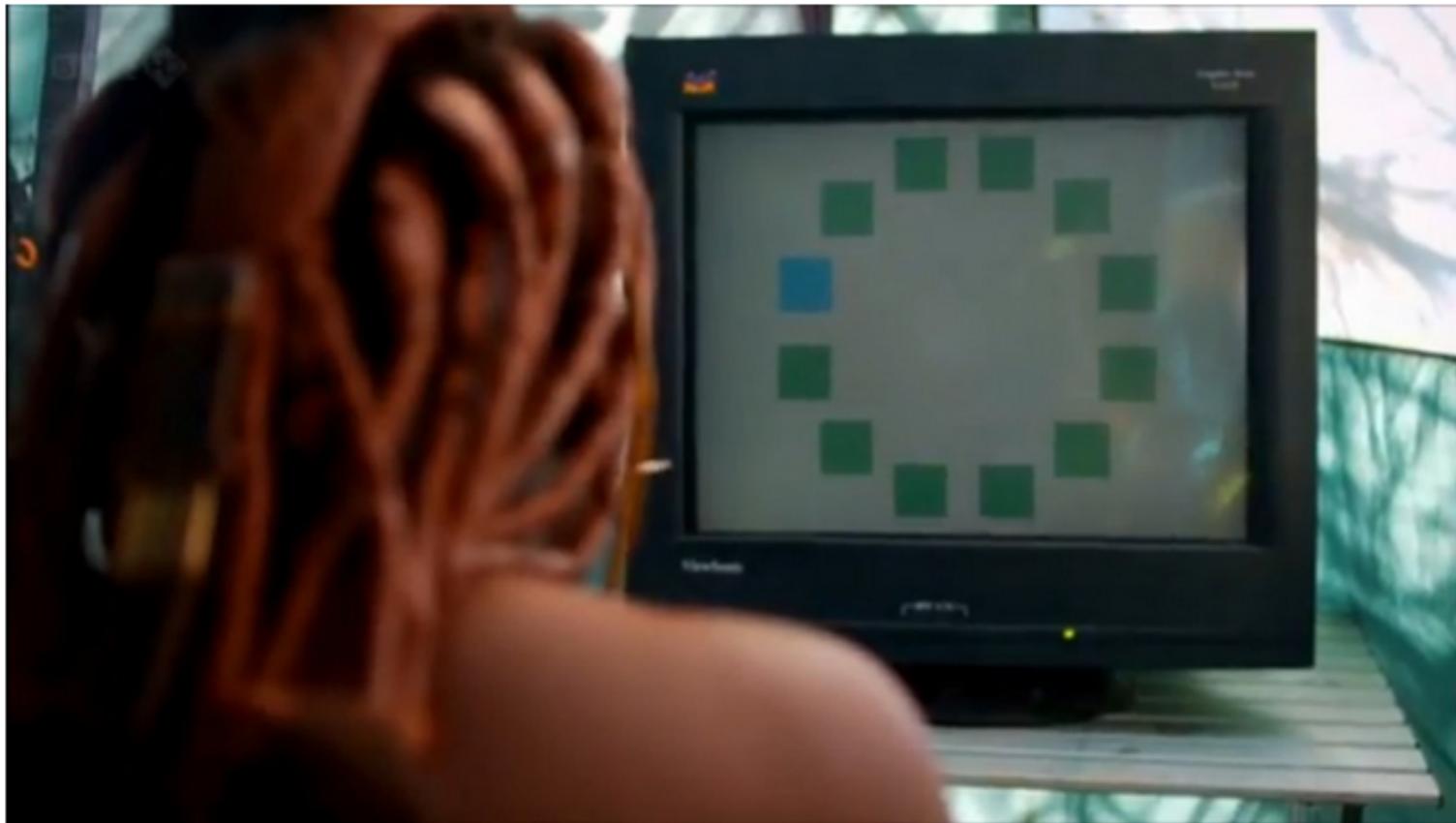
Classification = Categorizing = Labelling = Deciding



Russell Mondy/Flickr

What is Classification?

Which of these squares is a different shade?

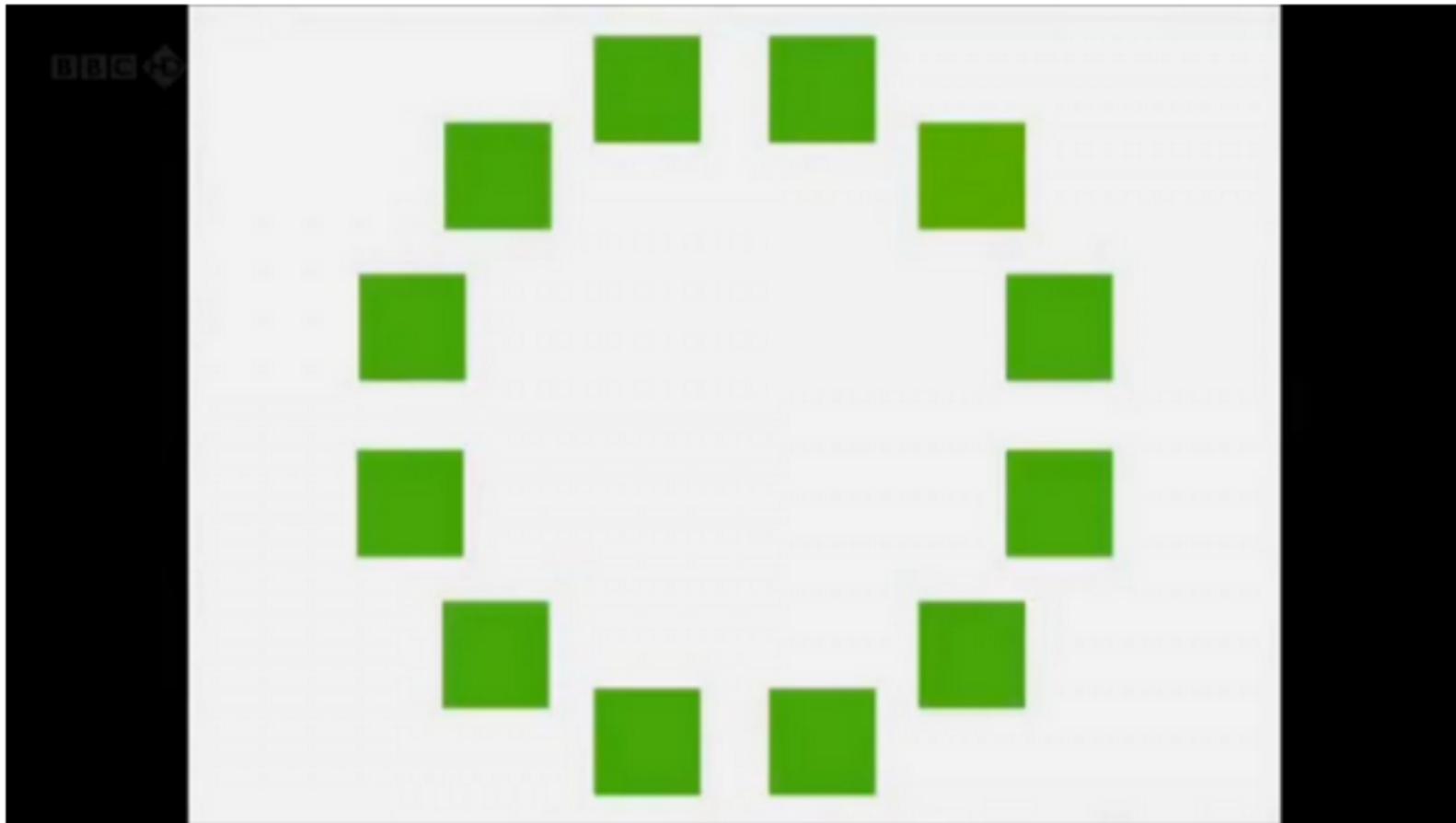


Vidipedia/Himba color experiment

Courtesy: <http://www.businessinsider.in/No-one-could-see-the-color-blue-until-modern-times/articleshow/46403449.cms>

What is Classification?

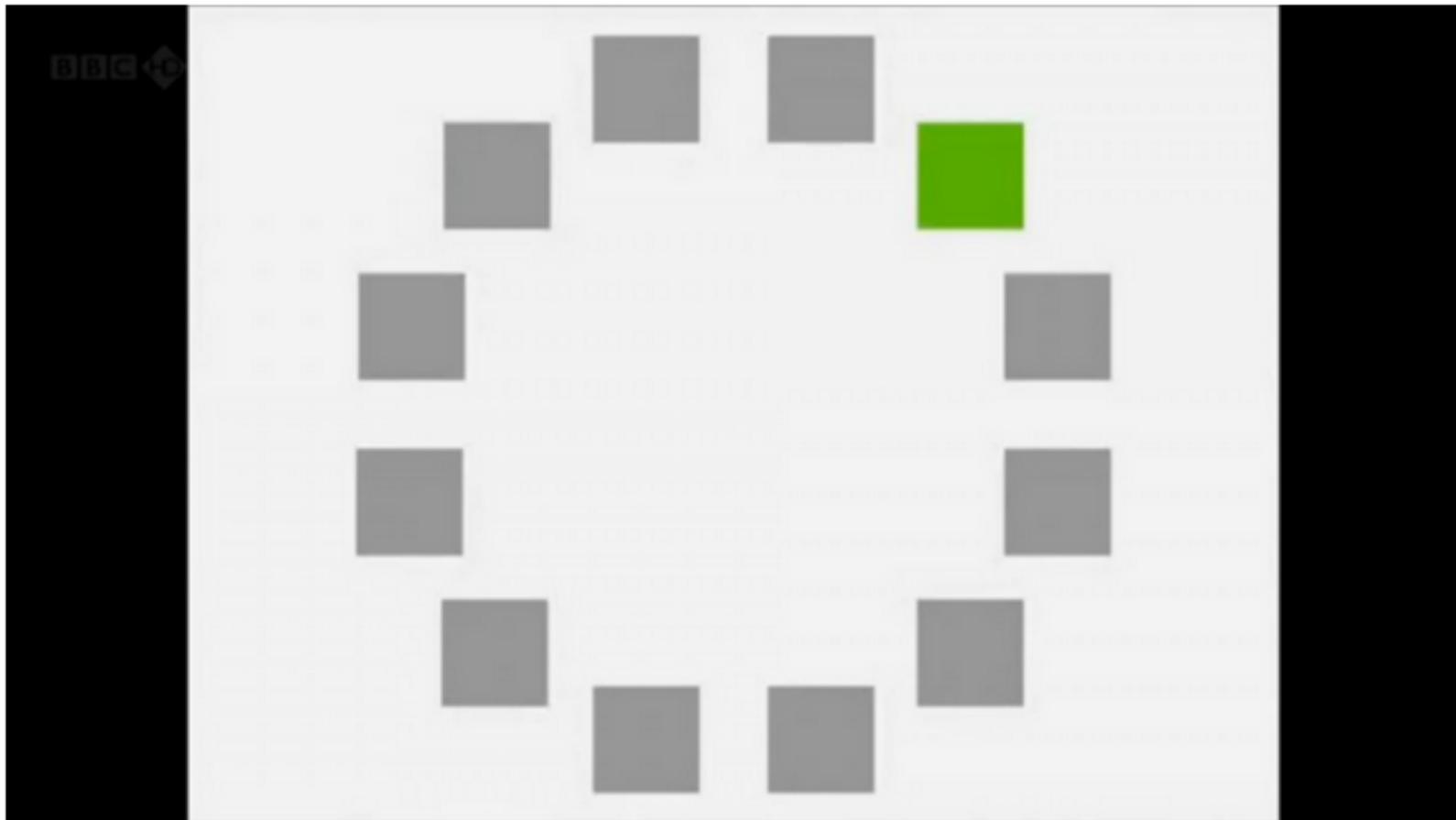
Which of these squares is a different shade?



Vidipedia/Himba Colour Experiment

What is Classification?

Which of these squares is a different shade?

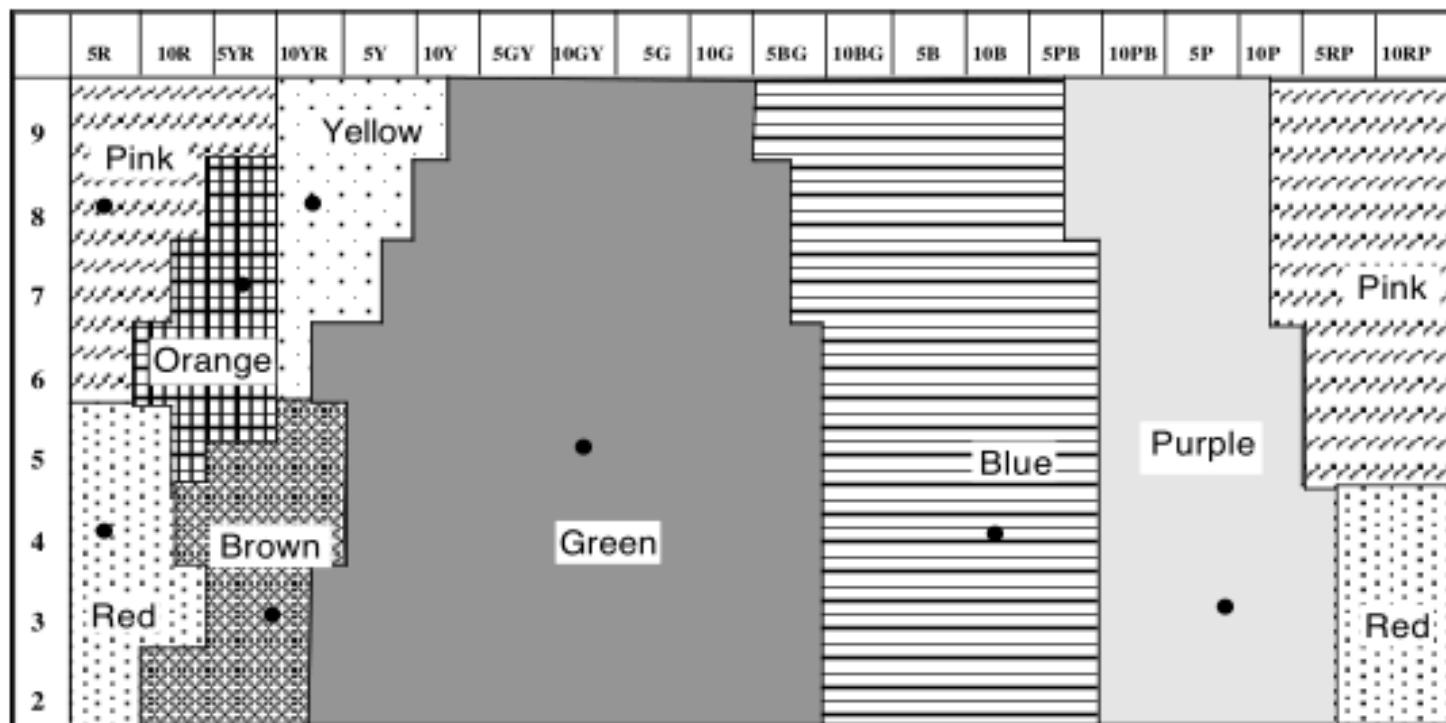


Vidipedia/Himba Colour Experiment

What is Classification?

Training data in the English language.

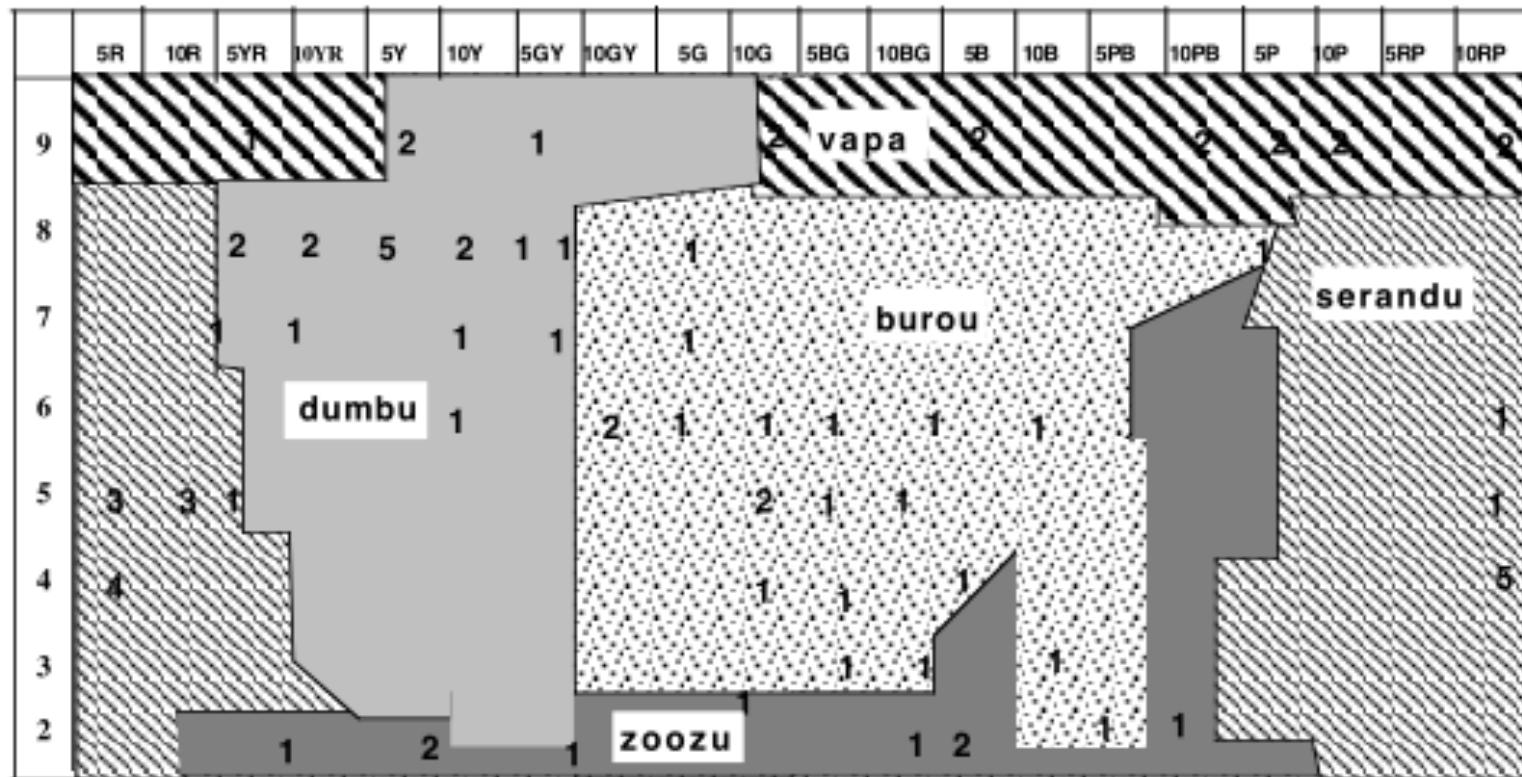
English naming distribution



What is Classification?

Training data in the Himba language.

Himba naming distribution



That ML Tool is the Classifier

What is a Classifier?

Something that performs classification.

Classification = categorizing

Classification = deciding

Classification = labelling

Classification = Deciding = Labelling

Classification = Deciding = Labelling

5' 8"
5'11"
5'8"
5' 2"
6'9"
6'8"
5'11"
6'2"
6'6"
6'10"

Classify these door heights as: **Short or Tall ?**

Classification = Deciding = Labelling

5' 8" Short

5'11" Short

5'8"

5' 2" Short

6'9"

6'8"

6'2" Tall

5'11"

6'6" Tall

6'10" Tall

For classification, you always start with some labelled data points.

Classification = Deciding = Labelling

<i>5' 8" Short</i>	<i>6'2" Tall</i>
<i>5'11" Short</i>	<i>5'11" Short</i>
<i>5'8" Short</i>	<i>6'6" Tall</i>
<i>5' 2" Short</i>	<i>6'9" Tall</i>
<i>6'8" Tall</i>	<i>6'10" Tall</i>

If you were doing **analysis**, you'd ask a **human** to come up with a rule like:
A door below 6' is **Short** else it's **Tall**

Classification = Deciding = Labelling

<i>5' 8" Short</i>	
<i>5'11" Short</i>	<i>6'2" Tall</i>
	<i>5'11"</i>
<i>5'8"</i>	
<i>5' 2" Short</i>	<i>6'6" Tall</i>
<i>6'9"</i>	
<i>6'8"</i>	<i>6'10" Tall</i>

In **ML**, you just provide some examples.
The **computer** discovers the rule.

Classification = Deciding = Labelling

<i>5' 8" Short</i>	
<i>5'11" Short</i>	<i>6'2" Tall</i>
	<i>5'11"</i>
<i>5'8"</i>	
<i>5' 2" Short</i>	<i>6'6" Tall</i>
<i>6'9"</i>	
<i>6'8"</i>	<i>6'10" Tall</i>

The ML algorithm learns something like:
A door below 6' is **Short** else it's **Tall**

Classification = Deciding = Labelling

<i>5' 8" Short</i>	<i>6'2" Tall</i>
<i>5'11" Short</i>	<i>5'11" Short</i>
<i>5'8" Short</i>	<i>6'6" Tall</i>
<i>5' 2" Short</i>	<i>6'9" Tall</i>
<i>6'8" Tall</i>	<i>6'10" Tall</i>

You will learn to create an ML algorithm
that learns something like this and that
works with text.

Topic Classification

The United Nations
Security Council today

Manchester United
beat Barca to reach

**Can you tell which is about Politics and which is
about Sports?**

Topic Classification

Politics

The United Nations
Security Council today

Sports

Manchester United
beat Barca to reach

**Can you train an ML algorithm to tell which is
about Politics and which is about Sports?**

**Yes, but you need to learn some probability
theory.**

What is a probability?

- Definitions
 - Outcome
 - Sample Space S
 - Event E
- Axioms of Probability
 - Axiom 1: $0 \leq P(E) \leq 1$
 - Axiom 2: $P(S) = 1$
 - Axiom 3: $P(\text{union } [E]) = \text{sum}[P(E)]$
 - for mutually exclusive events.
- Just Remember
 - A number between 0 and 1
 - Adds up to 1

What is a probability?

- A number between 0 and 1
- The sum of the probabilities on all outcomes is 1

Heads



Tails



- $P(\text{heads}) = 0.5$
- $P(\text{tails}) = 0.5$

Concept 1 - What is a joint probability?

- Chance of something happening and something else happening.
- Chance of something happening at the same time as something else.
- Chance of seeing the word Barack and the word Obama in a document.
- Chance of the document being about Sports and having the word ‘innings’ in it.

Concept 1 - What is a joint probability?

- Definitions
 - Outcomes of multiple (say 2) experiments
 - Sample Spaces S1 and S2
 - Events E and F
- Axioms of Probability
 - Axiom 1: $0 \leq P(EF) \leq 1$
 - Axiom 2: $P(S1 \cap S2) = 1$
 - Axiom 3: $P(\text{union } [EF]) = \text{sum}[P(EF)]$
 - for mutually exclusive events.
- Again just Remember
 - A number between 0 and 1
 - Adds up to 1 **for all combinations of events E and F**

Joint Probability Table

Combo	E	F	$P(EF)$
1	• T 	• T 	• 0.3
2	• T 	• H 	• 0.2
3	• H 	• T 	• 0.3
4	• H 	• H 	• 0.2
Total			• 1.0

Note, these are not coin tosses. Assume these two coins have been tossed 10000 times.
Can you tell which coin is biased?

P(E)

Combo	E	F	$P(E)$
1	• T 	• T 	• $0.3 + 0.2 = 0.5$
2	• T 	• H 	
3	• H 	• T 	• $0.3 + 0.2 = 0.5$
4	• H 	• H 	
Total			• 1.0

Reordered Joint Probability Table

Combo	E	F	P(EF)
1	• T 	• T 	• 0.3
3	• H 	• T 	• 0.3
2	• T 	• H 	• 0.2
4	• H 	• H 	• 0.2
Total			• 1.0

P(F)

Combo	E	F	P(F)
1	• T 	• T 	• $0.3 + 0.3 = 0.6$
3	• H 	• T 	
2	• T 	• H 	• $0.2 + 0.2 = 0.4$
4	• H 	• H 	
Total			• 1.0

Concept 2 - What is a conditional probability?

- Chance of something happening when you know that something else has happened.
- Chance of something happening given something else happens.
- Chance of seeing the word Barack in a document when you see the word Obama.
- Chance of the document being about Sports when you see the word ‘innings’ in it.

Concept 2 - What is a conditional probability?

- Conditional Probability
 - If $P(F) > 0$, then $P(E|F) = P(EF) / P(F)$
- Definitions
 - Outcome **given another outcome**
 - Sample Space S **conditioned on the outcome of another experiment**
 - Event E **given another event F**
- Axioms of Probability
 - Axiom 1: $0 \leq P(E|F) \leq 1$
 - Axiom 2: $P(S) = \text{sum over all } E \text{ of } P(E|F) = 1$
 - Axiom 3: $P(\text{union} [E|F]) = \text{sum} [P(E|F)]$
 - for mutually exclusive events E .

Conditional Probability Table

Combo	E	F	$P(EF) \text{ & } P(F)$	$P(E F) = P(EF)/P(F)$
1	• T 	• T 	• 0.3 & 0.6	• 0.5
2	• T 	• H 	• 0.2 & 0.4	• 0.5
3	• H 	• T 	• 0.3 & 0.6	• 0.5
4	• H 	• H 	• 0.2 & 0.4	• 0.5
Total			• 1.0	• 2.0 ??

Conditional Probability Table

Combo	E	F	$P(E)$	$P(E F) = P(EF)/P(F)$
1	• T 	• T 	• 0.5	• 0.5
2	• T 	• H 	• 0.5	• 0.5
3	• H 	• T 	• 0.5	• 0.5
4	• H 	• H 	• 0.5	• 0.5

Is there a meaning to the pattern in the last two columns?

Topic Classification

Politics

The United Nations

The United States and

Sports

Manchester United

Manchester and Barca

Start by taking some samples of documents on politics and some samples of sports documents.
We're using really short documents so you can do all the calculations manually & see that this ML algorithm really works!

Step 1: Learn Multinomial Probabilities

The United Nations

The United States and

Manchester United

Manchester and Barca

Politics

Sports

$$P(\text{and}/\text{Politics}) = 1/7$$

$$P(\text{The}/\text{Politics}) = 2/7$$

$$P(\text{United}/\text{Politics}) = 2/7$$

$$P(\text{Nations}/\text{Politics}) = 1/7$$

$$P(\text{States}/\text{Politics}) = 1/7$$

$$P(\text{Politics}) = 7/12$$

$$P(\text{and}/\text{Sports}) = 1/5$$

$$P(\text{Manchester}/\text{Sports}) = 2/5$$

$$P(\text{United}/\text{Sports}) = 1/5$$

$$P(\text{Barca}/\text{Sports}) = 1/5$$

$$P(\text{Sports}) = 5/12$$

One ML Algorithm

Step 1 was easy!!!

Are you ready for Step 2 ?

Step 2: There's no step 2

This is a Naïve Bayesian Classifier!!!

$$P(\text{and}/\text{Politics}) = 1/7$$

$$P(\text{The}/\text{Politics}) = 2/7$$

$$P(\text{United}/\text{Politics}) = 2/7$$

$$P(\text{Nations}/\text{Politics}) = 1/7$$

$$P(\text{States}/\text{Politics}) = 1/7$$

$$P(\text{Politics}) = 7/12$$

$$P(\text{and}/\text{Sports}) = 1/5$$

$$P(\text{Manchester}/\text{Sports}) = 2/5$$

$$P(\text{United}/\text{Sports}) = 1/5$$

$$P(\text{Barca}/\text{Sports}) = 1/5$$

$$P(\text{Sports}) = 5/12$$

One ML Algorithm

Let's put the classifier to work:

Let's see if it can classify the following
documents :

1. United Nations
2. Manchester United

We are using deliberately short documents!

Running the Topic Classifier

United Nations

$P(\text{Sports} | \text{United Nations})$

$$= P(\text{United} | S) * P(\text{Nations} | \text{Sports}) * P(\text{Sports})$$

$$= (1/5) * (0) * (5/12) = 0$$

$P(\text{Politics} | \text{United Nations})$

$$= P(\text{United} | P) * P(\text{Nations} | \text{Politics}) * P(\text{Politics})$$

$$= (2/7) * (1/7) * (7/12) = 1/(7*6)$$

Running the Topic Classifier

United Nations

$P(\text{Politics} \mid \text{United Nations})$

>

$P(\text{Sports} \mid \text{United Nations})$

So, the classifier has returned the category **POLITICS**

Running the Topic Classifier

Manchester United

$P(\text{Sports} | \text{Manchester United})$

$$= P(\text{Manchester} | S) * P(\text{United} | S) * P(\text{Sports})$$

$$= (2/5) * (1/5) * (5/12) = 2/(5 * 12)$$

$P(\text{Politics} | \text{Manchester United})$

$$= P(\text{Manchester} | P) * P(\text{United} | P) * P(\text{Politics})$$

$$= (0) * (2/7) * (7/12) = 0$$

Running the Topic Classifier

Manchester United

$P(\text{Sports} \mid \text{Manchester United})$

>

$P(\text{Politics} \mid \text{Manchester United})$

So, the classifier has returned the category **SPORTS**

Oh more on those zeros

In the slides on running the Naïve Bayesian classifier, the zero probabilities were in red because in a real classifier we would:

- a) Replace zero probabilities by a small non-zero value (this is called smoothing – read the Chen and Goodman paper for details)
- b) Instead of multiplying probabilities (which would lead to floating point underflow) we would add log probabilities.

Topic Classification

Politics

United Nations

Sports

Manchester United

We have successfully used an ML algorithm to tell us which document is about Politics and which is about Sports !!!

Can We Solve Other Problems?

Now, we have an ML tool in our toolkit – a Naïve Bayesian classifier.

Can We Solve Other Problems?

Now, we have an ML tool in our toolkit – a Naïve Bayesian classifier.

So, if we can represent a text analysis problem as a classification task, then we can solve it using ML.

Can We Solve Other Problems?

Now, we have an ML tool in our toolkit – a Naïve Bayesian classifier.

So, if we can represent a text analysis problem as a classification task, then we can solve it using ML.

So, let us learn how to represent text analysis problems as classification tasks.

Problem 1: Sentence Segmentation

The problem of identifying the end-points of sentences is called *sentence segmentation*. It can be reduced to a classification problem.

Sentence Segmentation

Yes, Mr. Anurag. You need a D.L. to drive.

Sentence segmentation

Classify each ‘.’, ‘!’ and ‘?’ into:

- 1) sentence terminator and
- 2) not a sentence terminator.

Sentence Segmentation

Hurray! We have turned the problem of sentence segmentation into a classification problem.

Once you have modeled a text analytics problem as an ML problem, there is one more step you need to perform to get a working solution.

Sentence Segmentation

FIND THE FEATURES!

Features

Which country's flag is this?



Features are the clues you need for good decision making.

Features

Which country's flag is this?



One important feature for solving this decision problem is colour.

Features for Topic Classification

For topic classification, the features are the **individual words** in the text.

$$P(\text{and}/\text{Politics}) = 1/7$$

$$P(\text{The}/\text{Politics}) = 2/7$$

$$P(\text{United}/\text{Politics}) = 2/7$$

$$P(\text{Nations}/\text{Politics}) = 1/7$$

$$P(\text{States}/\text{Politics}) = 1/7$$

$$P(\text{Politics}) = 7/12$$

$$P(\text{and}/\text{Sports}) = 1/5$$

$$P(\text{Manchester}/\text{Sports}) = 2/5$$

$$P(\text{United}/\text{Sports}) = 1/5$$

$$P(\text{Barca}/\text{Sports}) = 1/5$$

$$P(\text{Sports}) = 5/12$$

Sentence Segmentation

Yes, Mr. Anurag. You need a D.L. to drive.

What are the **features** you would use?

That's when you start reading the research papers!!!

Sentence Segmentation

Yes, Mr. Anurag. You need a D.L. to drive.

What are the **features** you would use?

Is_This_Character_a_Dot

Is_This_Within_Quotes

Is_Next_Letter_Capitalized

Is_Prev_Letter_Capitalized

Number_of_Words_in_Sentence_so_Far

Is_Next_Word_a_Name

Sentence Segmentation

Train an NB classifier on some text whose characters are marked as sentence terminators or not.

It will learn to assign characters to the categories **sentence terminator and **not a sentence terminator**.**

Problem 2: Tokenization

Now, get a D.L., Mr. Anurag.

Now , get a D.L. , Mr. Anurag .

For almost all analyses, you have to identify the individual words in the text.

How can you break a sentence into words?

Tokenization

Yes, Mr. Anurag. You need a D.L. to drive.

Tokenization

Classify each ‘.’, ‘!’, ‘;’, ‘‘’, ‘?’ etc. into:

- 1) word terminator and
- 2) not a word terminator.

Tokenization

We have turned the problem of word segmentation into a classification problem.

Train a classifier on text whose characters are marked as word terminators or not. It will learn to assign characters to the categories **word terminator** and **not a word terminator**.

Tokenization

Yes, Mr. Anurag. You need a D.L. to drive.

What are the **features** you would use?

Again, start reading the research papers.

Tokenization

Yes, Mr. Anurag. You need a D.L. to drive.

What are the **features** you would use?

Is_This_Character_a_Dot

Is_This_Within_Quotes

Is_Next_Letter_Capitalized

Is_Prev_Letter_Capitalized

Number_of_Words_in_Sentence_so_Far

Is_Next_Word_a_Name

Problem 2 an another guise: Word Segmentation

私の名前は山本です。

My name is Yamamoto.

私 の 名前 は 山本 です 。

In two of the CJK languages, the words in the text are written together without spaces.

How can you break a sentence into words?

Let's try tokenizing some Japanese text

私の名前は山本です。

My name is Yamamoto.

私 の 名前 は 山本 です 。

Can *Word Segmentation* be cast as a classification problem?

Word Segmentation As Classification

Take every boundary between two characters

私|の|名|前|は|山|本|で|す|。

Classify the boundary into “**word separator**” and
“**not word separator**”.

私|の|名|前|は|山|本|で|す|。

Replace “**word separators**” with spaces:

私 の 名前 は 山本 です 。

Word Segmentation

私の名前は山本です。

What are the **features** you would use?

Word Segmentation

私の名前は山本です。

You usually need a dictionary of words in the language.

A very important boolean feature is, can I see the letters after the last word boundary in the dictionary.

Time for Exercises!

Let's do some exercises on feature engineering!

1. Gender classification
2. Language classification
3. Topic classification

Problem 3: Identifying Stop Words

Stop words: In computing, **stop words** are words which are filtered out before or after processing of natural language data (text). (wikipedia)

Stop words are words which do not contain much semantic information. They mostly serve syntactic functions.

Generally, the grammatical category of words called **function words** (or the **closed category**) are considered **stop words**.

Some Linguistics

Word categories:

1. Noun
2. Verb
3. Adjective
4. Adverb
5. Interjection
6. Conjunction (and, or, if, neither ... nor)
7. Pronoun (I, it, you, them)
8. Preposition (in, of, out)

Some Linguistics

Word categories:

1. Noun
 2. Verb
 3. Adjective
 4. Adverb
 5. Interjection
 6. Conjunction (and, or, if, neither ... nor)
 7. Pronoun (I, it, you, them)
 8. Preposition (in, of, out)
- Open Category or
Content Words

Some Linguistics

Word categories:

1. Noun
2. Verb
3. Adjective
4. Adverb
5. Interjection
6. Conjunction (and, or, if, neither ... nor)
7. Pronoun (I, it, you, them) Closed Category or
8. Preposition (in, of, out) Function Words

Identify the function words in the following text:

Function words are words that have little lexical meaning or have ambiguous meaning, but instead serve to express grammatical relationships with other words within a sentence, or specify the attitude or mood of the speaker.

1. Conjunction (and, or, if, neither ... nor)
2. Pronoun (I, it, you, them)
3. Preposition (in, of, out)

Function Words

Identify the function words in the following text:

Function words are words **that** have little lexical meaning **or** have ambiguous meaning, **but** instead serve **to** express grammatical relationships **with other** words **within** a sentence, **or** specify **the** attitude **or** mood **of the** speaker.

1. Conjunction (and, or, if, neither ... nor)
2. Pronoun (I, it, you, them)
3. Preposition (in, of, out)

Function Words

If I am looking at text in an unknown language, how can I identify function words?

Function words are words **that** have little lexical meaning **or** have ambiguous meaning, **but** instead serve **to** express grammatical relationships **with other** words **within** a sentence, **or** specify **the** attitude **or** mood **of the** speaker.

1. Conjunction (and, or, if, neither ... nor)
2. Pronoun (I, it, you, them)
3. Preposition (in, of, out)

Function Words

If I am looking at text in an unknown language, how can I identify function words?

Function words are words **that** have little lexical meaning **or** have ambiguous meaning, **but** instead serve **to** express grammatical relationships **with other** words **within** a sentence, **or** specify **the** attitude **or** mood **of the** speaker.

Is this a good rule:

“Any word after a comma is a function word”?

If I am looking at text in an unknown language, how can I identify function words?

Function words are words **that** have little lexical meaning **or** have ambiguous meaning, **but** instead serve **to** express grammatical relationships **with other** words **within** a sentence, **or** specify **the** attitude **or** mood **of the** speaker.

“Any word after a comma is a function word”?

‘It was a cold, cold day in December.’

The most frequent words ...

Courtesy <http://www.wordcount.org/main.php>

The, of, and, to, a, in, that, it, is, was, i, for, on, you, he, be, with, as, by, at, have, are, this, not, but, had, his, they, from, she, which, or, we, an, there, her, were, one, do, been, all, their, has, would, will, what, if, can, when, so, no, said, who, more, about, up, them, some, could, him, into, its, then, two, out, ...

1. Conjunction (and, or, if, neither ... nor)
2. Pronoun (I, it, you, them)
3. Preposition (in, of, out)

Function Words

Statistical Approach

Counting (frequencies / statistics) can tell you things that rules cannot ...

So, we use probabilities and statistics to aid in decision making about text ... instead of rules like “I before E except after C”

Statistical Approach

The most frequent words approach is derived from Zipf's Law:

The Principle of Least Effort

“People will act so as to minimize their probably average rate of work”

The frequency of a word is inversely proportional to the length of the word

Problem 3 continued: Part of Speech (POS) Tagging

Anurag needs a D.L.

*Anurag/**NNP** needs/**VBZ** a/**DT** D.L./**NN***

How do you tag the words in a sentence?

Problem 3: Part of Speech (POS) Tagging

You turn the problem of POS Tagging into a classification problem where you label each word as:

1. Noun
2. Verb
3. Adjective
4. Adverb
5. Interjection
6. Conjunction (and, or, if, neither ... nor)
7. Pronoun (I, it, you, them)
8. Preposition (in, of, out)

Problem 3: Part of Speech (POS) Tagging

Anurag needs a D.L.

*Anurag/**NNP** needs/**VBZ** a/**DT** D.L./**NN***

POS Tagging

Run from the first to the last word in the sentence, classifying each word in the sentence into a part-of-speech category.

Problem 3: Part of Speech (POS) Tagging

I will run /VBZ

I was in the run /NN

POS Tagging as sequential classification

Why did the POS tagger decide differently about the word ‘run’ in these two cases?

Problem 3: Part of Speech (POS) Tagging

I/PRP will/MD run/VB

I/PRP was/VBD in/IN the/DT run/NN

POS Tagging as sequential classification

Why did the POS tagger decide differently about the word ‘run’ in these two cases?

Answer: The previous decision!!!

Sequential Classification

What is a Sequential Classifier?

Something that performs **classification** on an ordered collection, such that each decision depends on previous decision(s).

Classification = categorizing

Classification = deciding

Classification = labelling

Classifying = Labelling = Deciding

Sequential Classification

We have our first design pattern:

Sequential classification

When do you use it?

Sequence of decisions where every decision depends on an earlier decision.

Problem 4: Sentiment Classification

John loves to use Canon cameras.

?

Problem 4: Sentiment Classification

John loves to use Canon cameras.

positive

Sentiment Classification

Sentence-Level Sentiment: Classify the whole sentence into **positive, negative, neutral**

Sentiment Classification

- Café XYZ has a splendid selection of cakes.

Is this positive or negative?

What word clues you in to the polarity?

Words

- **Adjectives** Hatzivassiloglou & McKeown 1997, Wiebe 2000, Kamps & Marx 2002, Andreevskaia & Bergler 2006
 - positive: **honest important mature large patient**
 - Ron Paul is the only **honest** man in Washington.
 - Kitchell's writing is unbelievably **mature** and is only likely to get better.
 - To humour me my **patient** father agrees yet again to my choice of film

From Slides of Carmen Banea

Words

- Adjectives
 - negative: harmful hypocritical inefficient insecure
 - It was a macabre and hypocritical circus.
 - Why are they being so inefficient ?

From Slides of Carmen Banea

Sentiment Classification

- I like XYZ café very much.

Is this positive or negative?

What word clues you in to the polarity?

Words

- Other parts of speech Turney & Littman 2003, Riloff, Wiebe & Wilson 2003, Esuli & Sebastiani 2006
 - Verbs
 - positive: **praise, love**
 - negative: **blame, criticize**

From Slides of Carmen Banea

Sentiment Classification

- It was a pleasure to have a cake at café XYZ.

Is this positive or negative?

What word clues you in to the polarity?

Words

- Other parts of speech Turney & Littman 2003, Riloff, Wiebe & Wilson 2003, Esuli & Sebastiani 2006
 - Nouns
 - positive: pleasure, enjoyment
 - negative: pain, criticism

From Slides of Carmen Banea

Sentiment Classification

- B. Pang, L. Lee, and S. Vaithyanathan,
“Thumbs up? Sentiment classification using
machine learning techniques,” in *Proceedings
of the Conference on Empirical Methods in
Natural Language Processing (EMNLP) 2002*

Challenge: Sentiment Classification

Is this enough in the real world?

Can you use the above classifier in real life?

Challenge: Subjectivity Classification

John loves to use Canon cameras.

subjective

Subjectivity Classification

Sentence-Level Sentiment: Classify the whole sentence into **subjective, objective**

Subjectivity Detection

- I am only trying to protect my good name

Is this positive or negative?

Is this subjective or objective?

Subjectivity Detection

- B. Pang and L. Lee, “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts,” in *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 271–278, 2004.

Subjectivity Detection / Sentiment Classification

Sentiment Analysis	Subjectivity analysis
Positive	Subjective
Negative	
Neutral	Objective

From Slides of Carmen Banea

Time for Exercises!

Let's practise!

1. POS tagging
2. Sentiment classification
3. Subjectivity classification

Finer Granularity

We looked at

Document level sentiment classification.

Sentence level sentiment classification.

Can we get more granular than that?

Feature Level Sentiment Analysis

- M. Hu and B. Liu, “Mining opinion features in customer reviews,” in *Proceedings of AAAI*, pp. 755–760, 2004.
- A.-M. Popescu and O. Etzioni, “Extracting product features and opinions from reviews,” in *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005.

Feature Level Sentiment Analysis

- I like the red Toyota in the parking lot but think the green one is rather ugly.
 - + red Toyota
 - green one

Phrase Level Sentiment Analysis Step 1

John loves to use Canon cameras.

John = Holder

Loves = Sentiment

Canon cameras = Object

Step 1:

Sentiment Analysis

Phrase-Level Sentiment: Extract entities holder, object, sentiment

Phrase Level Sentiment Analysis Step 2

John loves to use Canon cameras.

{**John = Holder, Loves = Sentiment, Canon cameras = Object**} => positive sentiment relation

Step 2:

Sentiment Analysis

Phrase-Level Sentiment: Extract relations between
{holder, object, sentiment}

Sentiment Analysis

Good overview in 2009 OMSA article:
“Opinion Mining and Sentiment
Analysis” by Pang and Lee

Problem 5: Named Entity Recognition

Anurag is looking for a Hyundai car in Bangalore

Anurag = Person

Hyundai car = Vehicle

Bangalore = Place

How do you extract the named entities in a sentence using a classifier?

Problem 5: Named Entity Recognition

Anurag is looking for a Hyundai car in Bangalore

Anurag = Person

Hyundai car = Vehicle

Bangalore = Place

Anurag/Person is/Other looking/Other
for/Other a/Other Hyundai/Vehicle car/Vehicle
in/Other Bangalore/Place

Problem 5: Named Entity Recognition

We have turned the problem of Named Entity Recognition (NER) into a Tagging problem where you label each word as:

1. Person
2. Vehicle
3. Place

Problem 5: Named Entity Recognition

But we know that we can turn the problem of Tagging into a Classification problem over the same labels:

1. Person
2. Vehicle
3. Place

Problem 5: Named Entity Recognition

Anurag is looking for a Hyundai car in Bangalore

Anurag/Person is/Other looking/Other
for/Other a/Other Hyundai/Vehicle car/Vehicle
in/Other Bangalore/Place

Named Entity Recognition

Run from the first to the last word in the sentence, classifying each word in the sentence into a Named Entity Recognition category.

Hey We Have One More ML Tool

We just built one more very useful ML tool!

The Extractor ☺

Extraction

What is an Extractor?

Something that performs extraction.

Extraction = recognizing

Extraction = finding

Extraction = locating

Extraction = Finding = Locating

Extraction

We have our second design pattern:

Extraction

When do you use it?

When the number of choices is infinite.

Problem 6: Relation Extraction

Tim Cook is the new CEO of Apple Computers

Relation: **CEO_of**

Tim Cook (Person) **Apple Computers (Org)**

How do you identify relations between entities in a sentence using a classifier?

Relation Extraction

Text: Tim Cook is the new CEO of Apple Computers

Step 1:

Analysis: Tim/Person Cook/Person is the new CEO
of Apple/Org Computers/Org

Relation Extraction

Text: Tim Cook is the new CEO of Apple Computers

Step 1:

Analysis: Tim/Person Cook/Person is the new CEO
of Apple/Org Computers/Org

Step 2:

Relation Extraction

{Tim/Person Cook/Person, Apple/Org Computers/Org} =>
CEO of

Relation Extraction

Text: Tim Cook is the new CEO of Apple Computers

Step 1:

Analysis: Tim/Person Cook/Person is the new CEO
of Apple/Org Computers/Org

Step 2:

Relation Extraction

Run through all pairs of named entities, classifying each pair into **CEO_of** or **Other**.

But look at what we're extracting!

Do you realize what we're extracting?

Meaning !!! 😊

Extracting Meaning

Semantic Analysis: I/Person am looking for a
Hyundai/Vehicle car/Vehicle in Bangalore/Place



Syntactic Analysis: I/Pronoun am/BE looking/V
for a/DT Hyundai/NP car/NN in/PP Bangalore/NP



Text: I am looking for a Hyundai car in B'llore

But is this comprehensive?

Are you telling me that this is all I have to do to extract every possible sort of simple meaning?

Yeah!!! 😊

All utterances fall into two main categories ...

Two Kinds of Utterances

Intentional: I want to buy a computer.

Information: There was heavy snowfall in Sikkim.

How do you deal with intentional utterances

I want to buy a computer.

Intention Analysis

Raw Text: Are you sad that Steve Jobs died?

Analysis: This person is inquiring about someone's emotions concerning Steve Jobs

Intention Holder: I

Intention: inquire

Sentiment Analysis

Raw Text: I am sad that Steve Jobs died

Analysis: This person holds a positive opinion
on Steve Jobs

Sentiment Holder:

I

Object of Sentiment:

Steve Jobs

Polarity of Sentiment:

positive

How do you deal with informational utterances

There was heavy snowfall in Sikkim.

Event Analysis

Raw Text: There is heavy snowfall in Sikkim.

Analysis: Snowfall event

Event:

snowfall

Fact Analysis

Raw Text: Bangalore is the capital of K'taka

Analysis: capital_of relation exists

Entity: **Bangalore/Place**

Karnataka/Place

Bangalore capital_of K'taka

Entity Extraction

Text: I am looking for a Hyundai car in Bangalore

Semantic Analysis: I/Person am looking for a
Hyundai/Thing car/Thing in Bangalore/Place

Entities:

I

Person

Hyundai car

Thing

Bangalore

Place

Relation Extraction

Text: Tim Cook is the new CEO of Apple Computers

Analysis: Tim/Person Cook/Person is the new CEO
of Apple/Org Computers/Org

Relation: CEO_of
Tim Cook (Person)

Apple Computers(Org)

Converting Unstructured to Structured Data

Read the whitepaper:

http://aiaioo.com/whitepapers/text_analytics_360.pdf

Problem 6 again: Question Answering

Who is the CEO of Apple Computers?

Relation: **CEO_of**

? (Person)

Apple Computers (Org)

How do you answer a factoid question?

Problem 6 again: Question Answering

Who is the CEO of Apple Computers?

Relation: **CEO_of**

? **(Person)**

Apple Computers (Org)

Factoid question answering can be modeled using relation extraction.

Problem 6 again: Question Answering

Who is the CEO of Apple Computers?

Relation: **CEO_of**

? **(Person)**

Apple Computers (Org)

A factoid question specifies a relation and one of the entities.

Problem 6 again: Question Answering

Fetch this document using IR:

Tim Cook is the new CEO of Apple Computers

You've already developed a way to mark it as:

Relation: **CEO_of**

Tim Cook (Person) **Apple Computers (Org)**

*Since you have the relation and the query entity,
you just have to return the requested entity*

So, I can do anything (badly?)

I got it! I got it! I can do anything.

But how do I know how well I am doing it ? 😊

Measurement

Measurement:

This **measurement** thing is very important in any design process, because ...

Measurement

Measurement:

... it lets you compare two designs and decide which is better.

How do you measure the performance of a classifier?

Before we go on ...

... promise me ...

**... that you will never forget what I am about to
tell you ... 😞**

Measurement

Break up the data points into training and test parts (usually an 80:20 split) and never test on your training data.

Measurement

Break up the data points into training and test – usually an 80:20 split.

Train on 80%

Test on the remaining 20%

Why not test on the training data?

Measurement

Or if you are still developing features,
break up the data points into training,
development and test – usually a
70:10:20 split.

Train on 70%

Develop on 10%

Test on the remaining 20%

Why not develop on the test data?

That applies to college as well ...

... you won't get accurate measurements ...

**... if you test students using questions that
appeared in the question bank!**

**... and you will be encouraging students to learn
things by rote!**

Classification Quality Metric - Accuracy

Correct Answers

Total Number of Questions

Classification Quality Metric - Accuracy

If your categories are **Politics** and **Sports**

Politics Documents classified as **Politics**
+ **Sports Documents** classified as **Sports**

Total Number of Documents

Classification Quality Metric - Accuracy

Point of View = Sports = (+ve)

	Gold - Politics	Gold – Sports
Observed - Politics	TN (True Negative)	FN (False Positive)
Observed - Sports	FP (False Negative)	TP (True Positive)

$$A(M) = \frac{TN + TP}{TN + FP + FN + TP}$$

Accuracy

Point of View - Politics

	Gold – Sports (1000)	Gold – Politics (1000)
Observed – Sports	TN = 990	FN = 100
Observed – Politics	FP = 10	TP = 900

$$A(M) = \frac{TN + TP}{TN + FP + FN + TP} = ?$$

Accuracy

Point of View - Politics

	Gold – Sports (1000)	Gold – Politics (1000)
Observed – Sports	TN = 990	FN = 100
Observed – Politics	FP = 10	TP = 900

$$A(M) = \frac{TN + TP}{TN + FP + FN + TP} = 94.5\%$$

Classification Quality Metric - Recall

How many Politics document did it find

Total number of Politics documents
in the test data

Recall

Point of View - Politics

	Gold – Sports (1000)	Gold – Politics (1000)
Observed – Sports	TN = 990	FN = 100
Observed – Politics	FP = 10	TP = 900

$$\text{Recall} = \frac{tp}{tp + fn} = ?$$

Recall

Point of View - Politics

	Gold – Sports (1000)	Gold – Politics (1000)
Observed – Sports	TN = 990	FN = 100
Observed – Politics	FP = 10	TP = 900

$$\text{Recall} = \frac{tp}{tp + fn} = 90\%$$

Recall

Point of View - Sports

	Gold – Sports (1000)	Gold – Politics (1000)
Observed – Sports	TP = 990	FP = 100
Observed – Politics	FN = 10	TN = 900

$$\text{Recall} = \frac{tp}{tp + fn} = ?$$

Recall

Point of View - Sports

	Gold – Sports (1000)	Gold – Politics (1000)
Observed – Sports	TP = 990	FP = 100
Observed – Politics	FN = 10	TN = 900

$$\text{Recall} = \frac{tp}{tp + fn} = 99\%$$

Classification Quality Metric - Precision

How many were really Politics documents

Total number of documents the classifier identified as Politics

Precision

Point of View - Politics

	Gold – Sports (1040)	Gold – Politics (960)
Observed – Sports	TN = 990	FN = 10
Observed – Politics	FP = 50	TP = 950

$$\text{Precision} = \frac{tp}{tp + fp} = ?$$

Precision

Point of View - Politics

	Gold – Sports (1040)	Gold – Politics (960)
Observed – Sports	TN = 990	FN = 10
Observed – Politics	FP = 50	TP = 950

$$\text{Precision} = \frac{tp}{tp + fp} = 95\%$$

Precision

Point of View - Sports

	Gold – Sports (1040)	Gold – Politics (960)
Observed – Sports	TP = 990	FP = 10
Observed – Politics	FN = 50	TN = 950

$$\text{Precision} = \frac{tp}{tp + fp} = ?$$

Precision

Point of View - Sports

	Gold – Sports (1040)	Gold – Politics (960)
Observed – Sports	TP = 990	FP = 10
Observed – Politics	FN = 50	TN = 950

$$\text{Precision} = \frac{tp}{tp + fp} = 99\%$$

Metrics for Measuring Classification Quality

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

F-Score = the harmonic mean of Precision and Recall

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

F-Score Generalized

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}$$

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

F-Score Generalized

$$F_{\beta} = (1 + \beta^2) \frac{P * R}{\beta^2 * P + R}$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

F-Score Generalized

$$F_{\beta} = (1 + \beta^2) \frac{P * R}{\beta^2 * P + R}$$

$$\alpha = \frac{1}{1 + \beta^2}$$

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

Precision and Recall Tradeoff?

	Gold Class 1	Gold Class 2
Observed Class 1	TP	FP
Observed Class 2	FN	TN

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$



Precision, Recall, Average, F-Score

	Precision	Recall	Average	F-Score
Classifier 1	50%	50%	50%	?
Classifier 2	30%	70%	50%	?
Classifier 3	10%	90%	50%	?

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

What is the sort of classifier that fares worst?

Precision, Recall, Average, F-Score

	Precision	Recall	Average	F-Score
Classifier 1	50%	50%	50%	50%
Classifier 2	30%	70%	50%	42%
Classifier 3	10%	90%	50%	18%

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

What is the sort of classifier that fares worst?

Measurement Exercise

Data Point	Real	Observed
D1	Politics	Sports
D2	Politics	Politics
D3	Politics	Politics
D4	Sports	Politics
D5	Sports	Sports
D6	Sports	Sports
D7	Sports	Politics
D8	Sports	Sports

What is the accuracy?

Measurement Exercise

Data Point	Real	Observed
D1	Politics	Sports
D2	Politics	Politics
D3	Politics	Politics
D4	Sports	Politics
D5	Sports	Sports
D6	Sports	Sports
D7	Sports	Politics
D8	Sports	Sports

What is the precision for Sports?

Measurement Exercise

Data Point	Real	Observed
D1	Politics	Sports
D2	Politics	Politics
D3	Politics	Politics
D4	Sports	Politics
D5	Sports	Sports
D6	Sports	Sports
D7	Sports	Politics
D8	Sports	Sports

What is the recall for Sports?

Measurement Exercise

Data Point	Real	Observed
D1	Politics	Sports
D2	Politics	Politics
D3	Politics	Politics
D4	Sports	Politics
D5	Sports	Sports
D6	Sports	Sports
D7	Sports	Politics
D8	Sports	Sports

What is the F-score for Sports?

Time for Exercises!

Let's practise measurement!

1. Precision
2. Recall
3. F-Score
4. Confusion Matrices

Comparing Classifiers

Classification does not have to be about ML.

You can make decisions (classify) using rules.

There are also many kinds of ML.

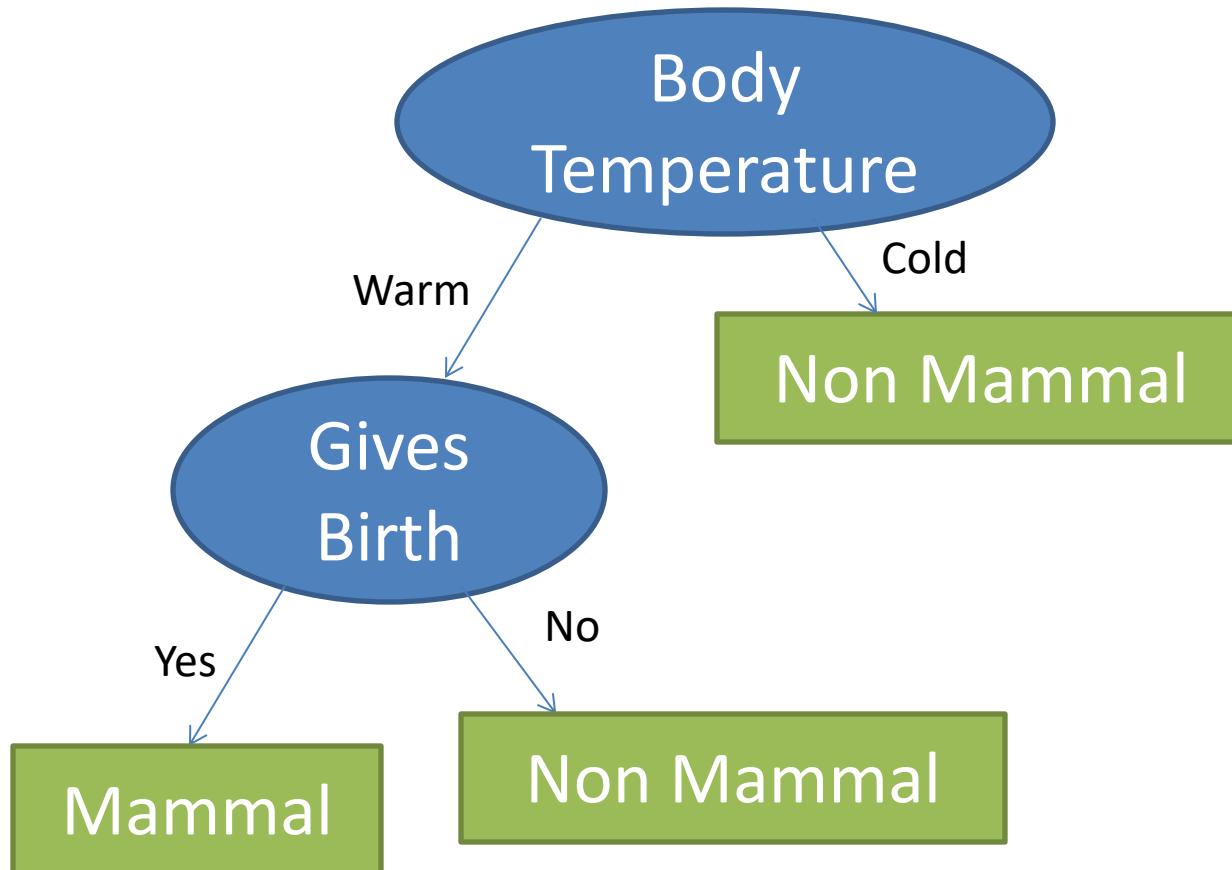
Now, we have a way to compare them.

Let's Look at these Types of Classifiers

- Rule-based classifiers
- Word Lists (Gazetteers)
- Artificial Neural Networks (Perceptrons)
- Naïve Bayesian Classifier
- Maximum Entropy Classifiers
- Support Vector Machines

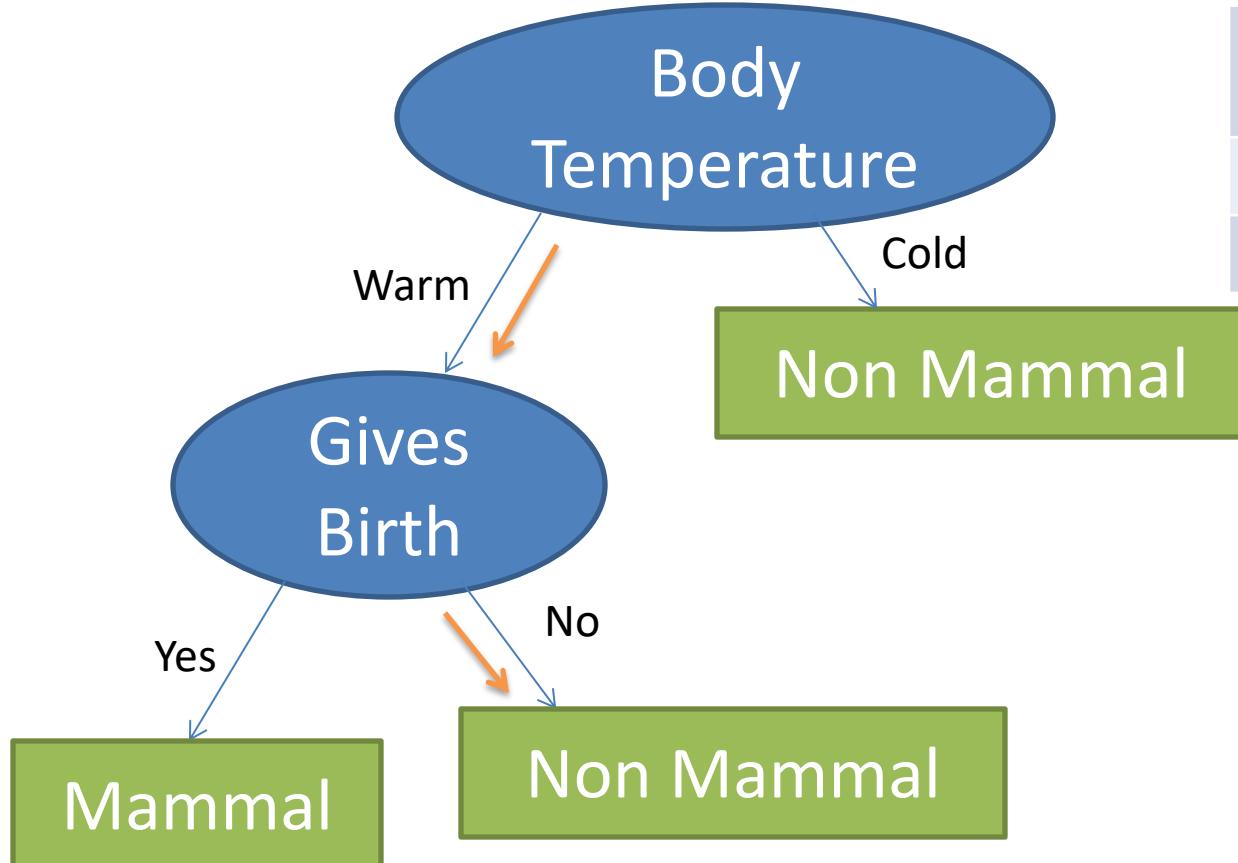
Rule Based Classification

Decision Tree Classifier



Decision Tree Classifier

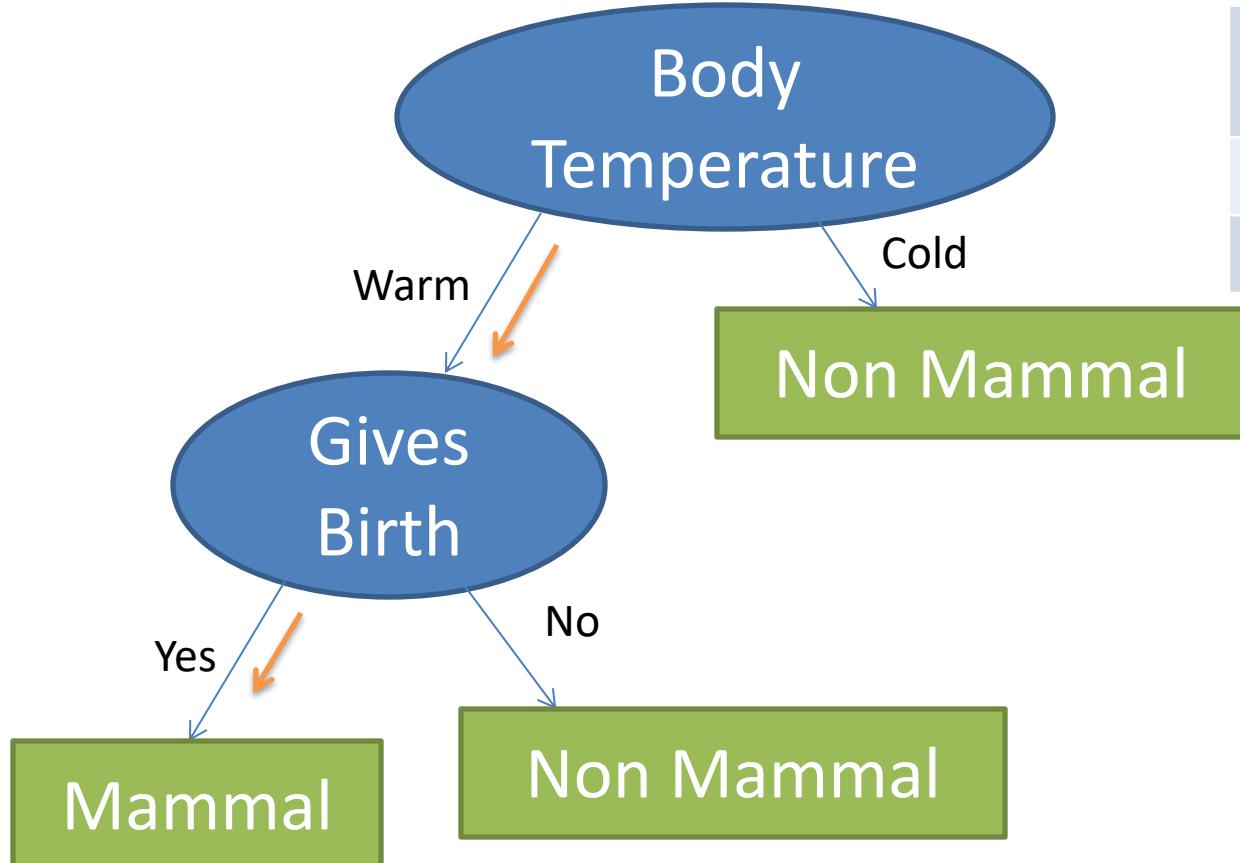
Classifying a Data Point



Data Point	Flamingo
Body Temperature	Warm
Gives Birth	No
Class	?

Decision Tree Classifier

Classifying a Data Point



Data Point	Dog
Body Temperature	Warm
Gives Birth	Yes
Class	?

Decision Tree Classifier

Learning to Classify from Examples

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Decision Tree Classifier

Learning to Classify from Examples

Hunt's Algorithm

Defaulted = No

Decision Tree Classifier

Learning to Classify from Examples

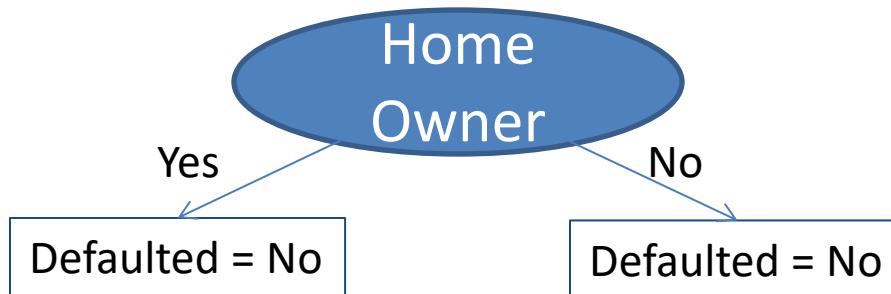
Now pick a splitting feature, let's say “Home Owner”



Decision Tree Classifier

Learning to Classify from Examples

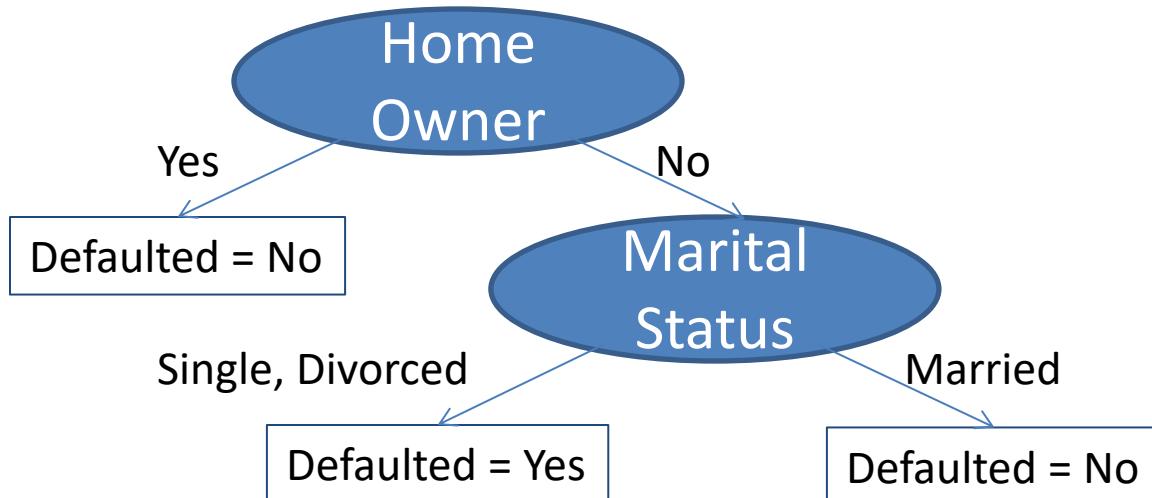
Hunt's Algorithm



Decision Tree Classifier

Learning to Classify from Examples

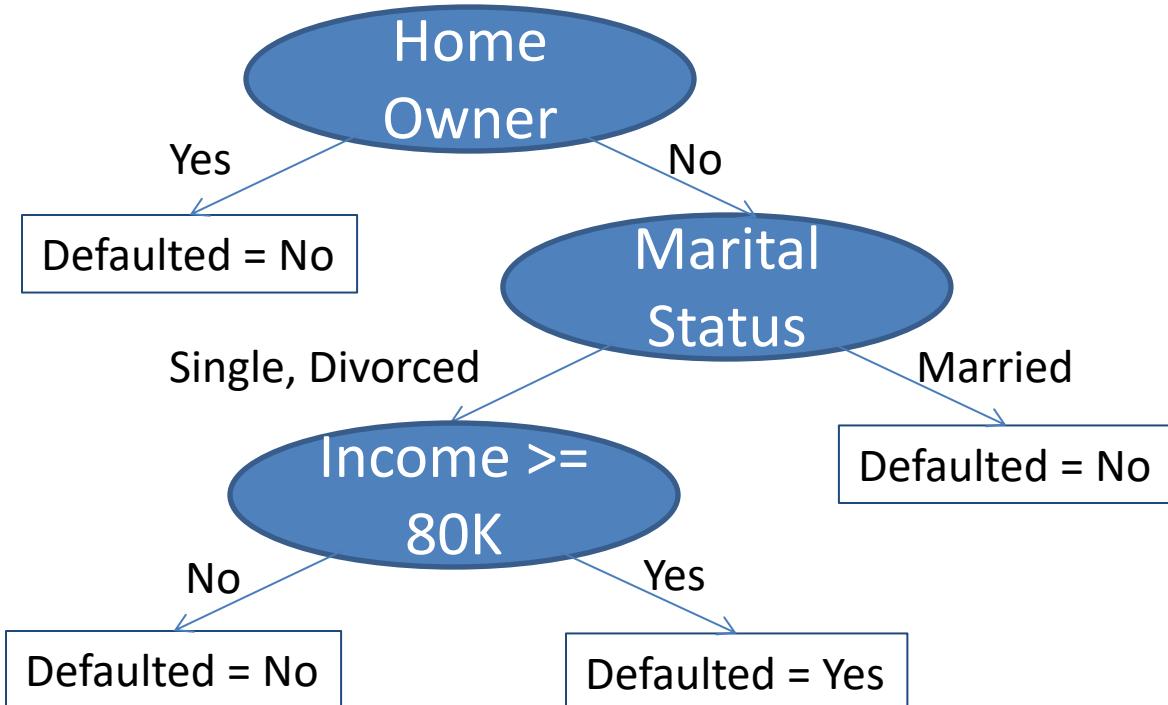
Hunt's Algorithm



Decision Tree Classifier

Learning to Classify from Examples

Hunt's Algorithm



How Do You Choose The Next Splitting Feature

Impurity Metrics

Entropy

Gini Coefficient

Classification Error

How Do You Choose The Next Splitting Feature

Impurity Metrics

$\text{Entropy}(t) = - \sum \text{over all classes of } [p(i|t) \log p(i|t)]$

$\text{Gini}(t) = 1 - \sum \text{over all classes of } [p(i|t)^2]$

$\text{Classification Error}(t) = 1 - \max \text{over all classes of } [p(i|t)]$

Impurity Metric Example

$\text{count(Class1)} = 0, \text{count(Class2)} = 6$

$\text{Entropy}(t) = - 0/6 * \log 0/6 - 6/6 * \log 6/6$

$\text{Gini}(t) = 1 - (0/6)^2 - (6/6)^2$

$\text{Classification Error}(t) = 1 - \max[0/6, 6/6]$

Impurity Metric Example

$\text{count(Class1)} = 1, \text{count(Class2)} = 5$

$\text{Entropy}(t) = - 1/6 * \log 1/6 - 5/6 * \log 5/6 = 0.650$

$\text{Gini}(t) = 1 - (1/6)^2 - (5/6)^2 = 0.278$

$\text{Classification Error}(t) = 1 - \max[1/6, 5/6] = 0.167$

Impurity Metric Example

$\text{count(Class1)} = 3, \text{count(Class2)} = 3$

$\text{Entropy}(t) = -3/6 * \log 3/6 - 3/6 * \log 3/6 = 1$

$\text{Gini}(t) = 1 - (3/6)^2 - (3/6)^2 = 0.5$

$\text{Classification Error}(t) = 1 - \max[3/6, 3/6] = 0.5$

How Do You Choose The Next Splitting Feature

Use an Impurity Metric: Entropy, Gini or Classification Error

Pick the attribute that decreases the impurity metric the most

Example: Choosing Splitting Features

Features:

size: small medium large

colour: red blue green

shape: brick wedge sphere pillar

%% yes

medium blue brick

small red sphere

large green pillar

large green sphere

%% no

small red wedge

large red wedge

large red pillar

Reference: Bratko sections 18.5, 18.6

Example: Choosing Splitting Features

Initial decision tree is one node with all examples.

There are 4 positive examples and 3 negative examples
i.e. probability of positive is $4/7 = 0.57$; probability of
negative is $3/7 = 0.43$

Entropy is: $-(0.57 * \log 0.57) - (0.43 * \log 0.43) = 0.99$

%% yes
medium blue brick
small red sphere
large green pillar
large green sphere

%% no
small red wedge
large red wedge
large red pillar

Reference: Bratko sections 18.5, 18.6

How Do You Choose The Next Splitting Feature

Suppose we split on size?

Try split on *size* which has three values:
large, *medium* and *small*.

Example: Choosing Splitting Features

There are four instances with **size = large**.

There are **two large positive** examples and **two large negative** examples.

The probability of positive is 0.5

The entropy is: $-(0.5 * \log 0.5) - (0.5 * \log 0.5) = 1$

%% yes

medium blue brick

small red sphere

large green pillar

large green sphere

%% no

small red wedge

large red wedge

large red pillar

Reference: Bratko sections 18.5, 18.6

Example: Choosing Splitting Features

There is 1 instance with **size = medium**.

There is **1 medium positive example** and **0 medium negative examples**.

The probability of positive is 1

The entropy is: $-(1 * \log 1) - (0 * \log 0) = 0$

%% yes

medium blue brick

small red sphere

large green pillar

large green sphere

%% no

small red wedge

large red wedge

large red pillar

Reference: Bratko sections 18.5, 18.6

Example: Choosing Splitting Features

There are 2 instances with **size = small**.

There is **1 medium positive example** and **1 medium negative example**.

The probability of positive is 0.5

The entropy is: $-(0.5 * \log 0.5) - (0.5 * \log 0.5) = 1$

%% yes
medium blue brick
small red sphere
large green pillar
large green sphere

%% no
small red wedge
large red wedge
large red pillar

Reference: Bratko sections 18.5, 18.6

Example: Choosing Splitting Features

Initial decision tree was one node with all examples.

There are 4 positive examples and 3 negative examples.

After splitting on size, we got 3 nodes with 4, 2 and 1 examples. So, the expected information is:

$$\left(1 \times \frac{4}{7}\right) + \left(1 \times \frac{2}{7}\right) + \left(0 \times \frac{1}{7}\right) = 0.86$$

%% yes

medium blue brick

small red sphere

large green pillar

large green sphere

%% no

small red wedge

large red wedge

large red pillar

Reference: Bratko sections 18.5, 18.6

Example: Choosing Splitting Features

Initial decision tree was one node with entropy 0.99.
After splitting on size, we got 3 nodes with 4, 2 and 1 examples with expected information:

$$\left(1 \times \frac{4}{7}\right) + \left(1 \times \frac{2}{7}\right) + \left(0 \times \frac{1}{7}\right) = 0.86$$

So, the information gain is: $0.99 - 0.86 = 0.13$

%% yes

medium blue brick

small red sphere

large green pillar

large green sphere

%% no

small red wedge

large red wedge

large red pillar

Reference: Bratko sections 18.5, 18.6

How Do You Choose The Next Splitting Feature

Suppose we split on colour?

Try split on *colour* which has three values:
blue, red and green.

Colour has an information gain of 0.52

So, colour is a better parameter to split on than size.

How Do You Choose The Next Splitting Feature

Suppose we split on shape?

Try split on *shape* which has four values:
brick, *wedge*, *sphere* and *pillar*.

Shape has an information gain of 0.7

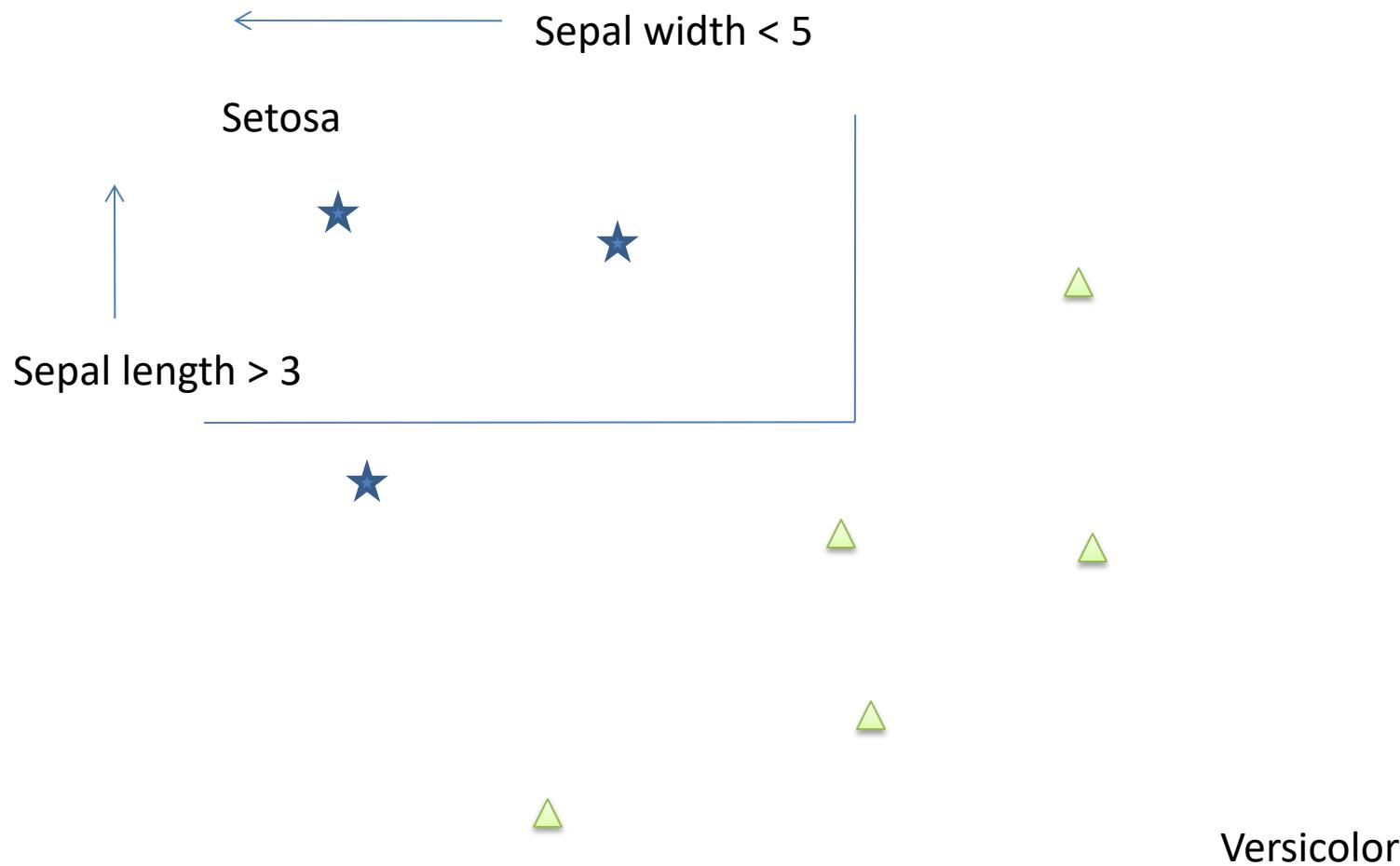
So, shape works better than colour or size.

Therefore split on shape.

Rule-Based Classifiers

- Rules such as the following are mathematically equivalent to a Decision Tree
- Example:
 - If `sepal_length > 3` and `sepal_width < 5` then
class=setosa else class=versicolor
- If the only features are sepal length and width (2 features), you can draw a line in 2D separating the two classes.

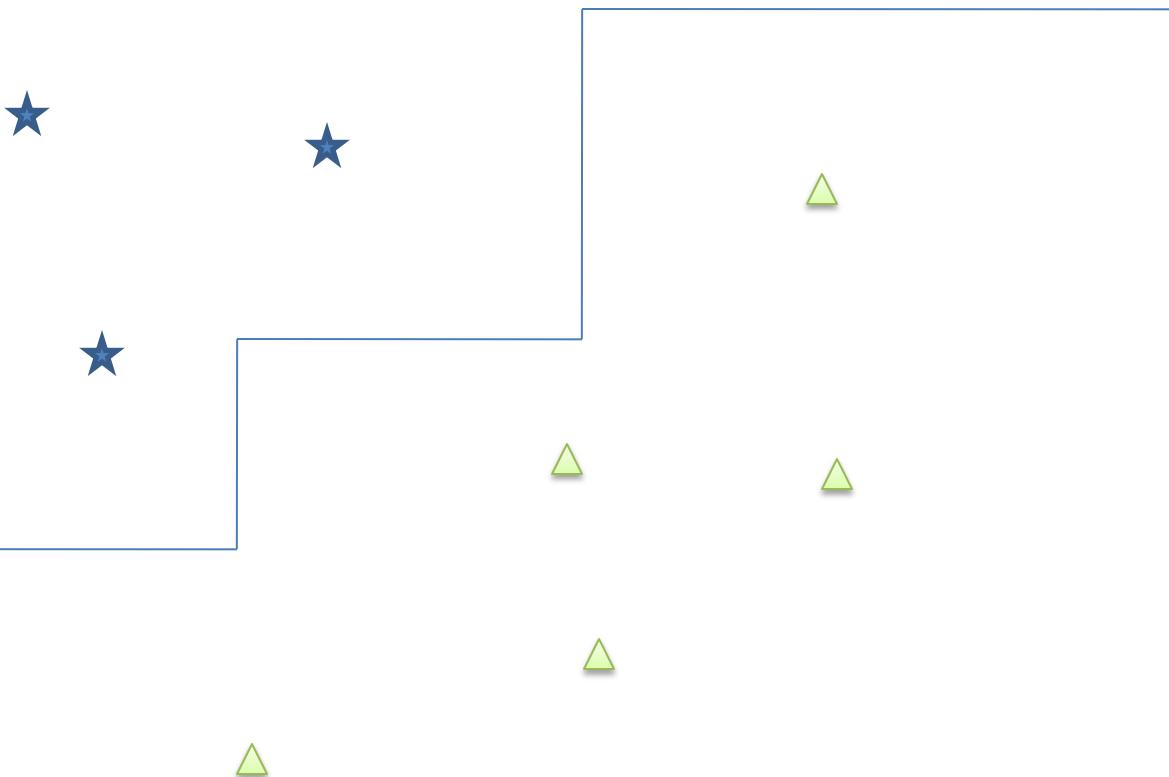
Rule-Based Classifiers



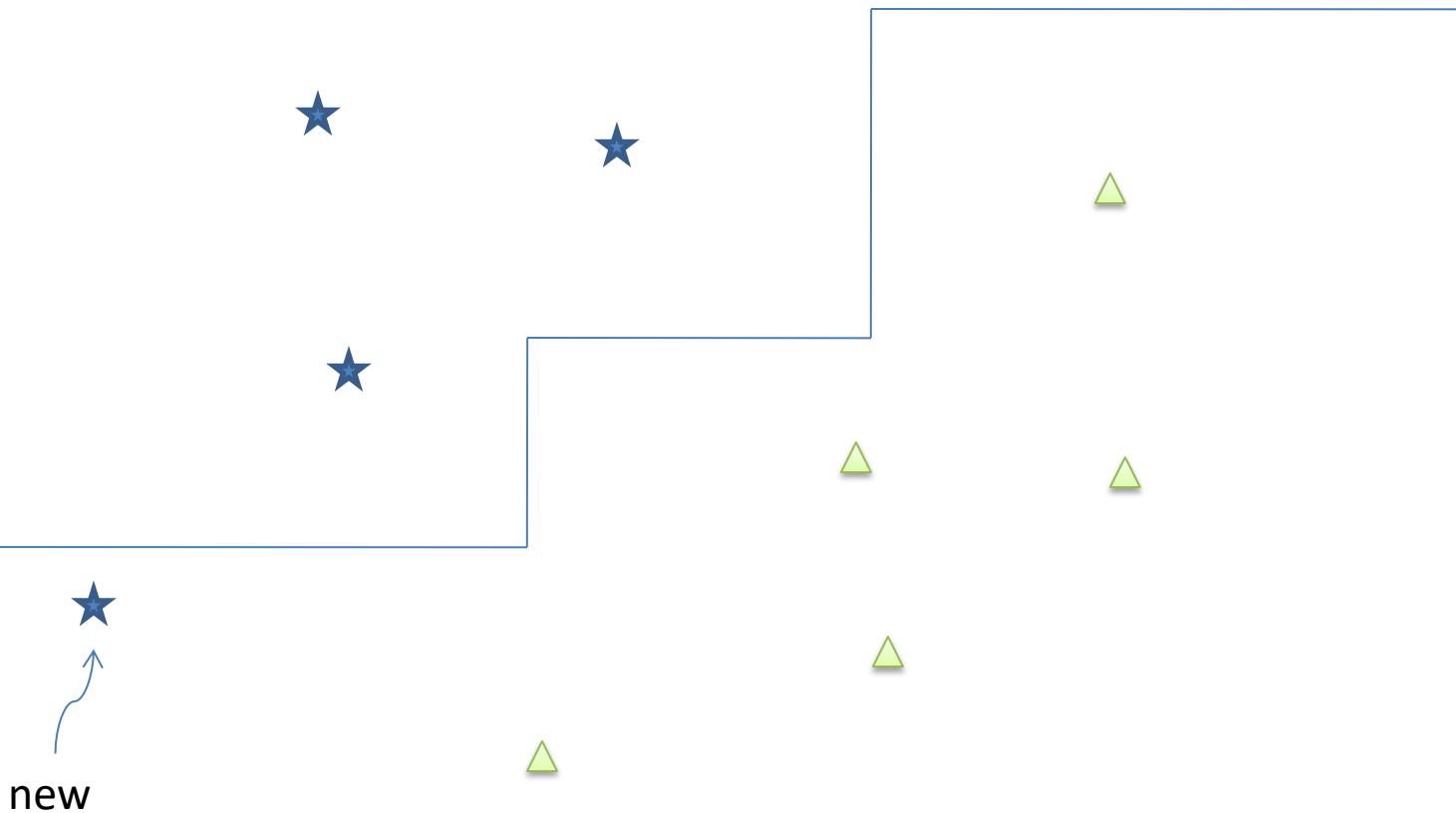
Rule-Based Classifiers

- Try a more complex rule to eliminate the blue point that is being classified as green.
- Example:
 - If `sepal_length > 3 and sepal_width < 5 OR sepal_length > 2 and sepal_width < 4` then `class=setosa` else `class=versicolor`
- Now we get 2 more lines in 2D.

Rule-Based Classifiers



Rule-Based Classifiers



Rule-Based Classifiers

- So, I'll need an infinite number of rules!
- Is there a machine learning algorithm that can handle this data?

Word-List Based Classification

Word List (Gazetteer)

- A very simple classification technique that is used very often and works well if done correctly.

Classification for Text Categorization

Let's say you want to classify these. Make a list of key words. What sort of words?

The United Nations Security Council will record its first clear condemnation of Syria for its continuing crackdown on protests, as the army continues its advance into Hama.

Warwickshire's Clarke equalled the first-class record of seven catches for an outfielder in an innings but Lancashire took control on day three.

Text Categorization using Lists of Words

These lists of words are also called gazetteers

Politics

United
Nations
Security
Record
Condemnation
Syria
Crackdown
Protests
Army
Hama

Sports

Manchester
United
Warwickshire
Clarke
First-class
Record
Catches
Outfielder
Innings
Lancashire

Word List (Gazetteer)

- I can write rules like “If any word in the document matches a word in the word list for a category, it belongs to that category”.
- This type of word list classification is a type of rule-based classifier.
- Example:
 - If word1 exists in doc OR word2 exists in doc OR word3 exists in doc then class=C1 else class=C2

Word List (Gazetteer)

- You can greatly improve the quality of this by counting.
- I count occurrences of words from the word list for each category in the document.
- If $\text{count}(\text{category1}) > \text{count}(\text{category2})$ in document, then category 1 else category2

Word List Topic Classification Example

United Nations

Manchester United

Word List Topic Classification Example

United Nations

count(Politics) = 2
count(Sports) = 1

Manchester United

count(Politics) = 1
count(Sports) = 2

Word List Topic Classification Example

Politics

United Nations

$\text{count}(\text{Politics}) = 2$

>

$\text{count}(\text{Sports}) = 1$

Sports

Manchester United

$\text{count}(\text{Politics}) = 1$

<

$\text{count}(\text{Sports}) = 2$

**We have successfully used a rule-based
algorithm to tell us which document is about
Politics and which is about Sports !!!**

Word List (Gazetteer)

- A very simple classification technique that is used very often but never written about in textbooks.
- I count occurrences of words from the word list for each category. If $\text{count}(\text{category1}) > \text{count}(\text{category2})$ in document, then category 1 else category2
- Word lists are an improvement over decision tree classifiers because they involve counting.

Word List (a.k.a. Gazetteer)

- The problem with words lists is, all the words (features) have equal weightage.
- But hey, some words (features) might be better indicators of the topic than others.
- So, what if we added a weight to the features ... like what you see on the next slide ...

Adding Weights to the Features

Let's improve the gazetteer with weights

Politics

UN	1.0
Adopts	0.1
Condemnation	0.2
Syria	0.3
Crackdown	0.8
Protests	1.0
Army	0.8
Hama	1.0

Sports

Warwickshire	0.3
Clarke	0.1
First-class	0.6
Record	0.3
Catches	0.6
Outfielder	1.0
Innings	0.9
Lancashire	0.5

Weighted Gazetteers

- But that's exactly what machine learning models are – weighted gazetteers.

Weighted Gazetteers

- But, counts multiplied by weights represent lines!
- Eg: $mx + ny$ is a 2 dimensional line
- Eg: $ax + by + cz$ is a 3 dimensional line
- ...

Weighted Gazetteers

- But that's exactly what machine learning models are – weighted gazetteers.
- And the linear classifier had been invented!

Linear Classifiers

- Artificial Neural Networks (Perceptrons)
- Naïve Bayesian Classifier
- Maximum Entropy Classifiers
- Support Vector Machines

Perceptron

Perceptron

For binary (two class classification – 0 or 1)

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

$f(x)$ returns the class 1 if $w \cdot x + b > 0$.

$f(x)$ returns the class 0 if $w \cdot x + b \leq 0$.

Courtesy Wikipedia

Perceptron Training

Steps

1. Initialize the weights and the threshold.
Weights may be initialized to 0 or to a small random value.

Courtesy Wikipedia

Perceptron Training

Steps

2. For each example in the training set, perform the following steps:

Calculate the actual output:

$$y_j(t) = f[\mathbf{w}(t) \cdot \mathbf{x}_j] = f[w_0(t) + w_1(t)x_{j,1} + w_2(t)x_{j,2} + \cdots + w_n(t)x_{j,n}]$$

Update the weights:

$$w_i(t+1) = w_i(t) + \alpha(d_j - y_j(t))x_{j,i}$$

Courtesy Wikipedia

Perceptron Training

Steps

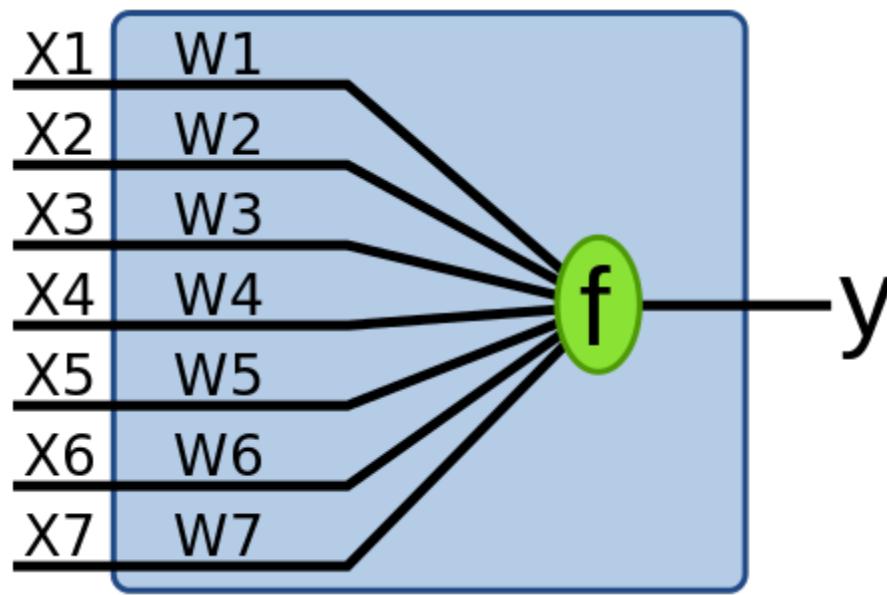
3. Repeat until the iteration error is less than a user-specified error threshold, or a predetermined number of iterations have been completed.

Courtesy Wikipedia

Now that you've learnt the weights

For binary (two class classification – 0 or 1)

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

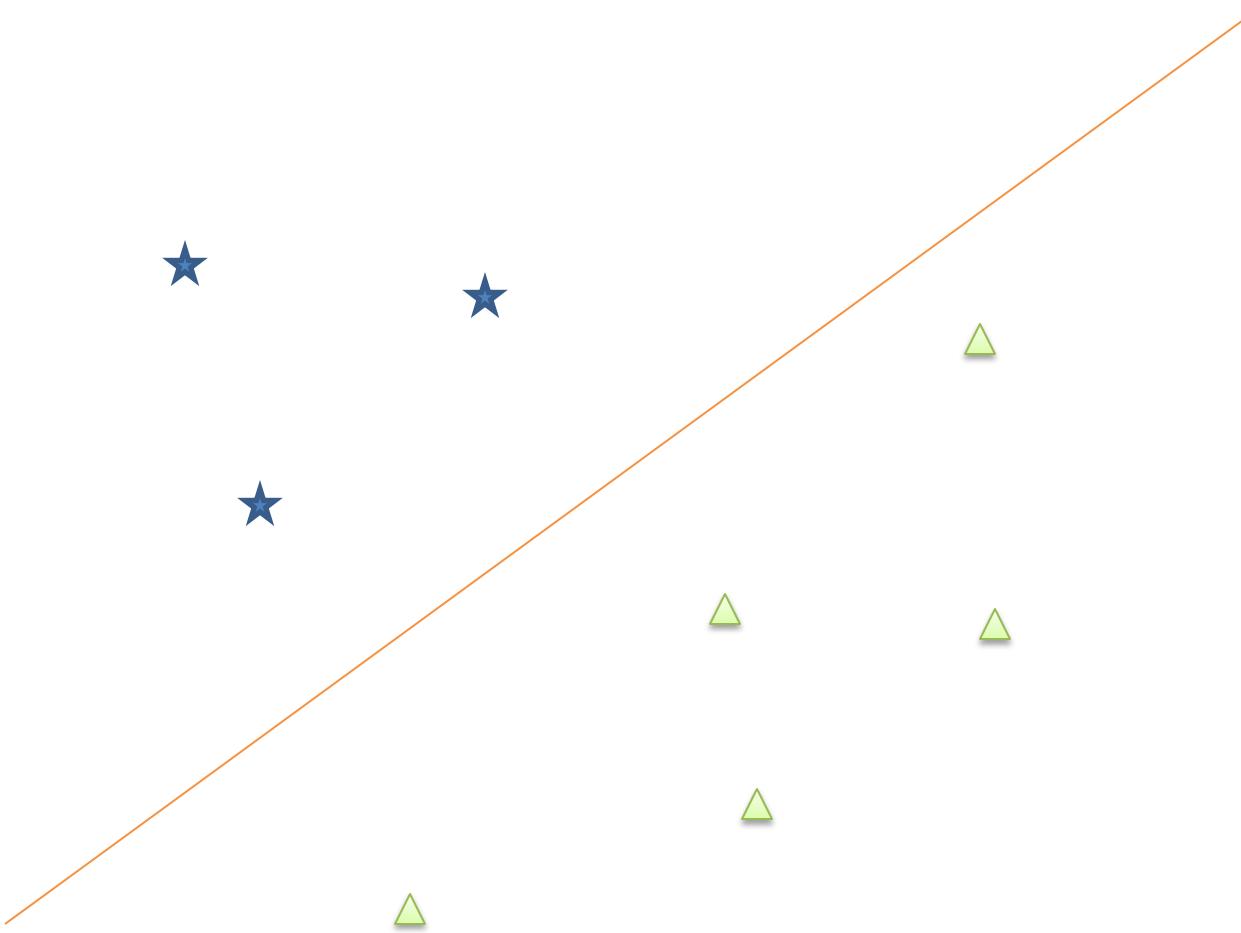


Courtesy Wikipedia

Perceptron

- Can separate two linearly separable data sets in hyperspace (a Euclidean space of dimension n).
- Now we get 1 sloping line in 2D.

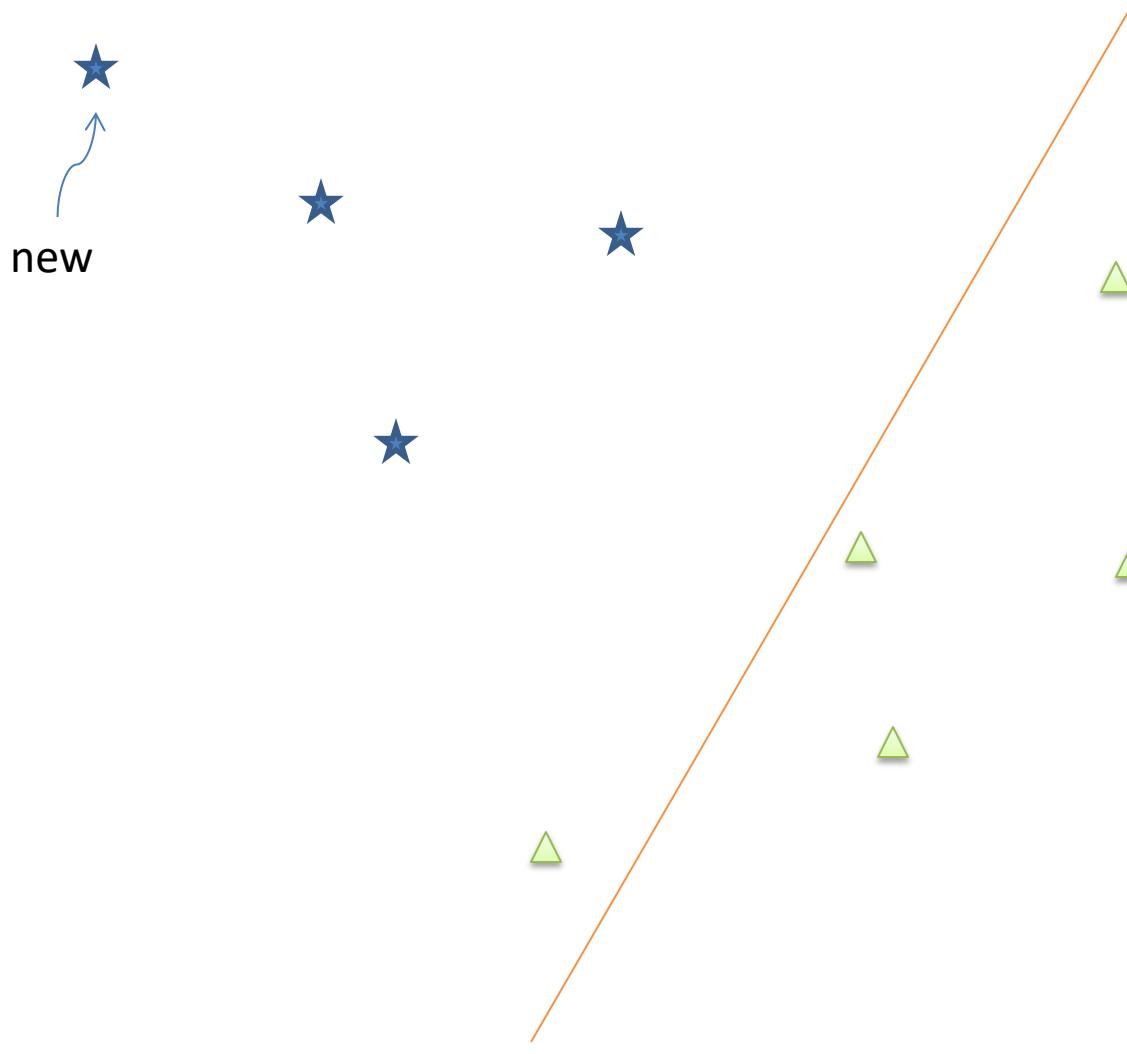
Perceptrons



The Problem with Perceptrons

- Dependent on order of data
- Jitter (the separating plane can move around a lot as you train it)
- Sensitive to outliers appearing at the end of training

Perceptrons



The Jitter Problem with Perceptrons

Can we solve this problem using probabilities?

What if the Weights were Probabilities?

The gazetteer with weights

Politics

UN	1.0
Adopts	0.1
Condemnation	0.2
Syria	0.3
Crackdown	0.8
Protests	1.0
Army	0.8
Hama	1.0

Sports

Warwickshire	0.3
Clarke	0.1
First-class	0.6
Record	0.3
Catches	0.6
Outfielder	1.0
Innings	0.9
Lancashire	0.5

What if the Weights were Probabilities?

So ... for these categories we'd have

Politics		Sports	
UN	1.0	Warwickshire	0.3
Adopts	0.1	Clarke	0.1
Condemnation	0.2	First-class	0.6
Syria	0.3	Record	0.3
Crackdown	0.8	Catches	0.6
Protests	1.0	Outfielder	1.0
Army	0.8	Innings	0.9
Hama	1.0	Lancashire	0.5

$P(\text{Politics} \mid \text{Word})$

$P(\text{Sports} \mid \text{Word})$

What is a probability?

- A number between 0 and 1
- The sum of the probabilities on all outcomes is 1

Heads



Tails



- $P(\text{heads}) = 0.5$
- $P(\text{tails}) = 0.5$

Concept 1 - What is a joint probability?

- Chance of something happening and something else happening.
- Chance of something happening at the same time as something else.
- Chance of seeing the word Barack and the word Obama in a document.
- Chance of the document being about Sports and having the word ‘innings’ in it.

Concept 2 - What is a conditional probability?

- Chance of something happening when you know that something else has happened.
- Chance of something happening given something else happens.
- Chance of seeing the word Barack in a document when you see the word Obama.
- Chance of the document being about Sports when you see the word ‘innings’ in it.

Independence

Two events are independent if the chance of one event happening given the other is the same as the chance of the event happening irrespective of the other event.



To put it simply:
If $P(E | F) = P(E)$

Independence

Combo	E	F	$P(E)$	$P(E F) = P(EF)/P(F)$
1	• T 	• T 	• 0.5	• 0.5
2	• T 	• H 	• 0.5	• 0.5
3	• H 	• T 	• 0.5	• 0.5
4	• H 	• H 	• 0.5	• 0.5

E and F are independent if $P(E|F) = P(E)$

The Independence Puzzle

Two events are independent if the chance of one event happening given the other is the same as the chance of the event happening irrespective of the other event.



If E and F are
Independent, show:

$$P(EF) = P(E)*P(F)$$

The Independence Puzzle

Two events are independent if the chance of one event happening given the other is the same as the chance of the event happening irrespective of the other event.



If E and F are
Independent,

$$P(E|F) = P(E)$$

But Concept 2:

$$P(E|F) = P(EF)/P(F)$$

Independence

Combo	E	F	$P(EF)$	$P(E) * P(F)$
1	• T 	• T 	• 0.3	• $0.5 * 0.6$
2	• T 	• H 	• 0.2	• $0.5 * 0.4$
3	• H 	• T 	• 0.3	• $0.5 * 0.6$
4	• H 	• H 	• 0.2	• $0.5 * 0.4$
Total			• 1.0	• 1.0

Concept 3 - What is Bayesian Inversion?

Turning

$P(F|E)$ into $P(E|F)$.



Proof?

The equation to do this is:

$$P(E|F) = P(F|E) * P(E) / P(F) \text{ [Bayesian Inversion]}$$

Proof : Bayesian Inversion

Proof

$$P(FE) = P(F|E) * P(E)$$

$$P(EF) = P(E|F) * P(F)$$

Set $P(EF)=P(FE)$

and you get the equation:

$$P(E|F) = P(F|E) * P(E) / P(F) \text{ [Bayesian Inversion]}$$

Bayesian Inversion Table

Combo	E	F	P(E)	P(F)	P(E F)	$P(F E) = P(E F)*P(F)/P(E)$
1	• T 	• T 	• 0.5	• 0.6	• 0.5	• $0.5*0.6/0.5 = 0.6$
2	• T 	• H 	• 0.5	• 0.4	• 0.5	• $0.5*0.4/0.5 = 0.4$
3	• H 	• T 	• 0.5	• 0.6	• 0.5	• $0.5*0.6/0.5 = 0.6$
4	• H 	• H 	• 0.5	• 0.4	• 0.5	• $0.5*0.4/0.5 = 0.4$

Why are $P(F)$ and $P(F|E)$ identical?

Naïve Bayesian Classifier

Do You Remember This?

United Nations

$P(\text{Sports} | \text{United Nations})$

$$= P(\text{United} | S) * P(\text{Nations} | \text{Sports}) * P(\text{Sports})$$

$$= (1/5) * (0) * (5/12) = 0$$

$P(\text{Politics} | \text{United Nations})$

$$= P(\text{United} | P) * P(\text{Nations} | \text{Politics}) * P(\text{Politics})$$

$$= (2/7) * (1/7) * (7/12) = 1/(7*6)$$

And This?

Manchester United

$P(\text{Sports} | \text{Manchester United})$

$$= P(\text{Manchester} | S) * P(\text{United} | S) * P(\text{Sports})$$

$$= (2/5) * (1/5) * (5/12) = 2/(5 * 12)$$

$P(\text{Politics} | \text{Manchester United})$

$$= P(\text{Manchester} | P) * P(\text{United} | P) * P(\text{Politics})$$

$$= (0) * (2/7) * (7/12) = 0$$

Naïve Bayesian Classifier

You have the equation:

$$P(C|F) = P(F|C) * P(C) / P(F) \text{ [Bayesian Inversion]}$$

This is like saying:

$$P(\text{Class} | \text{Features}) =$$

$$P(\text{Features} | \text{Class}) * P(\text{Class}) / P(\text{Features})$$

Naive Bayesian Classifier

During Classification:

You try to find the most probable class for a document.

You calculate

$P(\text{Politics} \mid \text{document})$ and $P(\text{Sports} \mid \text{document})$

then pick **Politics** if

$P(\text{Politics} \mid \text{document}) > P(\text{Sports} \mid \text{document})$

else pick **Sports**

Naïve Bayesian Classifier

Classification is about comparing:
 $P(\text{Sports} \mid \text{Features})$ and $P(\text{Politics} \mid \text{Features})$

But we have:

$$P(\text{Class} \mid \text{Features}) = \\ P(\text{Features} \mid \text{Class}) * P(\text{Class}) / P(\text{Features})$$

So, for **Sports**

$$P(\text{Sports} \mid \text{Features}) = \\ P(\text{Features} \mid \text{Sports}) * P(\text{Sports}) / P(\text{Features})$$

Naïve Bayesian Classifier

Classification is about comparing:
 $P(\text{Sports} \mid \text{Features})$ and $P(\text{Politics} \mid \text{Features})$

So, for **Sports**

$$P(\text{Sports} \mid \text{Features}) =$$

$$P(\text{Features} \mid \text{Sports}) * P(\text{Sports}) / P(\text{Features})$$

So, for **Politics**

$$P(\text{Politics} \mid \text{Features}) =$$

$$P(\text{Features} \mid \text{Politics}) * P(\text{Politics}) / P(\text{Features})$$

Naïve Bayesian Classifier

Classification is about comparing:

$P(\text{Sports} | \text{Features})$ and $P(\text{Politics} | \text{Features})$

So, to compare **Sports** and **Politics**, we don't need the denominators:

$P(\text{Sports} | \text{Features}) =$

$P(\text{Features} | \text{Sports}) * P(\text{Sports}) / \cancel{P(\text{Features})}$

$P(\text{Politics} | \text{Features}) =$

$P(\text{Features} | \text{Politics}) * P(\text{Politics}) / \cancel{P(\text{Features})}$

Naive Bayesian Classifier

During Classification of a 1-word document “UN”:

P(Politics | “UN”) is computed from P(“UN” | Politics)

Using the formula $P(E|F) = P(F|E) * P(E) / P(F)$

Naive Bayesian Classifier

So, it's

$$P(\text{Politics} \mid \text{"UN"}) = P(\text{"UN"} \mid \text{Politics}) * P(\text{Politics}) \\ / P(\text{"UN"})$$

vs

$$P(\text{Sports} \mid \text{"UN"}) = P(\text{"UN"} \mid \text{Sports}) * P(\text{Sports}) \\ / P(\text{"UN"})$$

The denominator $P(\text{"UN"})$ is inconsequential in the comparison because it's the same in both equations being compared, so we can omit it.

Naive Bayesian Classifier

Take a document “UN Security Council”

So

$$P(\text{Politics} \mid \text{"UN"}, \text{"Security"}, \text{"Council"}) =$$

$$\frac{P(\text{"UN"}, \text{"Security"}, \text{"Council"} \mid \text{Politics}) * P(\text{Politics})}{P(\text{"UN"}, \text{"Security"}, \text{"Council"})}$$

The denominator $P(\text{"UN"}, \text{"Security"}, \text{"Council"})$ is inconsequential because it's the same in all things being compared.

Naive Bayesian Classifier

Take a document “UN Security Council”

Omitting the denominator, you only have to compare:

$$P(\text{Politics} \mid \text{"UN"}, \text{"Security"}, \text{"Council"}) =$$

$$P(\text{"UN"}, \text{"Security"}, \text{"Council"} \mid \text{Politics}) * P(\text{Politics})$$

with

$$P(\text{Sports} \mid \text{"UN"}, \text{"Security"}, \text{"Council"}) =$$

$$P(\text{"UN"}, \text{"Security"}, \text{"Council"} \mid \text{Sports}) * P(\text{Sports})$$

Naïve Bayesian Classifier

So, to compare **Sports** and **Politics**, we don't need the denominators:

$$P(\text{Sports} \mid \text{Features}) =$$

$$P(\text{Features} \mid \text{Sports}) * P(\text{Sports}) / \cancel{P(\text{Features})}$$

$$P(\text{Politics} \mid \text{Features}) =$$

$$P(\text{Features} \mid \text{Politics}) * P(\text{Politics}) / \cancel{P(\text{Features})}$$

Naïve Assumption

The reason for the word “naïve” in the name:
The features are all assumed independent.

$$P(\text{Features} | \text{Sports}) =$$

$$P(\text{Feature1} | \text{Sports}) * P(\text{Feature2} | \text{Sports}) * \dots$$

and

$$P(\text{Features} | \text{Politics}) =$$

$$P(\text{Feature1} | \text{Politics}) * P(\text{Feature2} | \text{Politics}) * \dots$$

Naïve Bayesian Classifier

Since:

$$P(\text{Features} | \text{Sports}) = P(\text{Feature1} | \text{Sports}) * P(\text{Feature2} | \text{Sports}) * \dots$$

$$P(\text{Features} | \text{Politics}) = P(\text{Feature1} | \text{Politics}) * P(\text{Feature2} | \text{Politics}) * \dots$$

We get:

$$P(\text{Sports} | \text{Features}) =$$

$$P(\text{Feature1} | \text{Sports}) * P(\text{Feature2} | \text{Sports}) * P(\text{Feature3} | \text{Sports}) * \\ \dots * P(\text{Sports}) / P(\text{Features})$$

$$P(\text{Politics} | \text{Features}) =$$

$$P(\text{Feature1} | \text{Politics}) * P(\text{Feature2} | \text{Politics}) * P(\text{Feature3} | \text{Politics}) * \\ \dots * P(\text{Politics}) / P(\text{Features})$$

That's How We Got This

United Nations

$P(\text{Sports} | \text{Features}) =$

$P(\text{Feature1} | \text{Sports}) * P(\text{Feature2} | \text{Sports}) * P(\text{Feature3} | \text{Sports}) * \dots * P(\text{Sports}) / P(\text{Features})$

$P(\text{Politics} | \text{Features}) =$

$P(\text{Feature1} | \text{Politics}) * P(\text{Feature2} | \text{Politics}) * P(\text{Feature3} | \text{Politics}) * \dots * P(\text{Politics}) / P(\text{Features})$

$P(\text{Sports} | \text{United Nations})$

$= P(\text{United} | \text{S}) * P(\text{Nations} | \text{Sports}) * P(\text{Sports})$

$= (1/5) * (0) * (5/12) = 0$

$P(\text{Politics} | \text{United Nations})$

$= P(\text{United} | \text{P}) * P(\text{Nations} | \text{Politics}) * P(\text{Politics})$

$= (2/7) * (1/7) * (7/12) = 1/(7 * 6)$

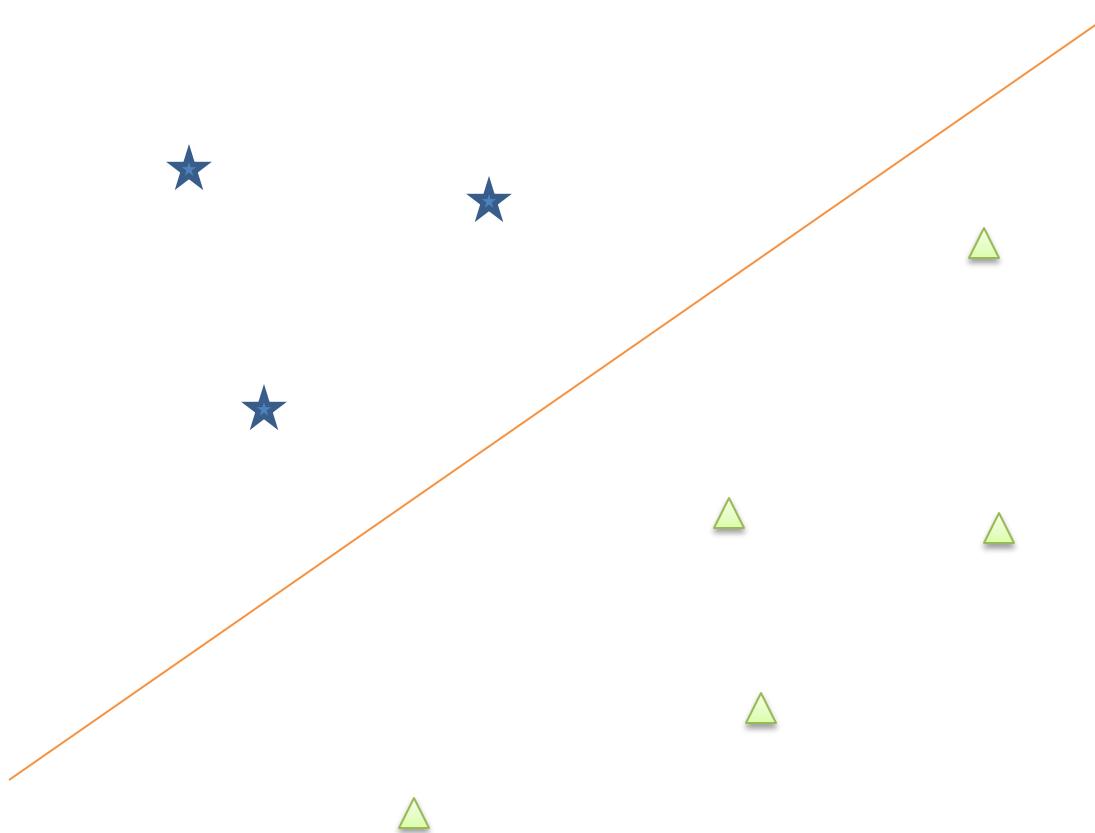
Naïve Bayesian Weights

Politics

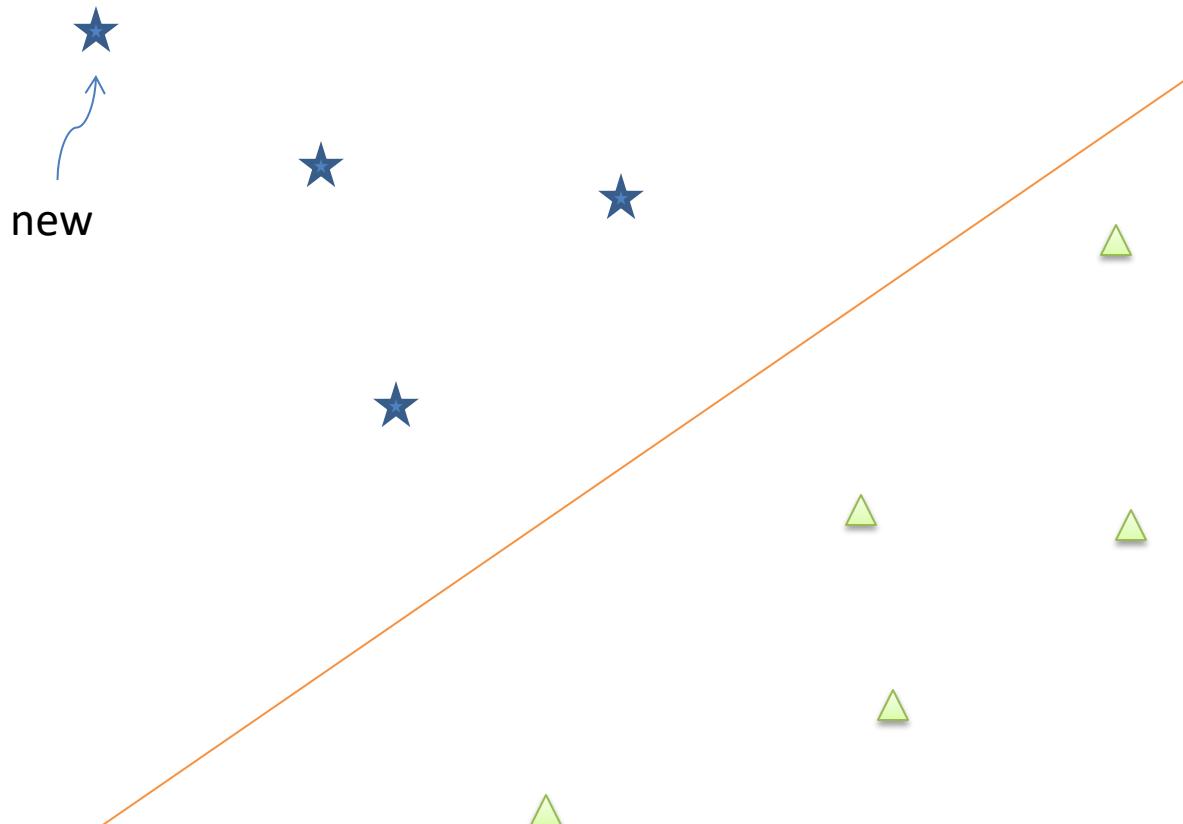
UN	0.7	$P(\text{UN} \mid \text{Politics})$
Adopts	0.3	$P(\text{Adopts} \mid \text{Politics})$
Condemnation	0.4	$P(\text{Condemnation} \mid \text{Politics})$
Syria	0.1	$P(\text{Syria} \mid \text{Politics})$
Crackdown	0.6	$P(\text{Crackdown} \mid \text{Politics})$
Protests	0.9	$P(\text{Protests} \mid \text{Politics})$
Army	0.6	$P(\text{Army} \mid \text{Politics})$
Hama	0.7	$P(\text{Hama} \mid \text{Politics})$

Combined by multiplication

Naïve Bayesian Classifiers



Naïve Bayesian Classifiers



Order independent!

Two Types of Naïve Bayesian Classifier

Politics

UN	0.7	$P(\text{UN} \mid \text{Politics})$
Adopts	0.3	$P(\text{Adopts} \mid \text{Politics})$
Condemnation	0.4	$P(\text{Condemnation} \mid \text{Politics})$
Syria	0.1	$P(\text{Syria} \mid \text{Politics})$
Crackdown	0.6	$P(\text{Crackdown} \mid \text{Politics})$
Protests	0.9	$P(\text{Protests} \mid \text{Politics})$
Army	0.6	$P(\text{Army} \mid \text{Politics})$
Hama	0.7	$P(\text{Hama} \mid \text{Politics})$

Two ways of estimating $P(\text{Feature} \mid \text{Class})$

Estimating Probabilities

How do you calculate $P(\text{"UN"} | \text{Politics})$?

There are two ways to do that

$P(\text{"UN"} | \text{Politics}) = C(\text{"UN in politics"}) / C(\text{"All words in politics"})$ [Multinomial]

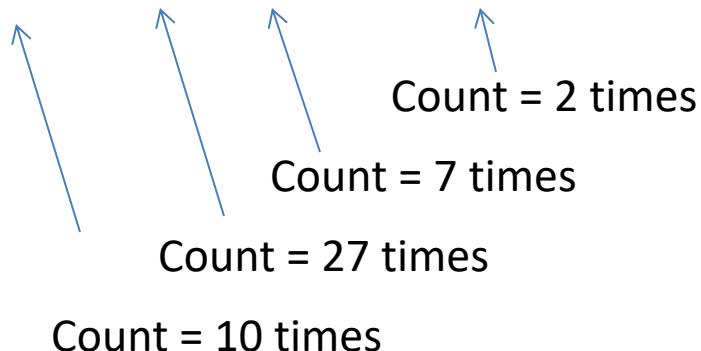
$P(\text{"UN"} | \text{Politics}) = C(\text{"documents on politics in which UN appears"}) / C(\text{"All documents on politics"})$ [Multivariate Bernoulli]

Estimating Probabilities

How do you calculate $P(\text{"UN"} | \text{Politics})$?

Multinomial Model of a Document

This is my document

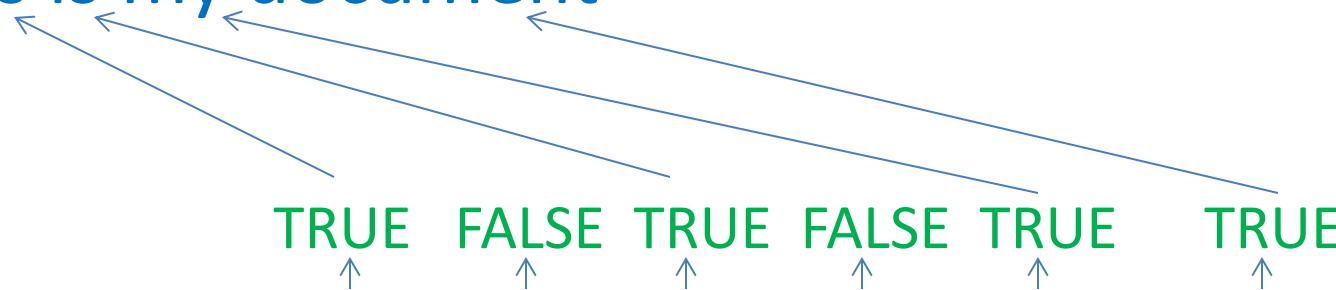


Estimating Probabilities

How do you calculate $P(\text{"UN"} | \text{Politics})$?

Multivariate Bernoulli Model of a Document

This is my document

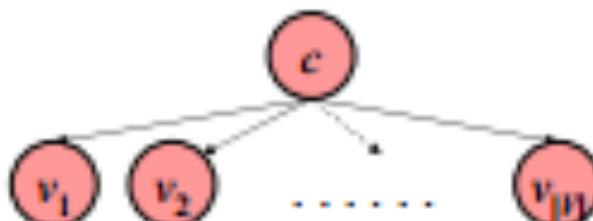
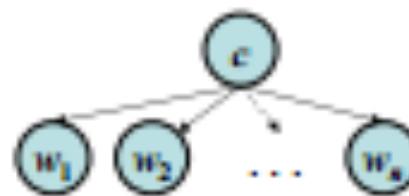


Vocabulary: {This, That, Is, Was, My, Document}

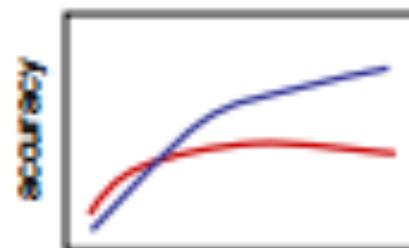
Estimating Probabilities

Two NB Formulations

- Two NB models for text categorization
 - The class-conditional unigram model, a.k.a. multinomial model
 - One node per word in the document
 - Driven by words which are present
 - Multiple occurrences, multiple evidence
 - Better overall – plus, know how to smooth
 - The binomial (binary) model
 - One node for each word in the vocabulary



- Incorporates explicit negative correlations
- Know how to do feature selection (e.g. keep words with high mutual information with the class variable)



Example of Estimating Probabilities

United Nations

United States

Politics

Manchester United

Barca

Sports

Example of Multinomial Probabilities

United Nations

United States

Politics

Manchester United

Barca

Sports

$$P(\text{United}/\text{Politics}) = 1/2$$

$$P(\text{Nations}/\text{Politics}) = 1/4$$

$$P(\text{States}/\text{Politics}) = 1/4$$

$$P(\text{Politics}) = 4/7$$

$$P(\text{Manchester}/\text{Sports}) = 1/3$$

$$P(\text{United}/\text{Sports}) = 1/3$$

$$P(\text{Barca}/\text{Sports}) = 1/3$$

$$P(\text{Sports}) = 3/7$$

Example of Multinomial Probabilities

United

Politics?

Sports?

$$P(\text{United}/\text{Politics}) = 1/2$$

$$P(\text{Nations}/\text{Politics}) = 1/4$$

$$P(\text{States}/\text{Politics}) = 1/4$$

$$P(\text{Politics}) = 4/7$$

$$P(\text{Manchester}/\text{Sports}) = 1/3$$

$$P(\text{United}/\text{Sports}) = 1/3$$

$$P(\text{Barca}/\text{Sports}) = 1/3$$

$$P(\text{Sports}) = 3/7$$

$$\begin{aligned} P(\text{Politics}|\text{United}) &= P(\text{United}|\text{Politics}) * P(\text{Politics}) \\ &= (1/2) * (4/7) = 2/7 \end{aligned}$$

$$\begin{aligned} P(\text{Sports}|\text{United}) &= P(\text{United}|\text{Sports}) * P(\text{Sports}) \\ &= (1/3) * (3/7) = 1/7 \end{aligned}$$

Example of Bernoulli Probabilities

United Nations

United States

Politics

Manchester United

Barca

Sports

$$P(\text{United}/\text{Politics}) = 2/2$$

$$P(\text{Nations}/\text{Politics}) = 1/2$$

$$P(\text{States}/\text{Politics}) = 1/2$$

$$P(\text{Politics}) = 2/4$$

$$P(\text{Manchester}/\text{Sports}) = 1/2$$

$$P(\text{United}/\text{Sports}) = 1/2$$

$$P(\text{Barca}/\text{Sports}) = 1/2$$

$$P(\text{Sports}) = 2/4$$

Example of Bernoulli Probabilities

United

Politics? Sports?

United !Nations !States !Barca !Manchester

$$P(\text{United}/\text{Politics}) = 2/2$$

$$P(\text{Nations}/\text{Politics}) = 1/2$$

$$P(\text{States}/\text{Politics}) = 1/2$$

$$P(\text{Politics}) = 2/4$$

$$P(\text{Manchester}/\text{Sports}) = 1/2$$

$$P(\text{United}/\text{Sports}) = 1/2$$

$$P(\text{Barca}/\text{Sports}) = 1/2$$

$$P(\text{Sports}) = 2/4$$

$$P(\text{Sports}|\text{United}) =$$

$$P(U|S)*P(!M|S)*P(!B|S)*(!N/S)*(!S/S)*P(S)$$

$$= (1/2)*(1/2)*(1/2)*(1)*(1)*(2/4) = 1/32$$

Example of Bernoulli Probabilities

United

Politics? Sports?

United !Nations !States !Barca !Manchester

$$P(\text{United}/\text{Politics}) = 2/2$$

$$P(\text{Nations}/\text{Politics}) = 1/2$$

$$P(\text{States}/\text{Politics}) = 1/2$$

$$P(\text{Politics}) = 2/4$$

$$P(\text{Manchester}/\text{Sports}) = 1/2$$

$$P(\text{United}/\text{Sports}) = 1/2$$

$$P(\text{Barca}/\text{Sports}) = 1/2$$

$$P(\text{Sports}) = 2/4$$

$$P(\text{Politics}|\text{United}) =$$

$$P(U|P) * P(!N|P) * P(!S|P) * (!B|P) * (!M|P) / P(P)$$

$$= (2/2) * (1/2) * (1/2) * (1) * (1) * (2/4) = 2/32$$

Example of Estimating Probabilities

The United Nations

The United States and

Manchester United

Manchester and Barca

Politics

Sports

Example of Multinomial Probabilities

The United Nations

The United States and

Manchester United

Manchester and Barca

Politics

Sports

$$P(\text{and}/\text{Politics}) = 1/7$$

$$P(\text{The}/\text{Politics}) = 2/7$$

$$P(\text{United}/\text{Politics}) = 2/7$$

$$P(\text{Nations}/\text{Politics}) = 1/7$$

$$P(\text{States}/\text{Politics}) = 1/7$$

$$P(\text{Politics}) = 7/12$$

$$P(\text{and}/\text{Sports}) = 1/5$$

$$P(\text{Manchester}/\text{Sports}) = 2/5$$

$$P(\text{United}/\text{Sports}) = 1/5$$

$$P(\text{Barca}/\text{Sports}) = 1/5$$

$$P(\text{Sports}) = 5/12$$

Example of Multinomial Probabilities

Nations and States

$$P(\text{and}/\text{Politics}) = 1/7$$

$$P(\text{The}/\text{Politics}) = 2/7$$

$$P(\text{United}/\text{Politics}) = 2/7$$

$$P(\text{Nations}/\text{Politics}) = 1/7$$

$$P(\text{States}/\text{Politics}) = 1/7$$

$$P(\text{Politics}) = 7/12$$

Politics?

Sports?

$$P(\text{and}/\text{Sports}) = 1/5$$

$$P(\text{Manchester}/\text{Sports}) = 2/5$$

$$P(\text{United}/\text{Sports}) = 1/5$$

$$P(\text{Barca}/\text{Sports}) = 1/5$$

$$P(\text{Sports}) = 5/12$$

$$P(\text{Politics} | \text{Nations, and, States}) =$$

$$P(\text{Nations} | P) * P(\text{and} | P) * P(\text{States} | \text{Politics}) * P(\text{Politics})$$

$$= (1/7) * (1/7) * (1/7) * (7/12) = 1/(49 * 12)$$

Example of Multinomial Probabilities

Nations and States

$$P(\text{and}/\text{Politics}) = 1/7$$

$$P(\text{The}/\text{Politics}) = 2/7$$

$$P(\text{United}/\text{Politics}) = 2/7$$

$$P(\text{Nations}/\text{Politics}) = 1/7$$

$$P(\text{States}/\text{Politics}) = 1/7$$

$$P(\text{Politics}) = 7/12$$

$$P(\text{Sports} | \text{Nations, and, States}) =$$

$$P(\text{Nations} | S) * P(\text{and} | S) * P(\text{States} | \text{Sports}) * P(\text{Sports})$$

$$= (0) * (1/5) * (0) * (5/12) = 0$$

Politics?

Sports?

$$P(\text{and}/\text{Sports}) = 1/5$$

$$P(\text{Manchester}/\text{Sports}) = 2/5$$

$$P(\text{United}/\text{Sports}) = 1/5$$

$$P(\text{Barca}/\text{Sports}) = 1/5$$

$$P(\text{Sports}) = 5/12$$

Example of Multinomial Probabilities

Training

The United Nations

The United States and

Manchester United

Manchester and Barca

Politics

Sports

Test

and United

Politics?

Sports?

Example of Multinomial Probabilities

and United

$$P(\text{and}/\text{Politics}) = 1/7$$

$$P(\text{The}/\text{Politics}) = 2/7$$

$$P(\text{United}/\text{Politics}) = 2/7$$

$$P(\text{Nations}/\text{Politics}) = 1/7$$

$$P(\text{States}/\text{Politics}) = 1/7$$

$$P(\text{Politics}) = 7/12$$

$$P(\text{Politics} \mid \text{and}, \text{United}) = ?$$

Politics?

Sports?

$$P(\text{and}/\text{Sports}) = 1/5$$

$$P(\text{Manchester}/\text{Sports}) = 2/5$$

$$P(\text{United}/\text{Sports}) = 1/5$$

$$P(\text{Barca}/\text{Sports}) = 1/5$$

$$P(\text{Sports}) = 5/12$$

Example of Multinomial Probabilities

and United

$$P(\text{and}/\text{Politics}) = 1/7$$

$$P(\text{The}/\text{Politics}) = 2/7$$

$$P(\text{United}/\text{Politics}) = 2/7$$

$$P(\text{Nations}/\text{Politics}) = 1/7$$

$$P(\text{States}/\text{Politics}) = 1/7$$

$$P(\text{Politics}) = 7/12$$

$$P(\text{Politics} | \text{and}, \text{United}) =$$

$$P(\text{and} | P) * P(\text{United} | \text{Politics}) * P(\text{Politics})$$

$$= ?$$

Politics?

Sports?

$$P(\text{and}/\text{Sports}) = 1/5$$

$$P(\text{Manchester}/\text{Sports}) = 2/5$$

$$P(\text{United}/\text{Sports}) = 1/5$$

$$P(\text{Barca}/\text{Sports}) = 1/5$$

$$P(\text{Sports}) = 5/12$$

Example of Multinomial Probabilities

and United

Politics?

Sports?

$$P(\text{and}/\text{Politics}) = 1/7$$

$$P(\text{The}/\text{Politics}) = 2/7$$

$$P(\text{United}/\text{Politics}) = 2/7$$

$$P(\text{Nations}/\text{Politics}) = 1/7$$

$$P(\text{States}/\text{Politics}) = 1/7$$

$$P(\text{Politics}) = 7/12$$

$$P(\text{and}/\text{Sports}) = 1/5$$

$$P(\text{Manchester}/\text{Sports}) = 2/5$$

$$P(\text{United}/\text{Sports}) = 1/5$$

$$P(\text{Barca}/\text{Sports}) = 1/5$$

$$P(\text{Sports}) = 5/12$$

$$P(\text{Politics} | \text{and}, \text{United}) =$$

$$P(\text{and} | P) * P(\text{United} | \text{Politics}) * P(\text{Politics})$$

$$= (1/7) * (2/7) * (7/12) = 2/(7 * 12)$$

Example of Multinomial Probabilities

and United

Politics?

Sports?

$$P(\text{and}/\text{Politics}) = 1/7$$

$$P(\text{The}/\text{Politics}) = 2/7$$

$$P(\text{United}/\text{Politics}) = 2/7$$

$$P(\text{Nations}/\text{Politics}) = 1/7$$

$$P(\text{States}/\text{Politics}) = 1/7$$

$$P(\text{Politics}) = 7/12$$

$$P(\text{and}/\text{Sports}) = 1/5$$

$$P(\text{Manchester}/\text{Sports}) = 2/5$$

$$P(\text{United}/\text{Sports}) = 1/5$$

$$P(\text{Barca}/\text{Sports}) = 1/5$$

$$P(\text{Sports}) = 5/12$$

$$P(\text{Sports} | \text{and}, \text{United}) =$$

$$P(\text{and} | S) * P(\text{United} | \text{Sports}) * P(\text{Sports})$$

$$= (1/5) * (1/5) * (5/12) = 1/(5 * 12)$$

NB Multinomial

TRAINMULTINOMIALNB(\mathbb{C}, \mathbb{D})

```
1  $V \leftarrow \text{EXTRACTVOCABULARY}(\mathbb{D})$ 
2  $N \leftarrow \text{COUNTDOCS}(\mathbb{D})$ 
3 for each  $c \in \mathbb{C}$ 
4 do  $N_c \leftarrow \text{COUNTDOCSINCLASS}(\mathbb{D}, c)$ 
5  $prior[c] \leftarrow N_c/N$ 
6  $text_c \leftarrow \text{CONCATENATETEXTOFLDOCSINCLASS}(\mathbb{D}, c)$ 
7 for each  $t \in V$ 
8 do  $T_{ct} \leftarrow \text{COUNTTOKENSOFTERM}(text_c, t)$ 
9 for each  $t \in V$ 
10 do  $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$ 
11 return  $V, prior, condprob$ 
```

APPLYMULTINOMIALNB($\mathbb{C}, V, prior, condprob, d$)

```
1  $W \leftarrow \text{EXTRACTTOKENSFROMDOC}(V, d)$ 
2 for each  $c \in \mathbb{C}$ 
3 do  $score[c] \leftarrow \log prior[c]$ 
4 for each  $t \in W$ 
5 do  $score[c] += \log condprob[t][c]$ 
6 return  $\arg \max_{c \in \mathbb{C}} score[c]$ 
```

<http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html>

NB Multivariate Bernoulli

```
TRAINBERNOULLINB( $\mathbb{C}, \mathbb{D}$ )
1  $V \leftarrow \text{EXTRACTVOCABULARY}(\mathbb{D})$ 
2  $N \leftarrow \text{COUNTDOCS}(\mathbb{D})$ 
3 for each  $c \in \mathbb{C}$ 
4 do  $N_c \leftarrow \text{COUNTDOCSINCLASS}(\mathbb{D}, c)$ 
5    $prior[c] \leftarrow N_c/N$ 
6   for each  $t \in V$ 
7     do  $N_{ct} \leftarrow \text{COUNTDOCSINCLASSCONTAININGTERM}(\mathbb{D}, c, t)$ 
8        $condprob[t][c] \leftarrow (N_{ct} + 1)/(N_c + 2)$ 
9 return  $V, prior, condprob$ 
```

```
APPLYBERNOULLINB( $\mathbb{C}, V, prior, condprob, d$ )
1  $V_d \leftarrow \text{EXTRACTTERMSFROMDOC}(V, d)$ 
2 for each  $c \in \mathbb{C}$ 
3 do  $score[c] \leftarrow \log prior[c]$ 
4   for each  $t \in V$ 
5     do if  $t \in V_d$ 
6       then  $score[c] += \log condprob[t][c]$ 
7       else  $score[c] += \log(1 - condprob[t][c])$ 
8 return  $\arg \max_{c \in \mathbb{C}} score[c]$ 
```

► Figure 13.1 NB algorithm (Bernoulli model): Training and testing. The add-one smoothing in Line 8 (top) is in analogy to Equation 119 with $B = 2$.

<http://nlp.stanford.edu/IR-book/html/htmledition/the-bernoulli-model-1.html>

Why Naïve Bayesian

One of the best classifiers known.
Can be learnt incrementally.
Learning is fast.

Problem with Naïve Bayesian Classifiers

- The confidence values returned by the classifier i.e., $P(C| \text{features})$ cannot be trusted.
- The classifier is always overconfident (it always places all its money on one class).
- Given two classes A and B, say the classifier picks A, then usually $P(A| \text{features}) > 0.9$.
- So you cannot use confidence thresholds

The imbalance grows with the number of features

United Nations

$P(\text{Sports} | \text{United Nations})$

$$= P(\text{United} | S) * P(\text{Nations} | \text{Sports}) * P(\text{Sports})$$

$$= (1/5) * (0) * (5/12) = 0$$

$P(\text{Politics} | \text{United Nations})$

$$= P(\text{United} | P) * P(\text{Nations} | \text{Politics}) * P(\text{Politics})$$

$$= (2/7) * (1/7) * (7/12) = 1/(7*6)$$

Maximum Entropy Classifier

Maximum Entropy Weights

Politics

UN	1.0	$P(\text{Politics} \text{UN})$
Adopts	0.1	$P(\text{Politics} \text{Adopts})$
Condemnation	0.2	$P(\text{Politics} \text{Condemnation})$
Syria	0.3	$P(\text{Politics} \text{Syria})$
Crackdown	0.8	$P(\text{Politics} \text{Crackdown})$
Protests	1.0	$P(\text{Politics} \text{Protests})$
Army	0.8	$P(\text{Politics} \text{Army})$
Hama	1.0	$P(\text{Politics} \text{Hama})$

Combined by weighted addition

How to calculate the $P(C|F)$ directly

Linear Combination of Features

- Take a set of features:
 - Words
 - Capitalization
 - Word context
 - Sub-words
 - Sentence length
 - Position in sentence
- Multiply each by weights
- Add up the products

Example from Text Categorization

“She served to win.”

Feature	Politics	Sports	Crime
serve	1	1	1
serve+object(None)	1	1	1
serve+object(Years)	0	0	0

Weights	Politics	Sports	Crime
serve	+1	-0.3	+1
serve+object(None)	-1.3	+2.7	-3.5
serve+object(Years)	+0.7	-4.5	+2.3

Weights are learnt during training of the system!

Example from Text Categorization

“He served for 10 years as Secretary of State.”

Weights	Politics	Sports	Crime
serve	1 * +1	1 * -0.3	1 * +1
serve+object(None)	0 * -1.3	0 * +2.7	0 * -3.5
serve+object(years)	0 * +0.7	0 * -4.5	0 * +2.3
serve+subcategory(for, years)	1 * +1.5	1 * -1.5	1 * +0.3
serve+subcategorization(as)	1 * +2.3	1 * -2	1 * -3.0
Total	4.3	-3.8	-1.7

How do you learn the weights?

Training a Maximum Entropy Classifier

Adjust the weights w till you find one that maximizes the **sum of the log posterior probabilities** over the training data

$$best \ w = \arg \max_w \sum_i \log P(y^{(i)} | x^{(i)})$$

Turns out this function is both differentiable and convex. So, you can use convex optimization algorithms and do hill-climbing over the space of weights.

Regularization

In practice, you subtract a weight from the **log posterior probabilities** to keep the weights from heading off toward infinity !

$$\hat{w} = \operatorname{argmax}_w \sum_i \log P(y^{(i)} | x^{(i)}) - \alpha R(w)$$

$$R(w) = \sum_{j=1}^N w_j^2$$

How do you calculate $P(y|x)$?

$P(y=\text{true}|x) = w * f ???$ (can't do this!!!)

Problem:

$w * f$ not guaranteed to be a true probability

The product $w*f$ ranges from +infinity to -infinity

So how do we convert $w*f$ into a probability?

Converting From Real Numbers to Probabilities

w.f ranges from $-\infty$ to $+\infty$.

Converting From Real Numbers to Probabilities

$e^{-w.f}$ ranges from 0 to +inf.

Converting From Real Numbers to Probabilities

$1 + e^{-w.f}$ ranges from 1 to +inf.

Converting From Real Numbers to Probabilities

$\frac{1}{1 + e^{-w.f}}$ ranges from 0 to 1.

Wow! This could pass off
as a probability!

Logistic function

Converting From Real Numbers to Probabilities

You can say

$$p(y = \text{true} | x) = \frac{1}{1 + e^{-w.f}}$$

If it adds up to 1 for all y.

Converting From Real Numbers to Probabilities

It's easy to make it add up to 1

$$p(y = \text{true} | x) = \frac{1}{1 + e^{-w.f}} = \frac{e^{w.f}}{e^{w.f} + 1} = \frac{e^{w.f}}{Z}$$

You use a normalizer Z.

Why is it called MaxEnt?

- It turns out that a distribution of the form satisfies maximum entropy constraints:

When constraints are estimated in this fashion, it is guaranteed that a unique distribution exists that has maximum entropy. Moreover, it can be shown [Della Pietra *et al.*, 1997] that the distribution is always of the exponential form:

$$\frac{e^{w.f}}{Z}$$

$$P(c|d) = \frac{1}{Z(d)} \exp\left(\sum_i \lambda_i f_i(d, c)\right), \quad (3)$$

where each $f_i(d, c)$ is a feature, λ_i is a parameter to be estimated and $Z(d)$ is simply the normalizing factor to ensure a proper probability:

$$Z(d) = \sum_c \exp\left(\sum_i \lambda_i f_i(d, c)\right). \quad (4)$$

What does maximum entropy mean?

- 45 possible tags and no constraints:

NN	JJ	NNS	VB	NNP	IN	MD	UH	SYM	VBG	POS	PRP	CC	CD	...
$\frac{1}{45}$...													

- When we know the right tag has to be one of the first four:

NN	JJ	NNS	VB	NNP	IN	MD	UH
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0

Why is it called Logistic Regression?

- Because the following function is called the logistic function or logistic distribution:

$$p(y = \text{true} | x) = \frac{1}{1 + e^{-w.f}} = \frac{e^{w.f}}{e^{w.f} + 1} = \frac{e^{w.f}}{Z}$$

Why is it called Logistic Regression?

- Because the following function is called the logistic function or logistic distribution:

$$p(y = \text{true} | x) = \frac{1}{1 + e^{-w.f}} = \frac{e^{w.f}}{e^{w.f} + 1} = \frac{e^{w.f}}{Z}$$

and training a maxent classifier is equivalent to doing **regression** using the logistic function.

What is Regression?

- It's like doing classification ... in that you're using a set of features x to predict a class y .
- Except that, in regression, the class y is a **real number**.
- Another name for regression is 'curve fitting'.

What's regression? Here's an example

No. of vague adjectives	Extra Price Paid for House (above Asking Price)
3	\$1000
2	\$3000
1	\$9000
0	\$14,000

Use of words like **fantastic, cute, charming** indicates
that a house has no solid qualities (**marble, furnished**)!?

Levitt and Dubner 2005

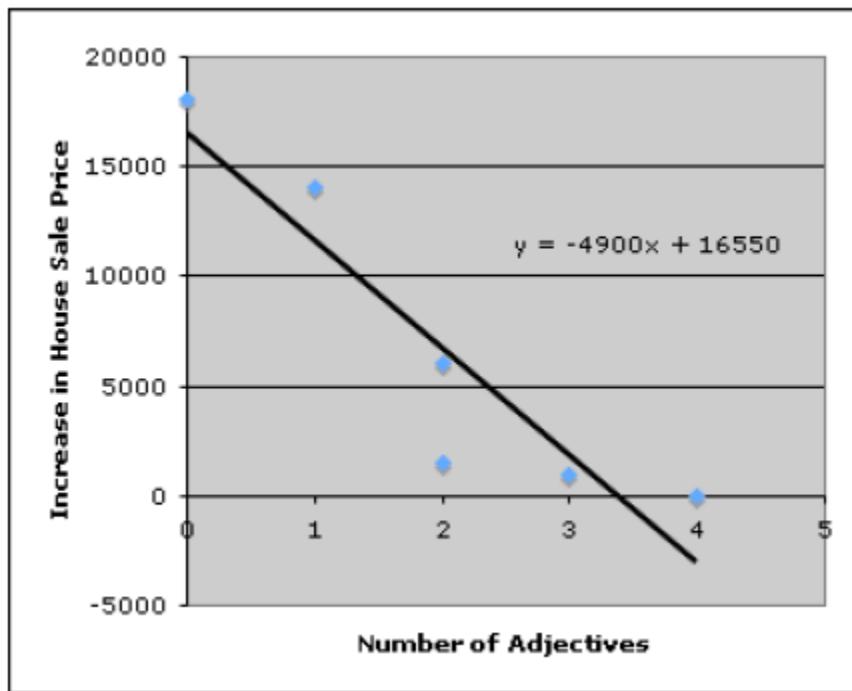
What's regression? Here's an example

x	y
3	\$1000
2	\$3000
1	\$9000
0	\$14,000

Use of words like **fantastic, cute, charming** indicates
that a house has no solid qualities (**marble, furnished**)!?

Levitt and Dubner 2005

Regression



Courtesy of the textbook

Regression is Curve Fitting

- $y = mx + c$
 - also written as $y = w_1f_1 + w_0f_0$
 - f_0 is a dummy feature
 - (linear combination of features using weights)

linear regression:
$$y = \sum_{i=0}^N w_i \times f_i$$

$$y = w.f$$

Solving a Linear Regression

$$y_{predicted}^{(j)} = \sum w_i * f_i^{(j)}$$

- It's like solving for 'm' in 'y = mx + c'
- You can solve for 'w' in terms of 'y' and 'w' over the training data

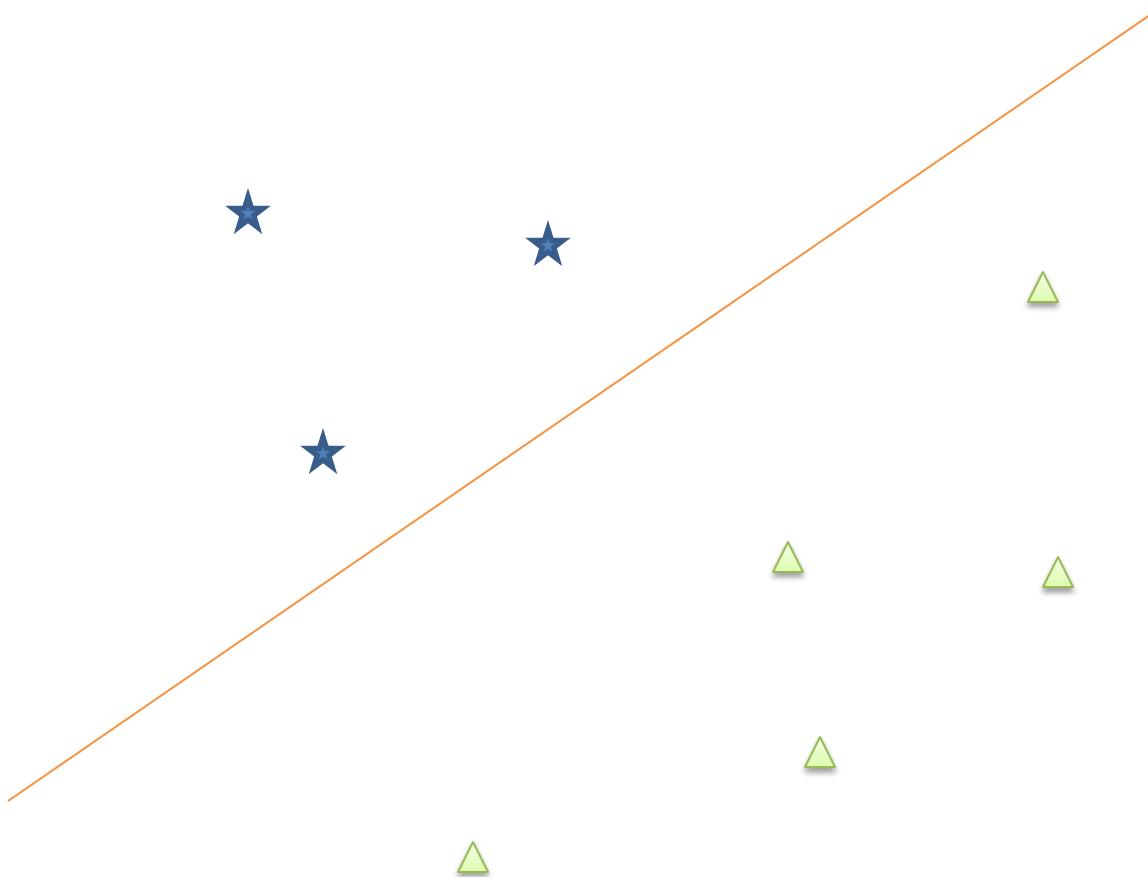
Solving a Logistic Regression

$$y_{predicted}^{(j)} = \frac{1}{1 + e^{-w.f}}$$

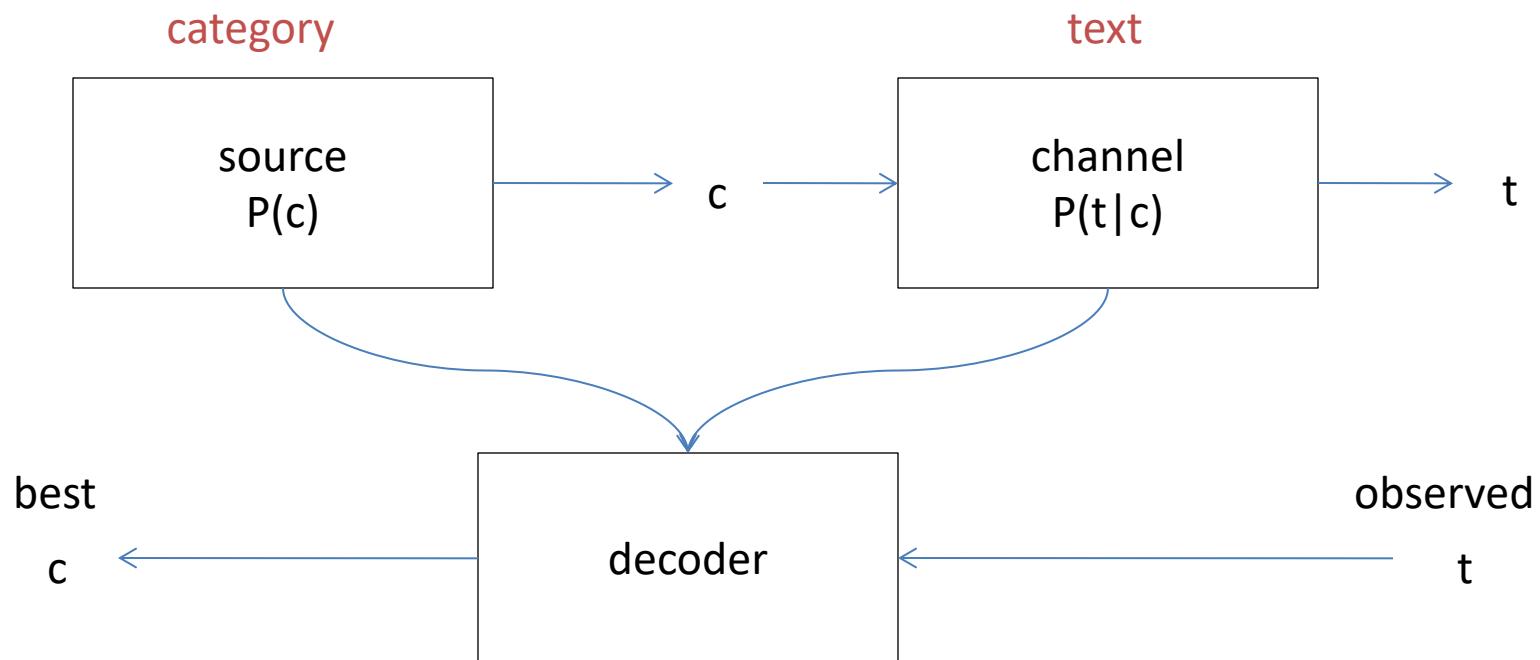
- It's not as easy as solving for 'm' in 'y = mx + c'
- You have to hill-climb over:

$$best\ w = \arg \max_w \sum_i \log P(y^{(i)} | x^{(i)})$$

Maximum Entropy Classifiers



Generative Model



$$\operatorname{argmax} P(c|t) = \operatorname{argmax} [P(c) * P(t|c)]$$

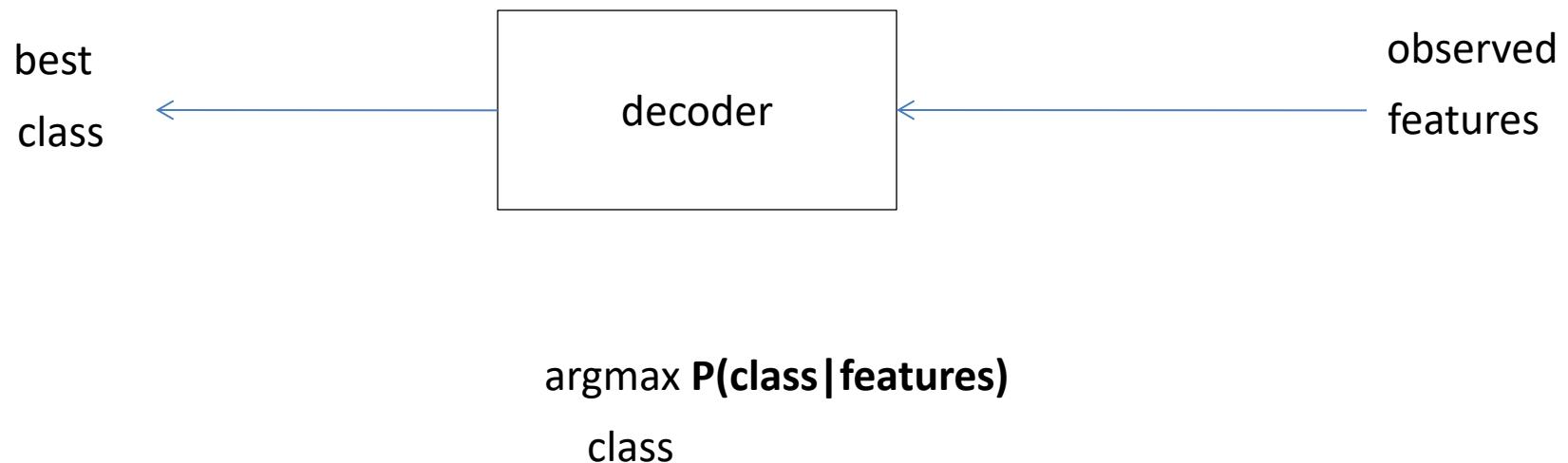
C

C

Using Bayesian Inversion

Discriminative Model

Calculate $P(\text{class} | \text{features})$ directly



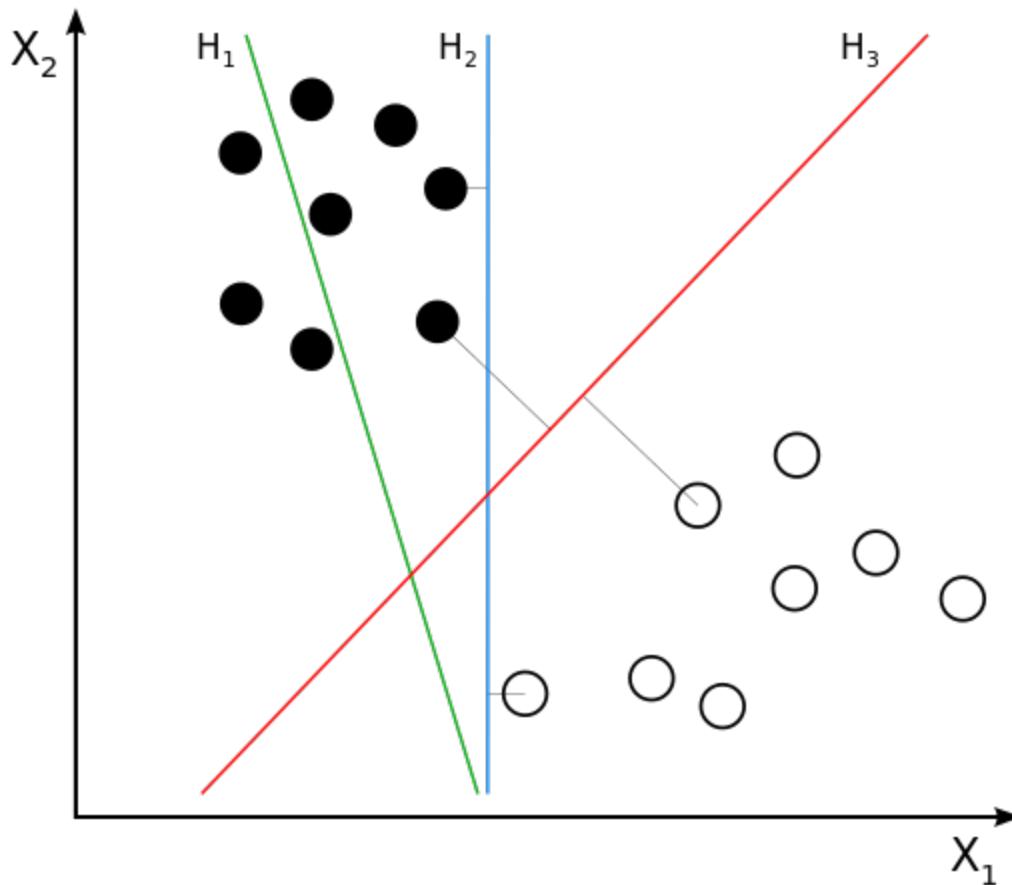
Classifiers Popular in Text Analytics

For most text analytics problems, you are deciding between one of three classifiers:

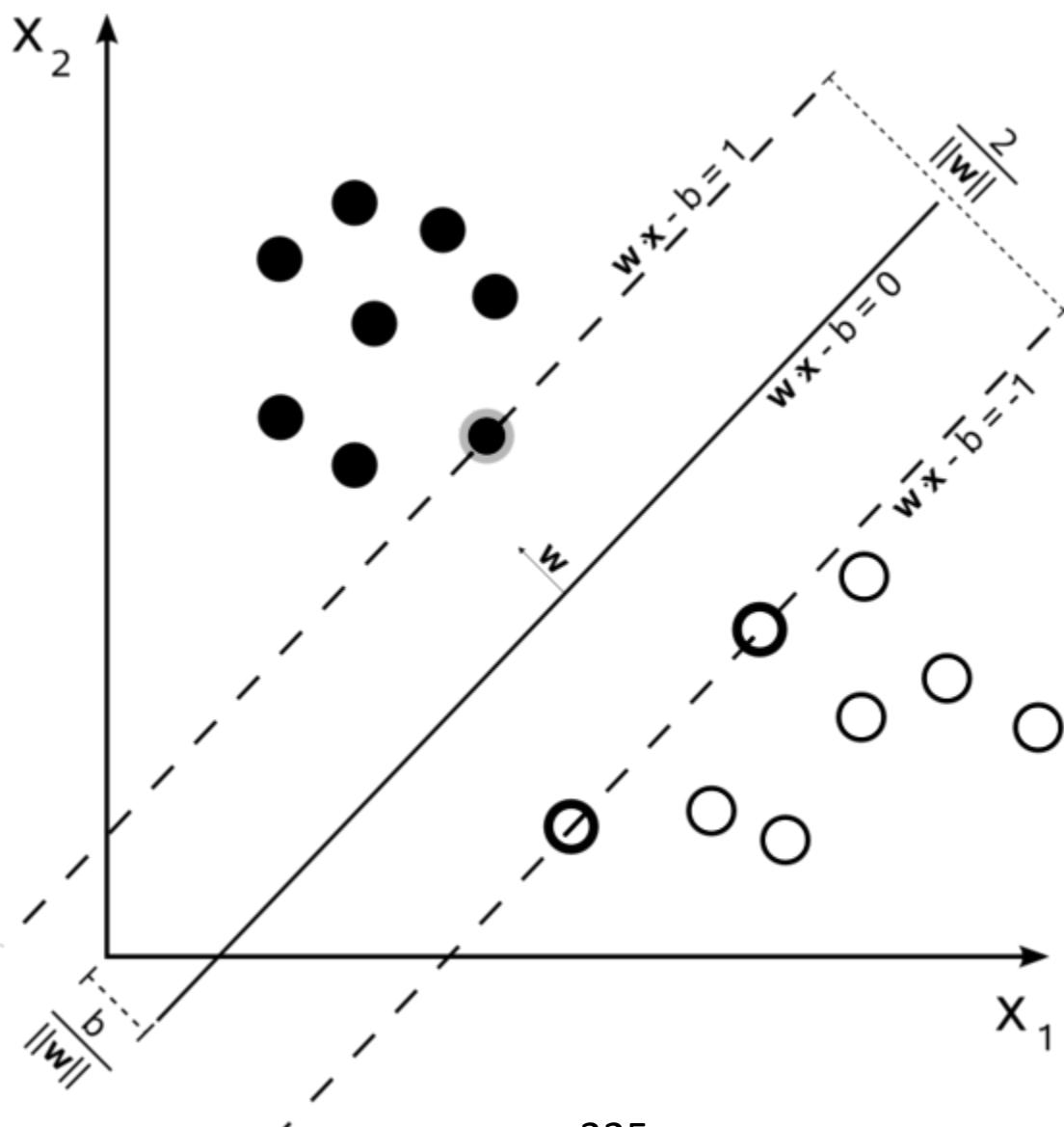
- Statistical Methods
 - Generative Statistical Models
 - Naïve Bayesian Classifier
 - Discriminative Models
 - Maximum Entropy Classifiers
- Non-Statistical
 - Support Vector Machines

Support Vector Machine Classifier

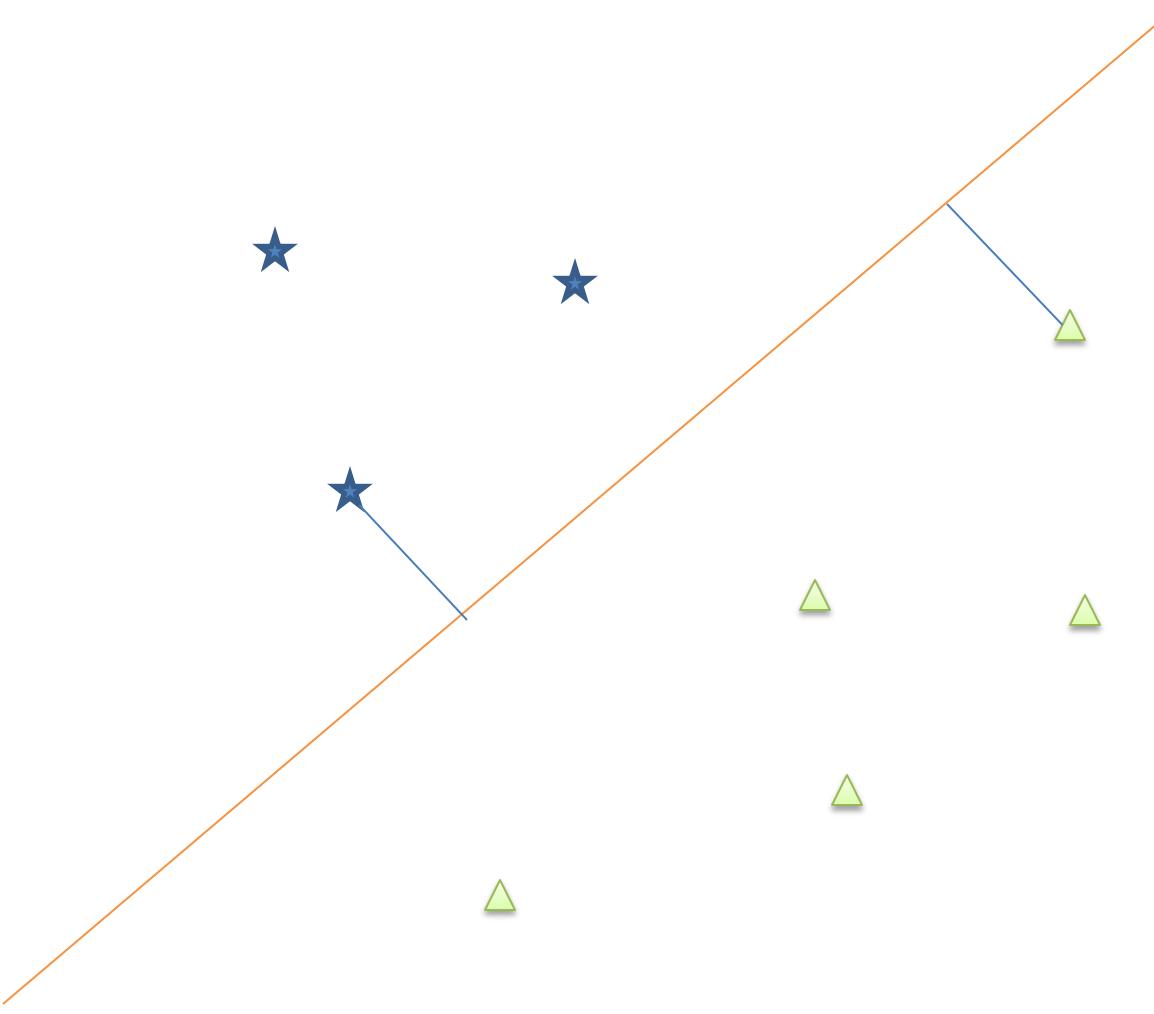
SVM Classifiers



SVM Classifiers



SVM Classifiers



Time for Exercises!

Let's practise!

1. Naïve Bayes classification
2. MaxEnt classification

Real world text analytics problem

How can you use text analytics to solve the following classification problem?

Classify the following jobs descriptions into:
full-time, part-time and contract

Real world text analytics problem

How can you use text analytics to solve the following classification problem?

Classify the following jobs descriptions into:
full-time, part-time and contract

Document 1: **full-time, contract**

Be willing to work in shifts (Morning Shift – 10 AM to 6 PM, and Afternoon – 2 PM to 10 PM), on a weekly rotational basis. Candidates shall be required to be based permanently in Mumbai for the tenure of their contract.

Real world text analytics problem

How can you use text analytics to solve the following classification problem?

Classify the following jobs descriptions into **full-time**, **part-time** and **contract** as shown below:

Document 2: **full-time, contract**

The position is for the initial period of one year, and will be extended depending on performance. Please note that this is a full-time position.

Real world text analytics problem

How can you use text analytics to solve the following classification problem?

Classify the following jobs descriptions into:
full-time, part-time and contract

Document 3: **full-time**

Global Delivery ITO Network Services is responsible for the delivery of contracted network services to contracted outsourcing customers.

Full-time/Part-time: Full-time

Real world text analytics problem

How can you use text analytics to solve the following classification problem?

Classify the following jobs descriptions into:
full-time, part-time and contract

Document 4: **full-time, part-time**

Function: IT, Human Resources

Schedule: Full-time/Part-time postings available

Can you combine classifiers?

Can you combine classifiers to solve the problem?

If so, how?

One Versus All / One Versus Rest

Classifier 1: Politics vs Rest

Politics

Finance

Sports

One Versus All / One Versus Rest

Classifier 2: Finance vs Rest

Politics

Finance

Sports

335

One Versus All / One Versus Rest

Classifier 3: Sports vs Rest

Politics

Finance

Sports

One Versus All / One Versus Rest

Several Congress and BJP leaders are **BCCI** officer bearers. The **BCCI** has **politicians** like BJP's Arun Jaitley and ...



Possibly Classifier 1 decides it's Politics and Classifier 3 decides it's Sports ...

One Versus All / One Versus Rest

Several Congress and BJP leaders are **BCCI** officer bearers. The **BCCI** has **politicians** like BJP's Arun Jaitley and ...

Politics

Sports

Finance

Also possible Classifier 1 decides it's not Politics and Classifier 3 decides it's not Sports and Classifier 2 decides it's not Finance

One Versus All / One Versus Rest

Several Congress and BJP leaders are **BCCI** officer bearers. The **BCCI** has **politicians** like BJP's Arun Jaitley and ...

Politics

Sports

Finance

Also possible Classifier 1 decides it's not Politics and Classifier 3 decides it's not Sports and Classifier 2 decides it's not Finance ... it's NOTA

Multi-label Classification

Multiple Labels

How do you assign multiple categories or multiple labels to a data point?

This is the same as answering a multi-choice question.

So, you're deciding that two answers are correct.

But remember ...

Multilabel

is not

Multiclass

This is Multiclass

A document just has to be classified into **one** of these classes.

Politics

Sports

Finance

This is Multiclass

Just means there is more than one class.

But multiclass is usually used to denote more than 2 classes.

If there are only 2 classes, it is sometimes called binary classification.

Politics

Sports

Finance

Multilabel

A document may have to be classified into **more than one** of these classes.

Politics

Sports

Finance

Multilabel

Politics

The BCCI
has **politicians** like
BJP's Arun Jaitley
and ...

Sports

On Day-One of the
auction,
Bangalore **bought**
Sri Lankan
Tillakaratne for

Finance

Measurement Exercise

Data Point	Real	Observed
D1	Politics, Sports	Sports
D2	Politics	Politics
D3	Politics, Sports	Politics
D4	Sports	Politics
D5	Sports	Sports, Politics
D6	Sports	Sports
D7	Sports	Politics
D8	Sports	Sports

What is the precision for Politics?

Measurement Exercise

Data Point	Real	Observed
D1	Politics, Sports	Sports
D2	Politics	Politics
D3	Politics, Sports	Politics
D4	Sports	Politics
D5	Sports	Sports, Politics
D6	Sports	Sports
D7	Sports	Politics
D8	Sports	Sports

What is the recall for Politics?

Measurement Exercise

Data Point	Real	Observed
D1	Politics, Sports	Sports
D2	Politics	Politics
D3	Politics, Sports	Politics
D4	Sports	Politics
D5	Sports	Sports, Politics
D6	Sports	Sports
D7	Sports	Politics
D8	Sports	Sports

What is the F-score for Politics?

Reuters Data Set

- 9603 training, 3299 test articles; ave. 200 words
- 118 categories
 - An article can be in more than one category
 - Learn 118 binary category distinctions

Common categories
(#train, #test)

- Earn (2877, 1087)
- Acquisitions (1650, 179)
- Money-fx (538, 179)
- Grain (433, 149)
- Crude (389, 189)
- Trade (369, 119)
- Interest (347, 131)
- Ship (197, 89)
- Wheat (212, 71)
- Corn (182, 56)

AN EXAMPLE OF REUTERS TEXT DOCUMENT 14832

THAI TRADE DEFICIT WIDENS IN FIRST QUARTER

Thailand's trade deficit widened to 4.5 billion baht in the first quarter of 1987 from 2.1 billion a year ago, the Business Economics Department said.

...

Export growth was smaller than expected due to lower earnings from many key commodities including **rice** whose earnings declined 18 pct, **maize** 66 pct, **sugar** 45 pct, **tin** 26 pct and canned pineapples seven pct.

Products registering high export growth were jewellery up 64 pct, clothing 57 pct and **rubber** 35 pct.

Categories: **rubber tin sugar corn rice grain trade**

THE REUTERS CATEGORIES FILE

test/14826 trade

test/14828 grain

test/14829 nat-gas crude

test/14832 rubber tin sugar corn rice grain trade

test/14833 palm-oil veg-oil

Multilabel Classification

We have our third design pattern:

Multilabel classification

When do you use it?

When data points can belong to more than one category.

Real world text analytics problem

Someone wants you to automatically evaluate essay answers for an English language test

Given an English sentence like:

It was the best of times; it was the worst of times; it was the age of wisdom; it was the age of foolishness.

Is this is a correct English sentence or not?

How do you measure the quality of a sentence?

Language Model

$s = \text{"the cat caught the mouse"}$

$$\begin{aligned} P(\text{"the cat caught the mouse"}) &= P(\text{"mouse"} | \text{"the cat caught the"}) \\ &\quad *P(\text{"the"} | \text{"the cat caught"}) \\ &\quad *P(\text{"caught"} | \text{"the cat"}) \\ &\quad *P(\text{"cat"} | \text{"the"}) *P(\text{"the"}) \end{aligned}$$

$$\begin{aligned} P(\text{"the cat caught the mouse"}) &= P(\text{"mouse"} | \text{"the"}) \\ &\quad *P(\text{"the"} | \text{"caught"}) \\ &\quad *P(\text{"caught"} | \text{"cat"}) \\ &\quad *P(\text{"cat"} | \text{"the"}) *P(\text{"the"}) \end{aligned}$$

$$P(\text{"mouse"} | \text{"the cat caught the"}) = P(\text{"mouse"} | \text{"the"})$$

Language Model

Given: $s = \text{"the cat caught the mouse"}$

Come up with a function $P(\text{"the cat caught the mouse"})$ [aka the language model]
Such that $P(\cdot)$ has the following property ...

Properties of a good language model:

$P(\text{"cat mouse the the caught"}) \ll P(\text{"the cat caught the mouse"})$
 $P(\text{"the banana ate the computer"}) \ll P(\text{"the man ate the banana"})$

Language Model

Given: $s = \text{"the cat caught the mouse"}$

Come up with a function $P(\text{"the cat caught the mouse"})$ [aka the language model]
Such that $P(\cdot)$ has the following property ...

Properties of a good language model:

$P(\text{"cat mouse the the caught"}) \ll P(\text{"the cat caught the mouse"})$
 $P(\text{"the banana ate the computer"}) \ll P(\text{"the man ate the banana"})$

We will see that sequential models work best for this!

Language Model

We could start with the chain rule:

$$P("FED") = P("F|ED") * P(E|D) * P(D)$$

Proof?

Language Model

We could start with the chain rule:

$$P("FED") = P("F|ED") * P(E|D) * P(D)$$

Proof?

$$P("FED") = P("F|ED") * P(ED)$$

So,

$$P("FED") = P("FED")$$

Language Model

Using the chain rule:

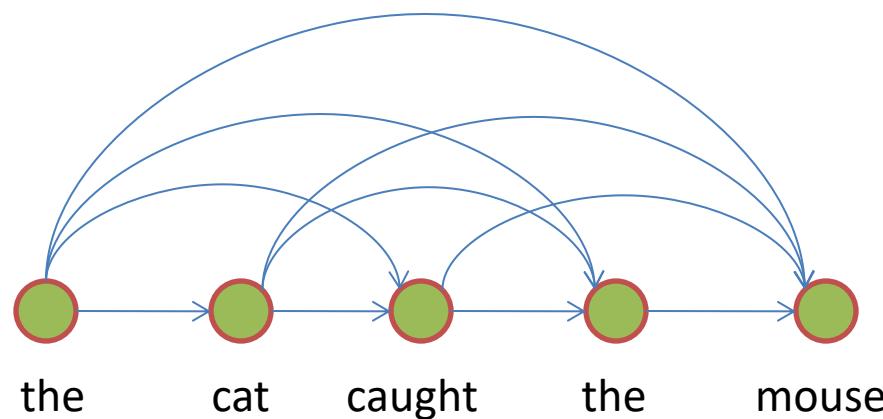
$$P("FED") = P("F|ED") * P(E|D) * P(D)$$

You get:

$$\begin{aligned} P(\text{"the cat caught the mouse"}) &= P(\text{"mouse"} | \text{"the cat caught the"}) \\ &\quad * P(\text{"the"} | \text{"the cat caught"}) \\ &\quad * P(\text{"caught"} | \text{"the cat"}) \\ &\quad * P(\text{"cat"} | \text{"the"}) * P(\text{"the"}) \end{aligned}$$

Language Model

$$P(\text{"the cat caught the mouse"}) = P(\text{"mouse"} | \text{"the cat caught the"}) \\ *P(\text{"the"} | \text{"the cat caught"}) \\ *P(\text{"caught"} | \text{"the cat"}) \\ *P(\text{"cat"} | \text{"the"}) *P(\text{"the"})$$



Problem of Over-fitting

Say $C(\text{"the cat caught the mouse"}) = 1$

Say $C(\text{"the cat caught the"}) = 1$

- You get $P(\text{"mouse"} | \text{"the cat caught the"}) = 1$

One Solution for Over-fitting

Use a smaller model (make a Markov assumption)

$$p(w_n \mid w_1..w_{n-1}) \approx p(w_n \mid w_{n-N+1}..w_{n-1})$$

Example: Unigrams

$$\begin{aligned} P(\text{"caught the"}) &= P(\text{"the"} \mid \text{"caught"})P(\text{"caught"}) \\ &= P(\text{"the"})P(\text{"caught"}) \end{aligned}$$

Big problem with unigrams

- $P(\text{"the the the the"}) > P(\text{"I like ice cream"})$
- $P(\text{"I an"}) > P(\text{"I am"})$

Another solution - Bigrams

$$P(\text{"caught the"}) = P(\text{"caught"} | \langle s \rangle)P(\text{"the"} | \text{"caught"})$$

$\langle s \rangle$ stands for start of sentence

Markov Assumption

Your memory of history is limited.

Say your memory of history is 1.

The equation $P("FED") = P("F|ED")*P(E|D)*P(D)$ becomes

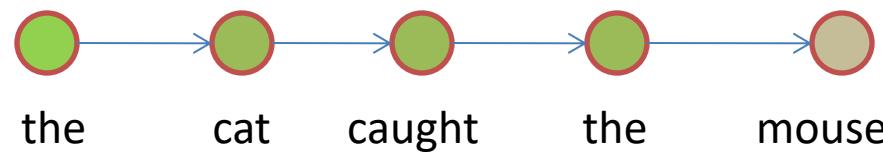
$$P("FED") = P("F|E")*P(E|D)*P(D)$$

You get:

$$\begin{aligned} P("the cat caught the mouse") &= P("mouse" | "the cat caught the") \\ &\quad * P("the" | "the cat caught") \\ &\quad * P("caught" | "the cat") \\ &\quad * P("cat" | "the") * P("the") \end{aligned}$$

Language Model – with Markov Assumption

$$\begin{aligned} P(\text{"the cat caught the mouse"}) = & \quad P(\text{"mouse"} | \text{"the"}) \\ & *P(\text{"the"} | \text{"caught"}) \\ & *P(\text{"caught"} | \text{"cat"}) \\ & *P(\text{"cat"} | \text{"the"}) \\ & *P(\text{"the"}) \end{aligned}$$



Formalism 1

So, let's try bigrams

- $P(\text{"the the the the"}) < P(\text{"I like ice cream"})$
- $P(\text{"I an"}) < P(\text{"I am"})$

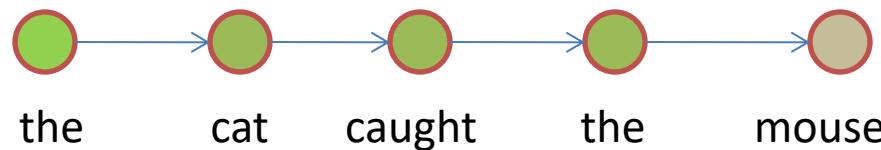
So far, so good!

- $P(\text{I saw a van}) \gg P(\text{eyes awe of an})$
- $P(\text{artichokes intimidate zippers}) \approx 0$

Nice!

Language Model – Parameters

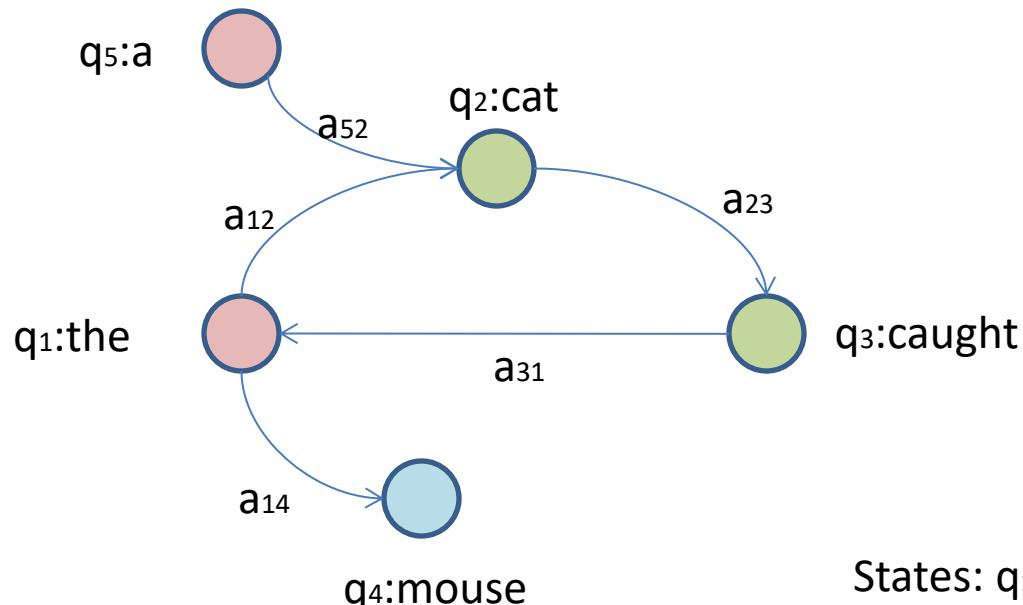
$$\begin{aligned} P(\text{"the cat caught the mouse"}) = & \quad P(\text{"mouse"} | \text{"the"}) \\ & *P(\text{"the"} | \text{"caught"}) \\ & *P(\text{"caught"} | \text{"cat"}) \\ & *P(\text{"cat"} | \text{"the"}) \\ & *P(\text{"the"}) \end{aligned}$$



The features are the words ... in this case, "the", "cat", etc.

The $p(\text{feature} | \text{previous feature})$ are the parameters (transition probabilities).

Visible Markov Model



States: $q_1 q_2 q_3$

Start states: $q_1 q_5$

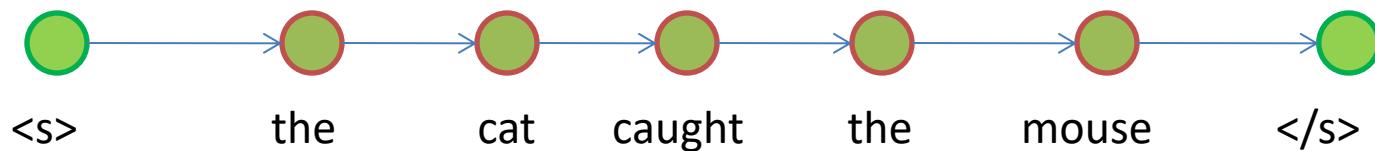
End states: q_4

Transition Probabilities: $A = \{a_{11} \ a_{12} \ a_{13} \ a_{14} \dots\}$

Initial Probabilities: $P = \{p_1 \ p_5\}$

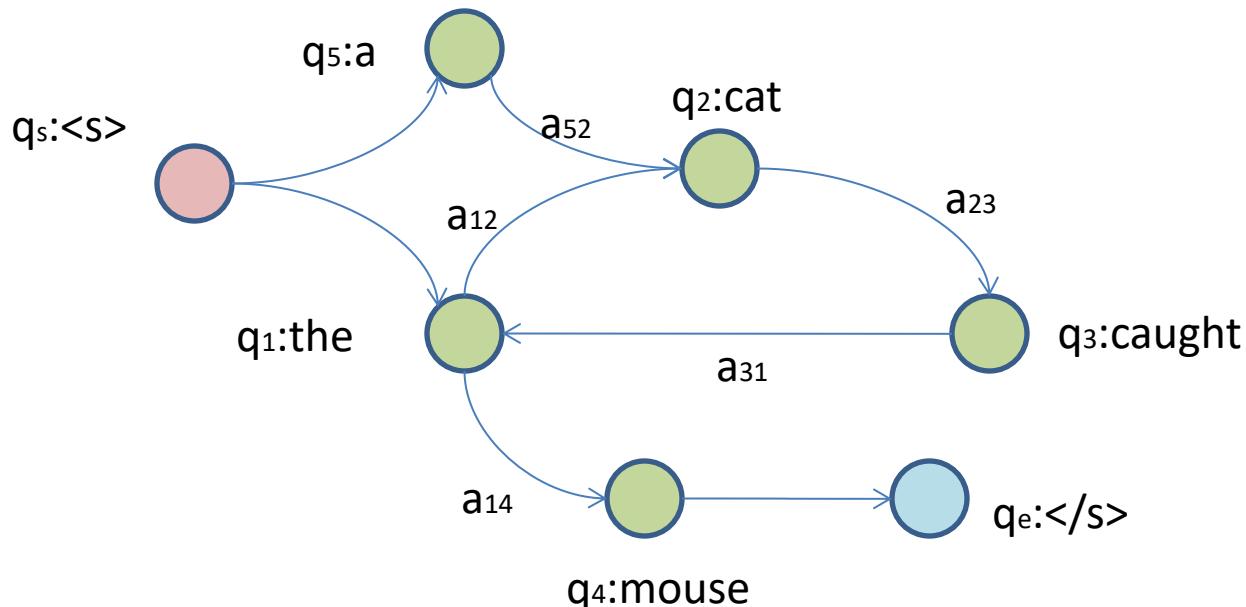
Language Model – with Markov Assumption

$$\begin{aligned} P(\text{"the cat caught the mouse"}) = & \quad P(\text{"mouse"} | \text{"the"}) \\ & *P(\text{"the"} | \text{"caught"}) \\ & *P(\text{"caught"} | \text{"cat"}) \\ & *P(\text{"cat"} | \text{"the"}) \\ & *P(\text{"the"} | \text{<start>}) \end{aligned}$$



Formalism 2

Visible Markov Model



States: $q_1 q_2 q_3 \dots$

Start state: q_s

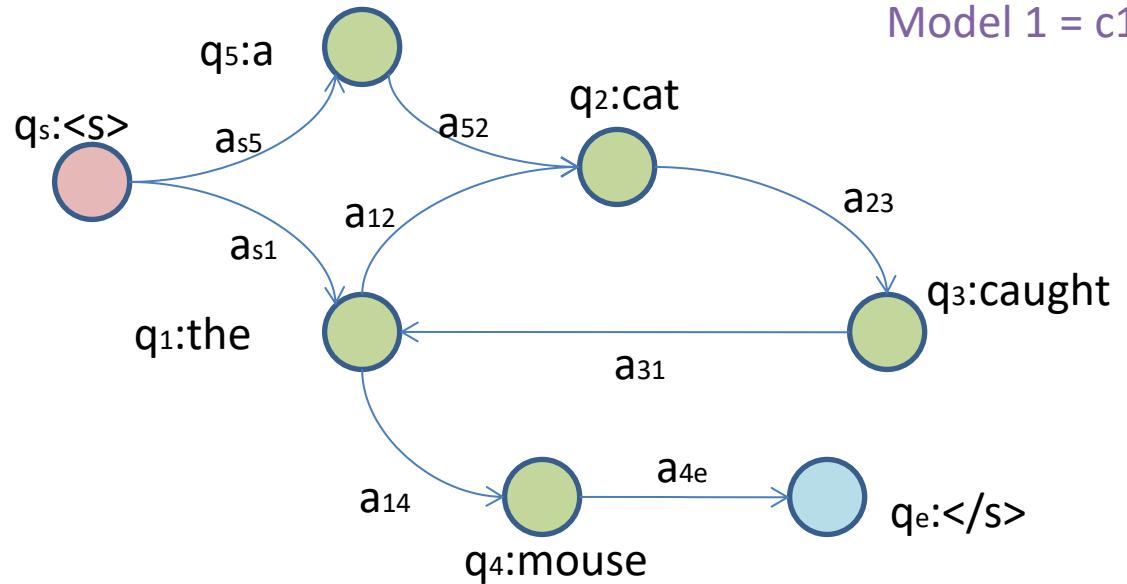
End state: q_e

Transition Probabilities: $A = \{a_{11} a_{12} a_{13} a_{14} \dots\}$

Initial Probabilities: $P = \{p_1 p_5\}$

How do you train a Language Model

$a_{s1} = ?$
 $a_{s5} = ?$
 $a_{12} = ?$
 $a_{14} = ?$



Sentences:

A cat caught the mouse

The cat caught the mouse

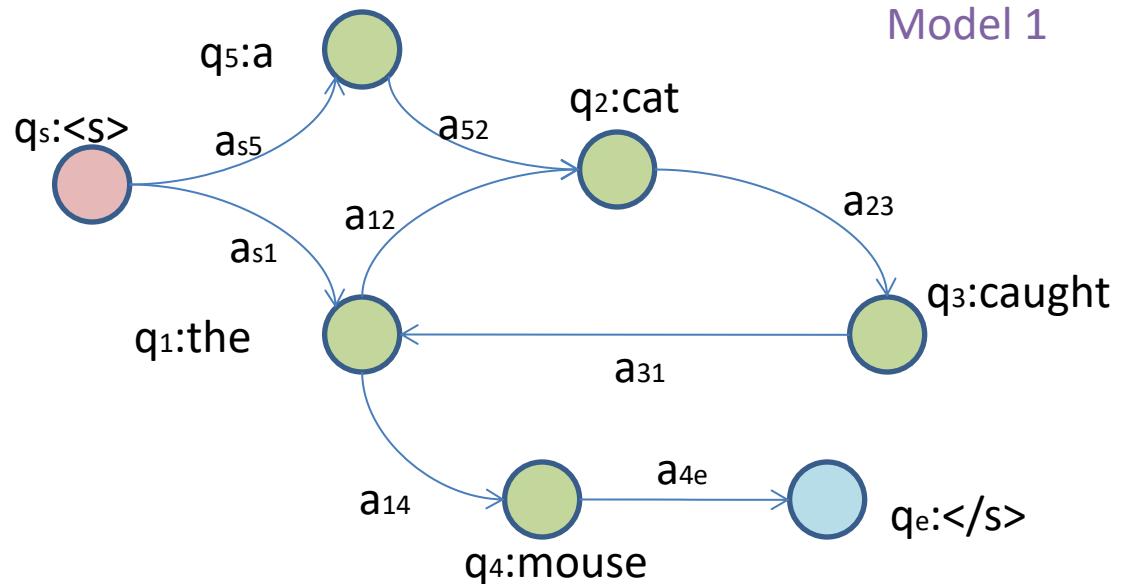
The mouse

Language Model – with Markov Assumption

- Evaluation
 - Given model m and observations O , what is $P(O|m)$
 - Let's say the observations are “the mouse”

Language Model – What is the Probability of a Sentence

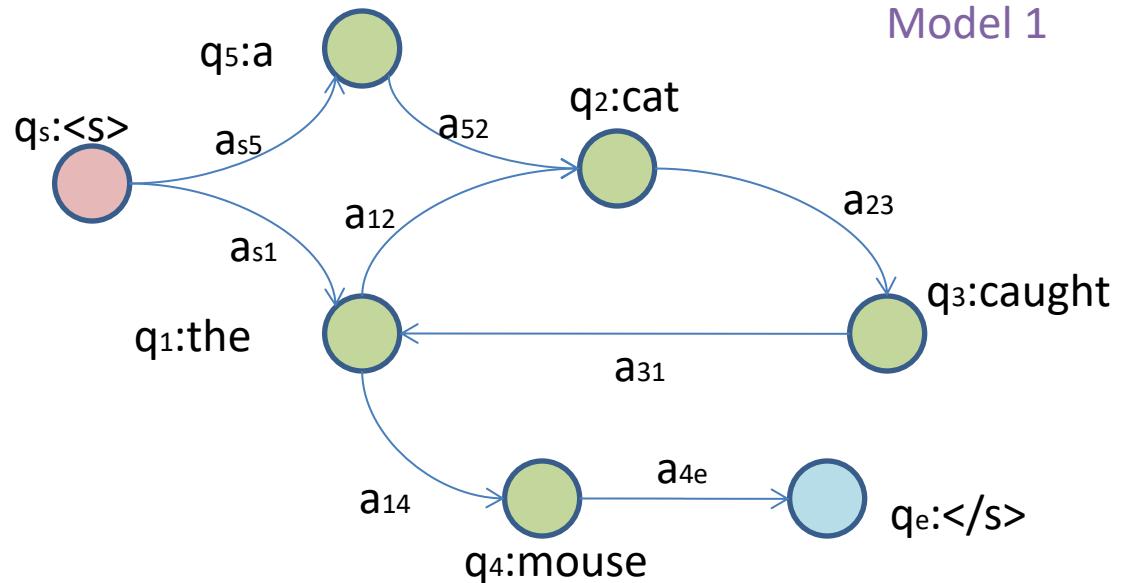
$a_{s1} = 0.2$
 $a_{s5} = 0.8$
 $a_{12} = 0.5$
 $a_{14} = 0.5$
other $a_{xy} = 1$



Sentence: **The mouse**

Language Model – What is the Probability of a Sentence

$a_{s1} = 0.2$
 $a_{s5} = 0.8$
 $a_{12} = 0.5$
 $a_{14} = 0.5$
other $a_{xy} = 1$

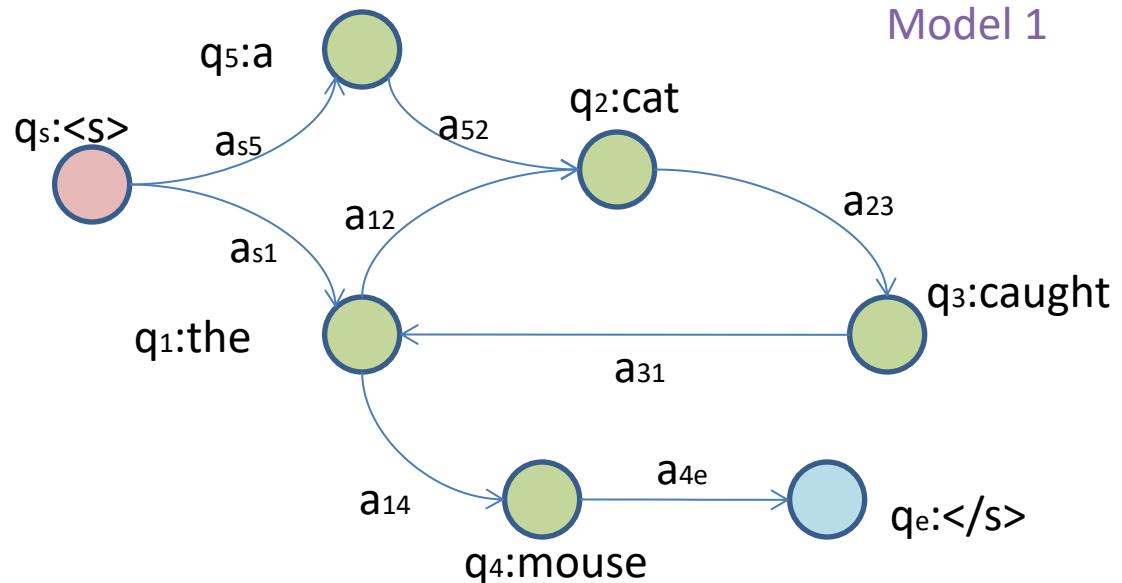


Sentence: **The mouse**

$$\begin{aligned} P(s) &= a_{s1} * a_{14} * a_{4e} \\ &= ? \end{aligned}$$

Language Model – What is the Probability of a Sentence

$a_{s1} = 0.2$
 $a_{s5} = 0.8$
 $a_{12} = 0.5$
 $a_{14} = 0.5$
other $a_{xy} = 1$

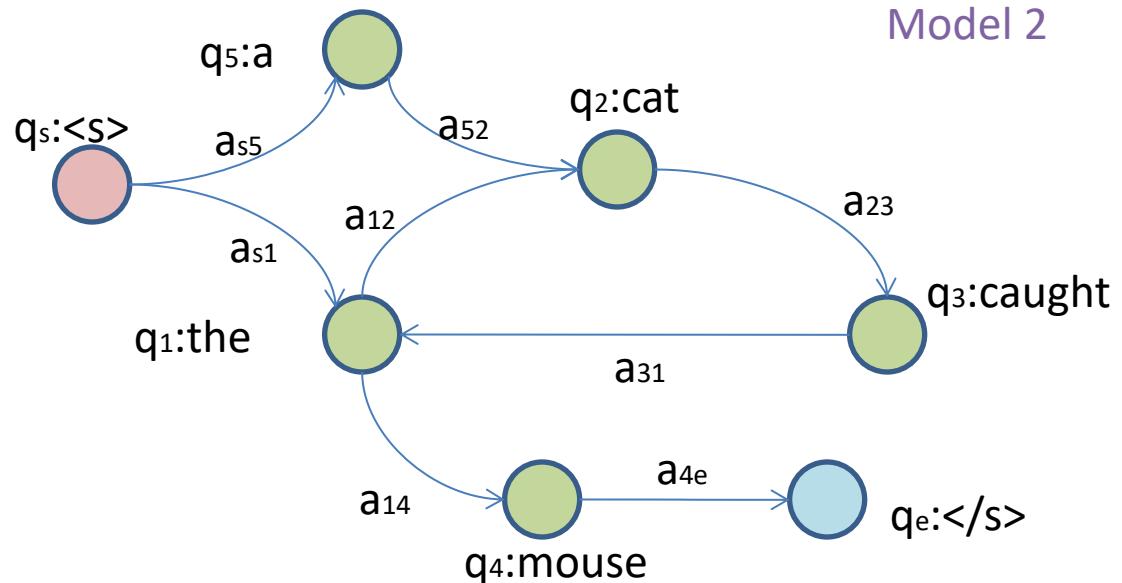


Sentence: **The mouse**

$$\begin{aligned} P(s) &= a_{s1} * a_{14} * a_{4e} \\ &= 0.2 * 0.5 * 1 = \textcolor{violet}{0.1} \end{aligned}$$

Language Model – What is the Probability of a Sentence

$a_{s1} = 0.8$
 $a_{s5} = 0.2$
 $a_{12} = 0.5$
 $a_{14} = 0.5$
other $a_{xy} = 1$



Sentence: **The mouse**

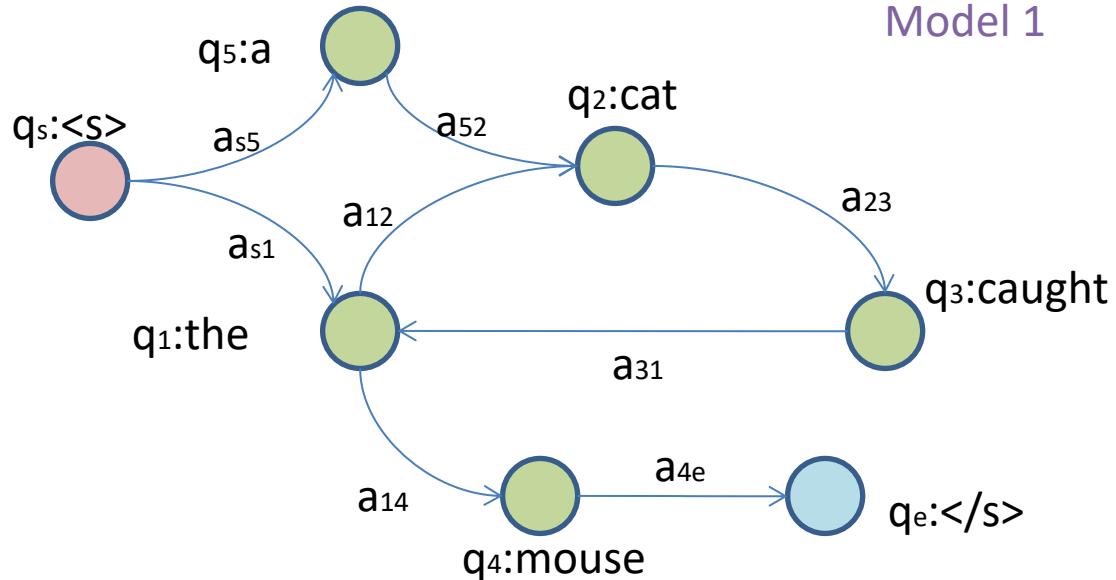
$$\begin{aligned} P(s) &= a_{s1} * a_{14} * a_{4e} \\ &= 0.8 * 0.5 * 1 = 0.4 \end{aligned}$$

Language Model – What is the Probability of a Sentence

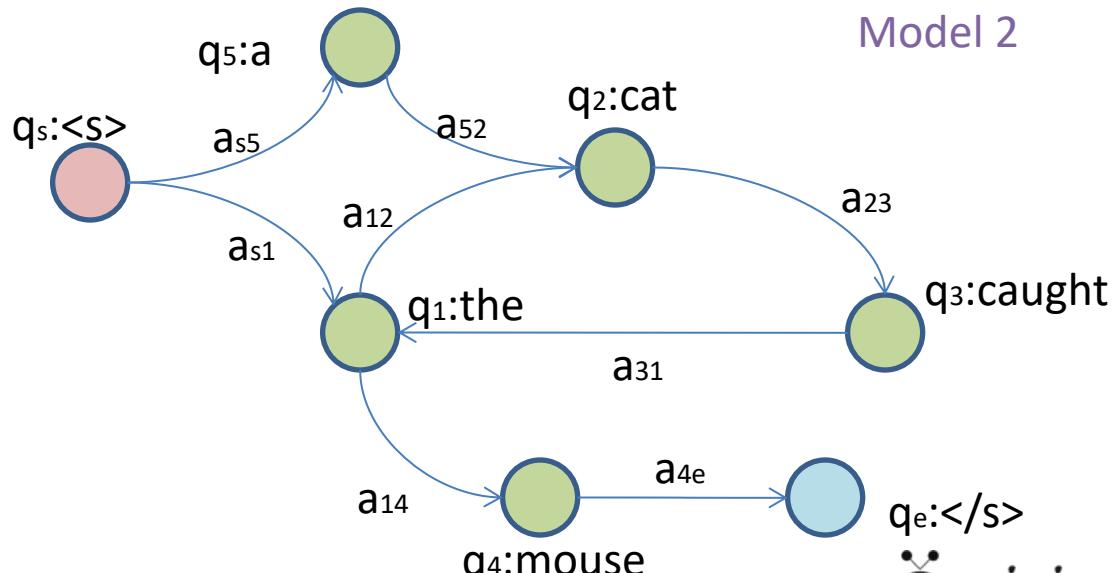
You try the sentence:

The cat caught the mouse

$a_{s1} = 0.2$
 $a_{s5} = 0.8$
 $a_{12} = 0.5$
 $a_{14} = 0.5$
 other $a_{xy} = 1$

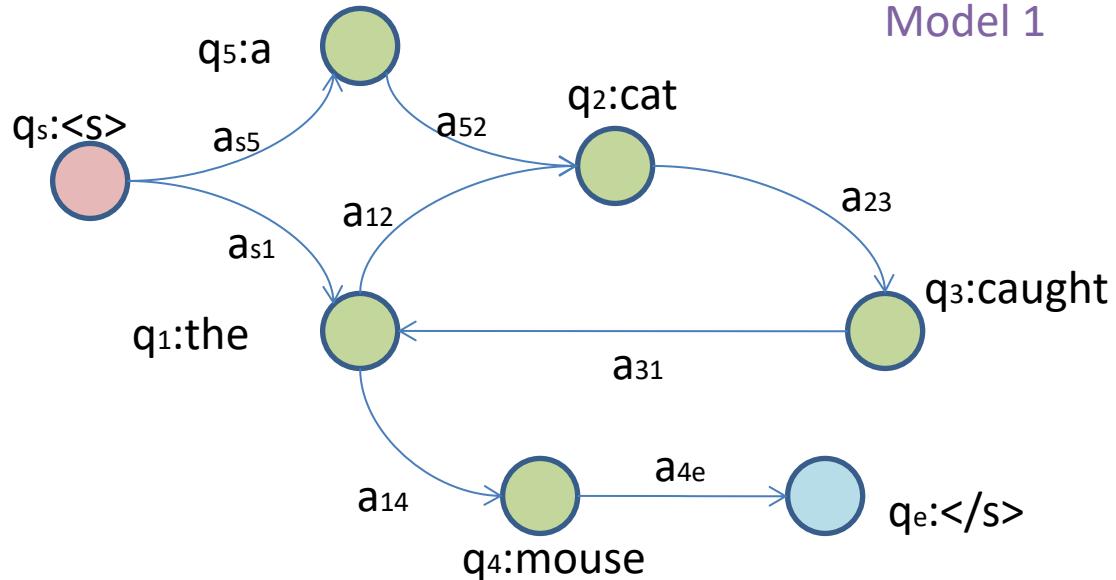


$a_{s1} = 0.8$
 $a_{s5} = 0.2$
 $a_{12} = 0.5$
 $a_{14} = 0.5$
 other $a_{xy} = 1$



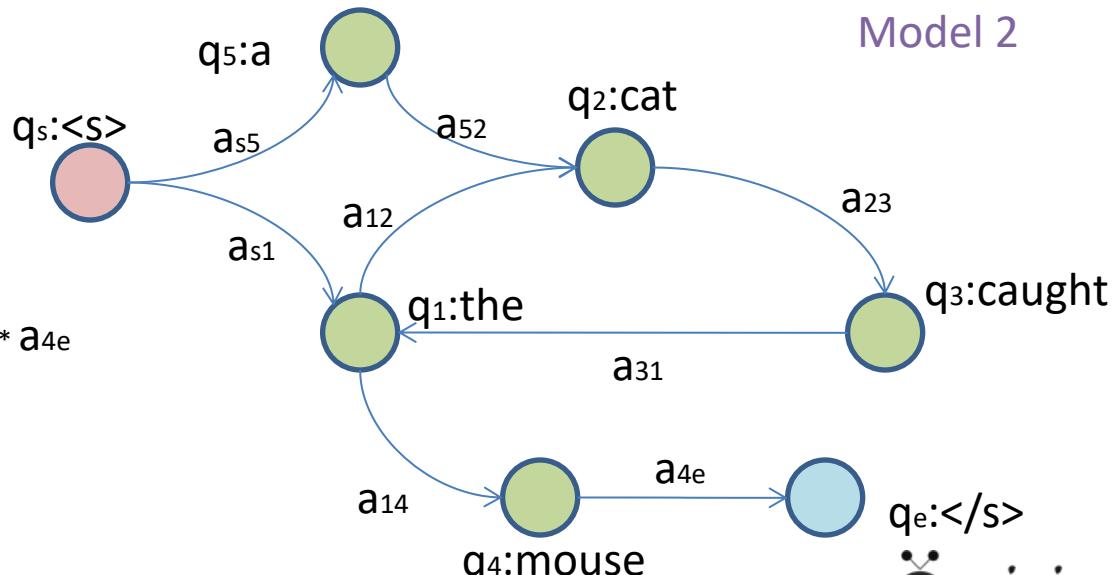
$a_{s1} = 0.2$
 $a_{s5} = 0.8$
 $a_{12} = 0.5$
 $a_{14} = 0.5$
 other $a_{xy} = 1$

$$P(s) = a_{s1} * a_{12} * a_{23} * a_{31} * a_{14} * a_{4e}$$



$a_{s1} = 0.8$
 $a_{s5} = 0.2$
 $a_{12} = 0.5$
 $a_{14} = 0.5$
 other $a_{xy} = 1$

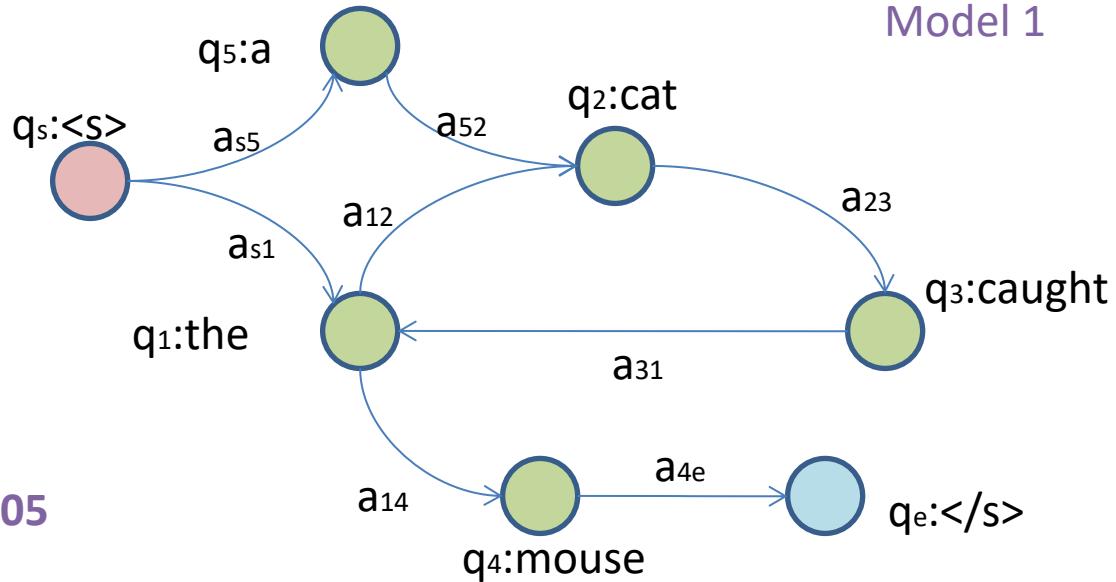
$$P(s) = a_{s1} * a_{12} * a_{23} * a_{31} * a_{14} * a_{4e}$$



$a_{s1} = 0.2$
 $a_{s5} = 0.8$
 $a_{12} = 0.5$
 $a_{14} = 0.5$
 other $a_{xy} = 1$

$$P(s) = a_{s1} * a_{12} \\ * a_{23} * a_{31} * a_{14} * a_{4e}$$

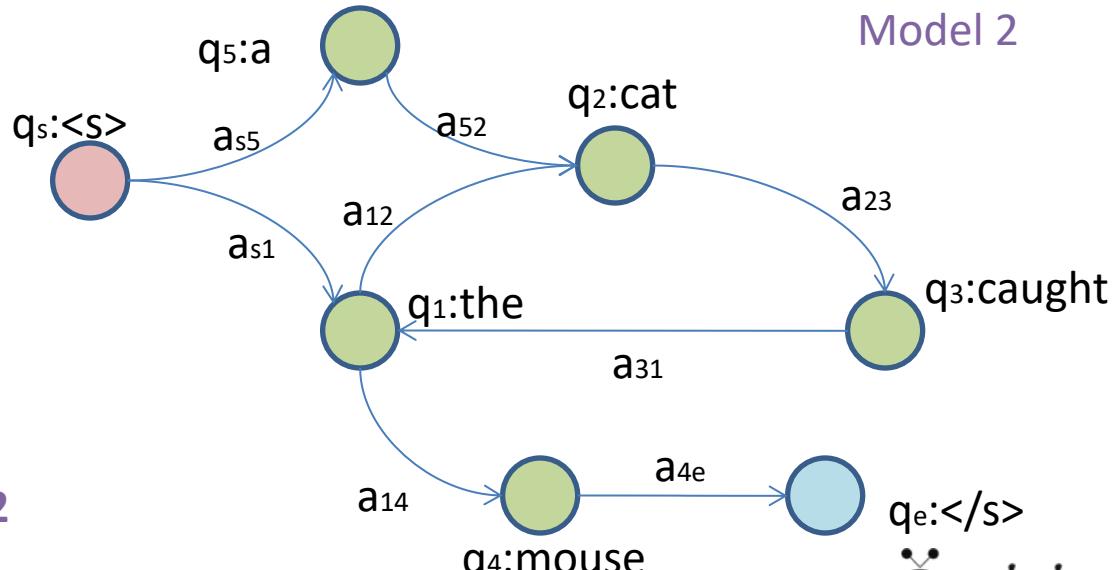
$$= 0.2 * 0.5 * 1 * 1 * 0.5 * 1 = 0.05$$



$a_{s1} = 0.8$
 $a_{s5} = 0.2$
 $a_{12} = 0.5$
 $a_{14} = 0.5$
 other $a_{xy} = 1$

$$P(s) = a_{s1} * a_{12} \\ * a_{23} * a_{31} * a_{14} * a_{4e}$$

$$= 0.8 * 0.5 * 1 * 1 * 0.5 * 1 = 0.2$$



Why is multiplying by 0 bad?

Assume ‘Vishwanathan’ is not in the corpus.

$$C(\text{"met Vishwanathan"}) = 0$$

Problem of Unknown words

$$C(\text{"met"}) \approx 30000$$

Unigram

$$\begin{aligned} P(\text{"met Vish.."}) &= P(\text{"met"})P(\text{"Vish.."}) \\ &= 0.0004 * 0 \end{aligned}$$

Bigram

$$\begin{aligned} P(\text{"met Vish.."}) &= P(\text{"met"} | \dots)P(\text{"Vish.."} | \text{"met"}) \\ &= 0.023 * 0 \end{aligned}$$

kills other estimates!

Quick puzzle

- What's the usual part of speech of unknown words?

Examples of Part of Speech:

Verbs, Adjectives, Adverbs, Conjunctions, Common Nouns,
Proper Nouns, Definite Articles, Indefinite Articles

Classes of Words:

Closed – pronouns, prepositions, particles (function words)
Open – nouns, verbs, adjectives (content words)

SMOOTHING

- Laplace, Lidstone, Jeffreys-Perks
- Witten-Bell
- Good-Turing
- Katz's Backoff
- Kneser-Ney
- Linear Interpolation

Definitions

- Word types
- Word tokens
- N : size of corpus [count tokens]
- V : size of vocabulary [count types]
- B : parameter size of the model
 - unigram $B=V$
 - bigram $B=V^2$
 - trigram $B=V^3$
 - n -gram $B=V^n$
- $C(w_1 \dots w_n)$ is the count of an n -gram
- C_t is the count of word w_t
- N_r : The number of n -grams with count r

LAPLACE SMOOTHING

Unigrams:

$$P_{ML}(w_t) = \frac{c_t}{N}$$

$$P_{Laplace}(w_t) = \frac{c_t + 1}{N + V}$$

LAPLACE SMOOTHING

Google Terabyte Corpus

$$P_{ML}(w_t) = \frac{c_t}{N}$$

$$P_{ML}("that") = \frac{c("that")}{N} = \frac{3400063273}{1024908267229} \approx 0.0033$$

$$P_{Laplace}("that") = \frac{c("that") + 1}{N + 4958224} = \frac{3400063273 + 1}{1024908267229 + 4958224} \approx 0.0033$$

LAPLACE SMOOTHING

Brown Corpus Unigrams

$$P_{ML}(w_t) = \frac{c_t}{N}$$

$$P_{ML}("that") = \frac{c("that")}{N} = \frac{10237}{1161192} \approx 0.0088$$

$$P_{Laplace}("that") = \frac{c("that") + 1}{N + V} = \frac{10237 + 1}{1161192 + 56057} \approx 0.0084$$

LAPLACE SMOOTHING

Brown Corpus Bigrams

$$P_{ML}(w_t) = \frac{c_t}{N}$$

$$P_{Laplace}("that\ he") = \frac{c("that\ he")}{N} = \frac{559}{1161192} \approx 4.8 * 10^{-4}$$

$$P_{Laplace}("that\ he") = \frac{c("that\ he") + 1}{N + B} = \frac{559 + 1}{1161192 + 56057 * 56057} \approx 1.78 * 10^{-7}$$

$$56057 * 56057 = 3.1423 * 1,000,000,000$$

LAPLACE SMOOTHING

Brown Corpus

Conditional Probabilities

$$P_{ML}(w_t \mid w_1 \dots w_{t-1}) = \frac{c(w_1 \dots w_t)}{c(w_1 \dots w_{t-1})}$$

$$P_{Laplace}("that he") = \frac{c("that he")}{c("that")} = \frac{559}{10237} \approx 0.055$$

$$P_{Laplace}("that he") = \frac{c("that he") + 1}{c("that") + V} = \frac{559 + 1}{10237 + 56057} \approx 0.0084$$

LIDSTONE's

$$P_{Lidstone}(w_t) = \frac{c_t + \lambda}{N + \lambda V}$$

JEFFREY-PERK's

$$\lambda = \frac{1}{2}$$

$$P_{Lidstone}(w_t) = \frac{c_t + \lambda}{N + \lambda V}$$

Language Model – Challenge

Can you build a classifier out of a language model?

Language Model – Challenge

Can you build a classifier out of this?

You just calculated $P(s)$

**For classification you need to
compute $P(c|s)$**

Language Model – Challenge

Can you build a classifier out of this?

You just calculated $P(s)$

**For classification you need to
compute $P(c|s)$**

Use Bayesian Inversion!!!

Language Model – Challenge

$$a_{s1} = 0.2$$

$$a_{s5} = 0.8$$

$$a_{12} = 0.5$$

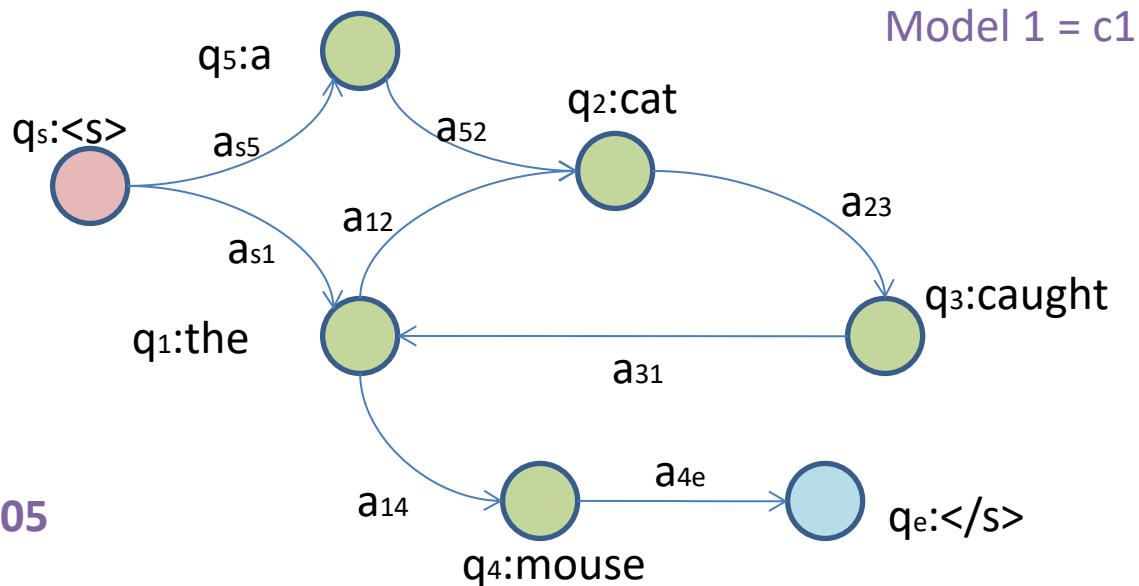
$$a_{14} = 0.5$$

other $a_{xy} = 1$

$$P(s) = a_{s1} * a_{12}$$

$$* a_{23} * a_{31} * a_{14} * a_{4e}$$

$$= 0.2 * 0.5 * 1 * 1 * 0.5 * 1 = \textcolor{purple}{0.05}$$



$$P(c|s) = P(s|c) * P(c) / P(s)$$

$$P(c|s) = a_{s1} * a_{12}$$

$$* a_{23} * a_{31} * a_{14} * a_{4e} * P(c) / P(s)$$

Chomsky's Disagreement

- In 1969 he wrote:
- But it must be recognized that the notion of "probability of a sentence" is an entirely useless one, under any known interpretation of this term.

Chomsky's Argument

- A completely new sentence must have a probability of 0, since it is an outcome that has not been seen.
- Since novel sentences are in fact generated all the time, there is a contradiction.

Language Model – Challenge

Defend the notion of the probability of a sentence
against Chomsky

UNKNOWN SENTENCE

– $P(\text{"Cohan is short"})$

— =

$$P(\text{"short" | "is"}) * P(\text{"is" | "Cohan"}) * P(\text{"Cohan"})$$

— =

$$C(\text{"is short"})/C(\text{"is"})$$
$$* C(\text{"Cohan is"})/C(\text{"Cohan"})$$
$$* C(\text{"Cohan"})/C(\text{"*"})$$

Solved!

Real world text analytics problem

Someone wants you to build a tool to improve the quality of their OCR system

Their OCR recognizes words incorrectly

times -> +imes

makes -> make5

Can you correct these mistakes?

Spell Checking

Let's say the word 'hand' is mistyped

Hand --- *Hamd

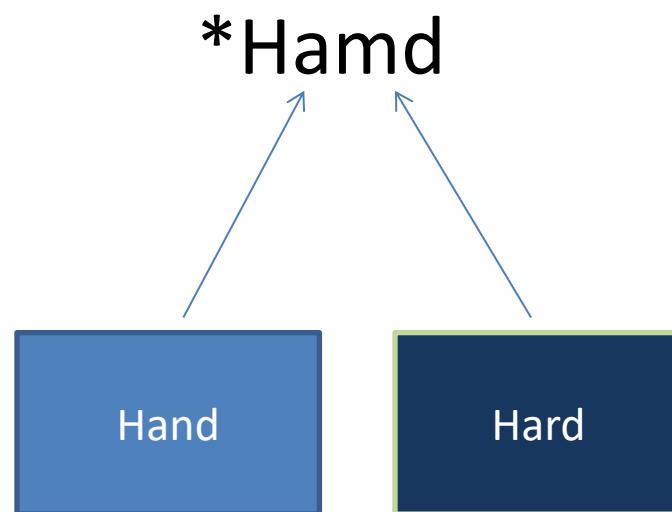
There you have an unknown word!

Spell Checking

Out-of-Vocabulary Error

*Hamd

Spell Checking



What we need to find ...

- $P(\text{hard} \mid \text{hamd})$
- $P(\text{hand} \mid \text{hamd})$
- Whichever is greater is the right one!

What we need to find ...

- But how do you find the probability $P(\text{hard} \mid \text{hamd})$ when ‘hamd’ is an *unknown word*?

This is related to Chomsky’s conundrum of the unknown sentence, isn’t it?

BAYES RULE

- Conditional Prob:
 - $P(F|E) = P(F|E) * P(E)$
 - $P(E|F) = P(E|F) * P(F)$
- $P(E|F) = P(F|E) * P(E) / P(F)$

BAYES RULE

- $P(E|F) = P(F|E) * P(E) / P(F)$
- $P(\text{hand}|\text{hamd}) = P(\text{hamd}|\text{hand}) * P(\text{hand}) / P(\text{hamd})$

Make a Massive Independence Assumption

$$P(\text{hamd} | \text{hand}) = P(h|h)P(a|a)P(m|n)P(d|d)$$

Kernighan's paper on spell correction

1990

Typo	Correction	Transformation	
acress	actress	@ t 2	deletion
acress	cress	a # 0	insertion
acress	caress	ac ca 0	reversal
acress	access	r c 2	substitution
acress	across	e o 3	substitution
acress	acres	s # 4	insertion
acress	acres	s # 5	insertion

BAYES RULE

- $P(E|F) = P(F|E) * P(E) / P(F)$
- $P(\text{hand}|\text{hamd}) = P(\text{hamd}|\text{hand}) * P(\text{hand}) / Z$

$$P(\text{hamd}|\text{hand}) = P(h|h)P(a|a)P(m|n)P(d|d)$$

The stuff in green is approximately 1, so ignore it!

- $P(\text{hand}|\text{hamd}) = P(m|n) * P(\text{hand}) / Z$

So we have what we needed to find ...

- $P(\text{hard} | \text{hamd}) = P(m | n) * P(\text{hand}) / Z$
- $P(\text{hand} | \text{hamd}) = P(m | r) * P(\text{hard}) / Z$
- Whichever is greater is the right one!

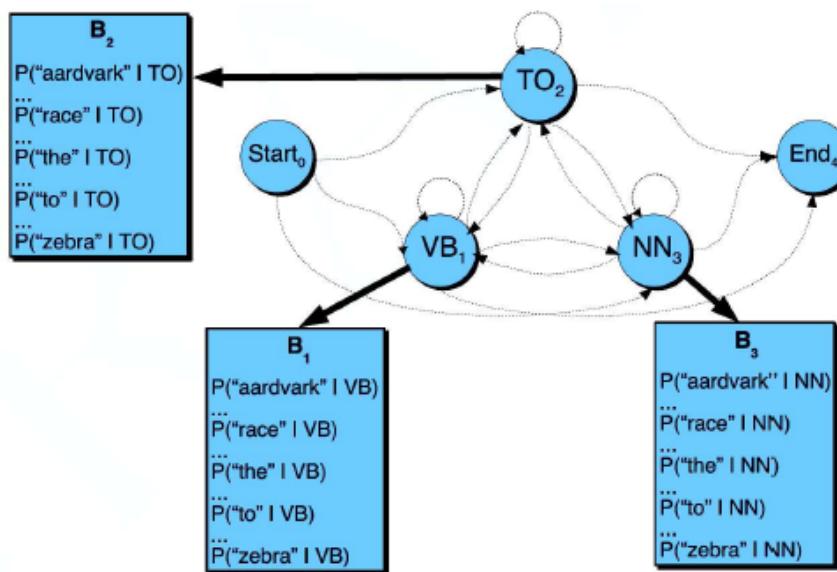
Note: Use of Unigram Probabilities

- The $P(m|n)$ part is called the channel probability
- The $P(\text{hand})$ part is called the prior probability
- Kernighan's experiment showed that $P(\text{hand})$ is very important (causes a 7% improvement)!

Method	Discrimination	%
<i>correct</i>	286/329	87 ± 1.9
no-prior	263/329	80 ± 2.2
no-channel	247/329	75 ± 2.4
neither	172/329	52 ± 2.8

Hidden Markov Model (HMM)

Transitions and Emissions



Hidden Markov Model (HMM)

Formalism

$Q = q_1 q_2 \dots q_N$

a set of N states

$A = a_{11} a_{12} \dots a_{n1} \dots a_{nn}$

a **transition probability matrix** A , each a_{ij} representing the probability of moving from state i to state j , s.t. $\sum_{j=1}^n a_{ij} = 1 \quad \forall i$

$O = o_1 o_2 \dots o_T$

a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \dots, v_V$.

$B = b_i(o_t)$

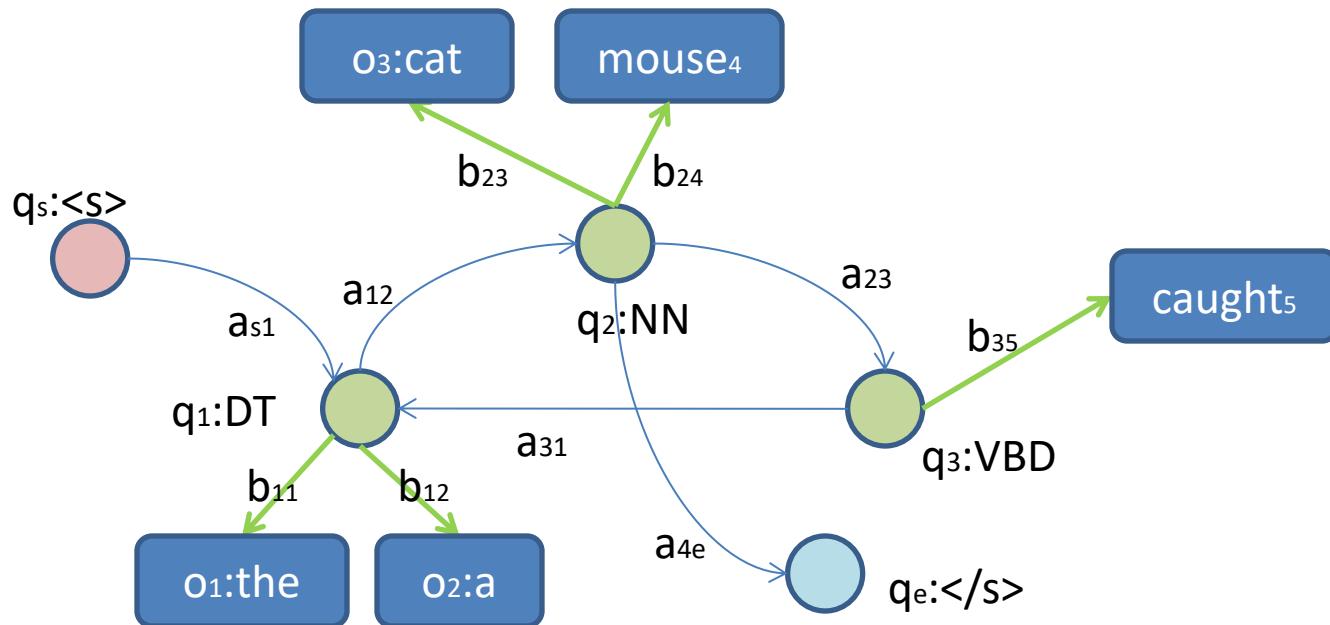
A sequence of **observation likelihoods**:, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i .

q_0, q_F

a special **start state** and **end (final) state** which are not associated with observations, together with transition probabilities $a_{01} a_{02} \dots a_{0n}$ out of the start state and $a_{1F} a_{2F} \dots a_{nF}$ into the end state.

Courtesy of the textbook – Jurafsky and Martin

Hidden Markov Model (HMM)



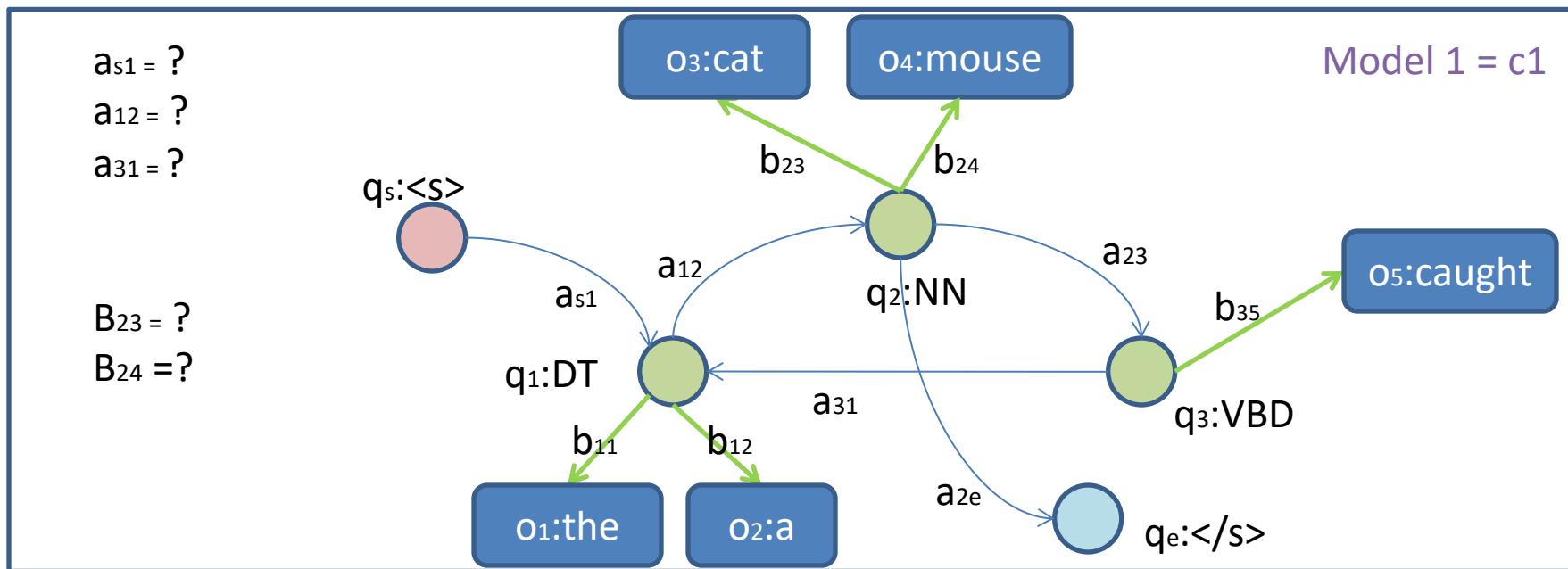
Representation 1: trained model

States: $q_1 q_2 q_3 \dots$
Start, end states: q_s, q_e

Observations: $o_1 o_2 o_3 \dots$

Transition Probabilities: $A = \{a_{11} a_{12} a_{13} a_{14} \dots\}$
Emission Probabilities: $B = \{b_{ij} \dots\}$

How do you train an HMM



Transition Probabilities: $A = \{a_{s1} \ a_{12} \ a_{23} \ a_{31} \dots\}$ Emission Probabilities: $B = \{b_{ij} \dots\}$

Sentences:

A/DT cat/NN caught/VBD the/DT mouse/NN

The/DT cat/NN caught/VBD the/DT mouse/NN

The/DT mouse/NN

Example of a Trained HMM Model

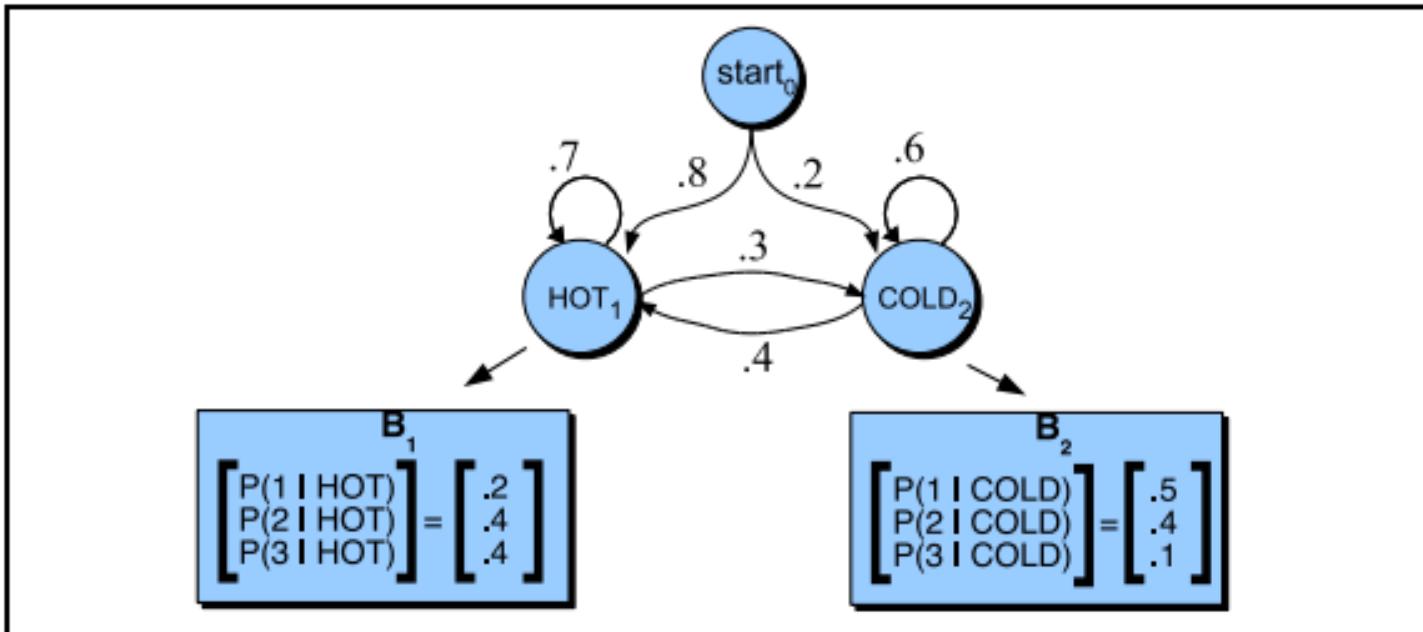


Figure 6.3 A Hidden Markov Model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables). Note that we have used a special zeroth start state; we could instead have represented the initial distribution over states by using the π vector, $\pi = [.8, .2]$. In general we will use the start state rather than the π vector in the remainder of this chapter.

Decoding Problem Example

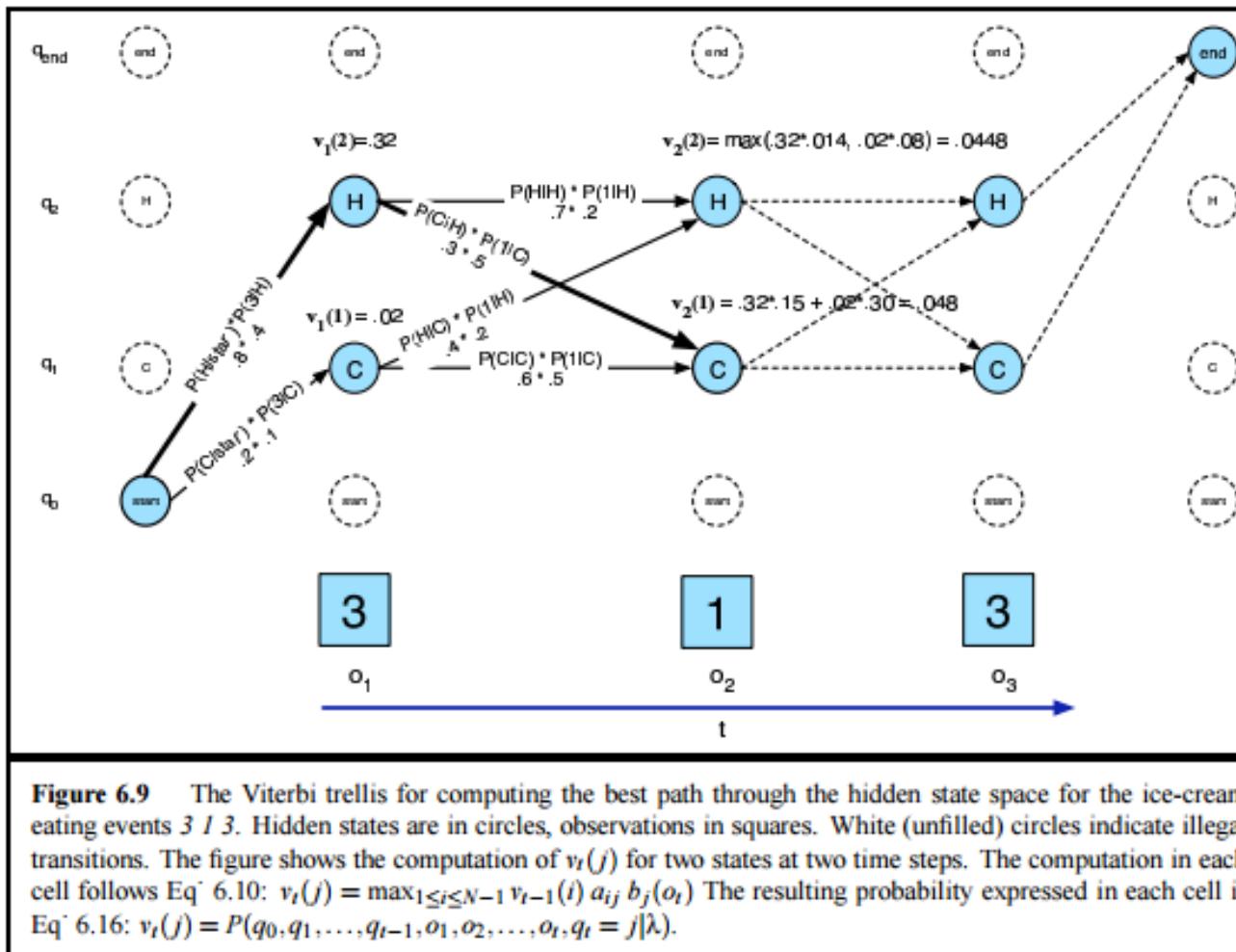


Figure 6.9 The Viterbi trellis for computing the best path through the hidden state space for the ice-cream eating events 3 1 3. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $v_t(j)$ for two states at two time steps. The computation in each cell follows Eq. 6.10: $v_t(j) = \max_{1 \leq i \leq N-1} v_{t-1}(i) a_{ij} b_j(o_t)$. The resulting probability expressed in each cell is Eq. 6.16: $v_t(j) = P(q_0, q_1, \dots, q_{t-1}, o_1, o_2, \dots, o_t, q_t = j | \lambda)$.

Back-Pointers

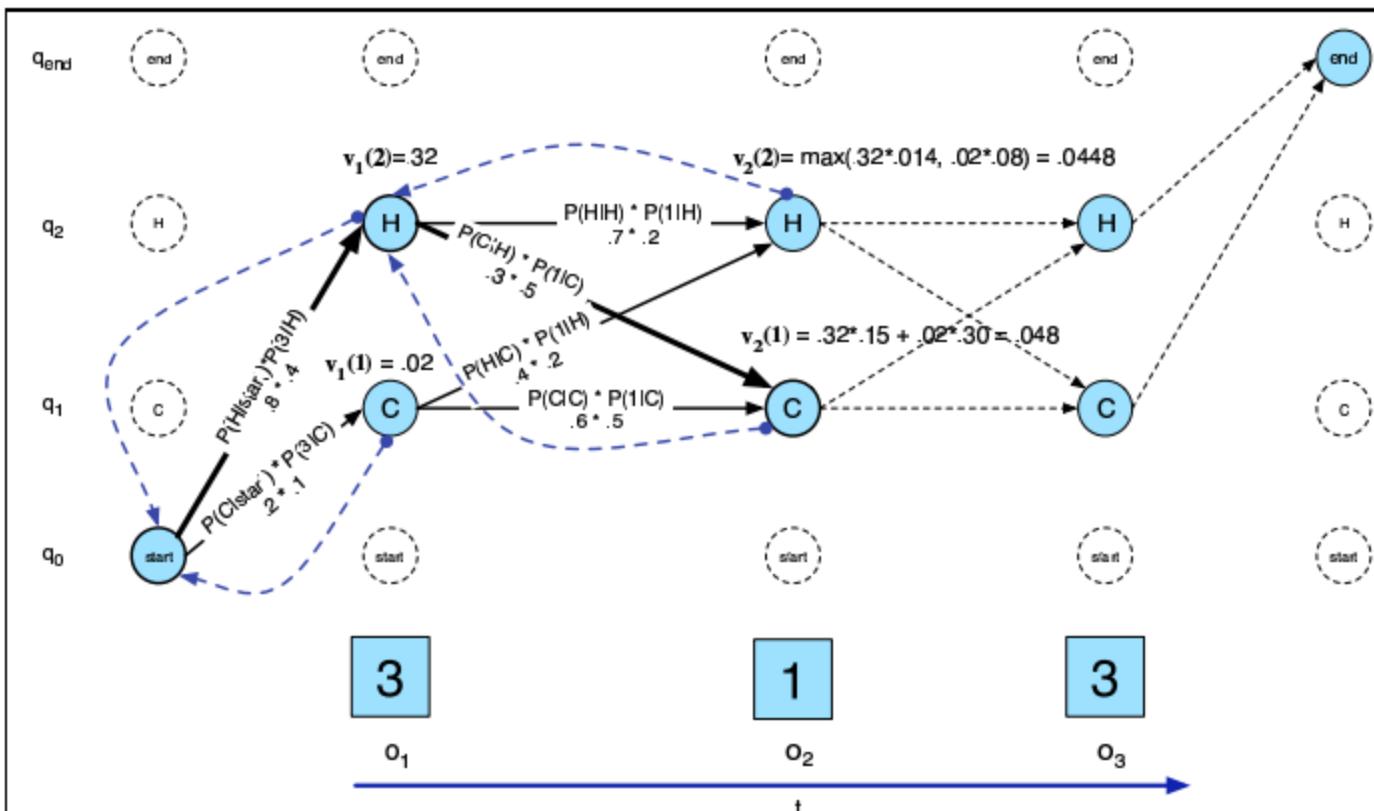


Figure 6.11 The Viterbi backtrace. As we extend each path to a new state account for the next observation, we keep a backpointer (shown with broken blue lines) to the best path that led us to this state.

HMMs as a Decoder for POS Tagging

Sentence:

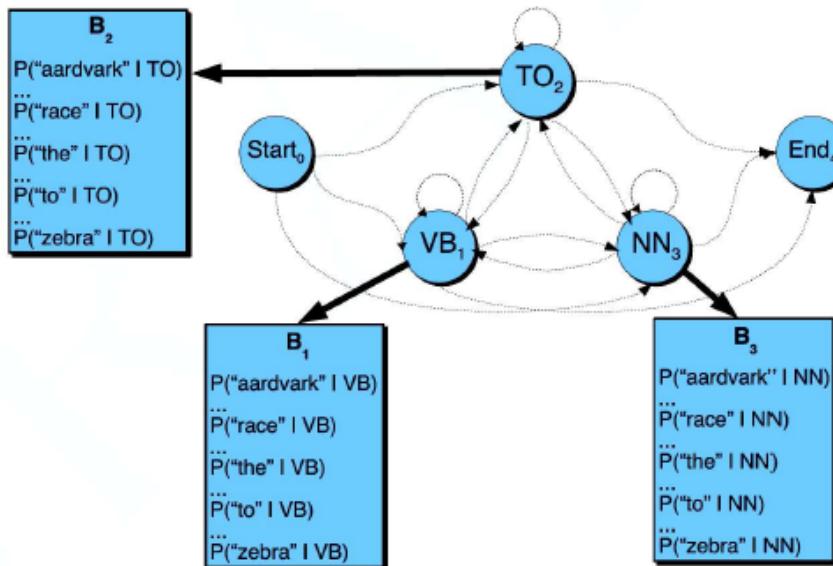
A cat caught the mouse

Decoded hidden states:

DT NN VBD DT NN

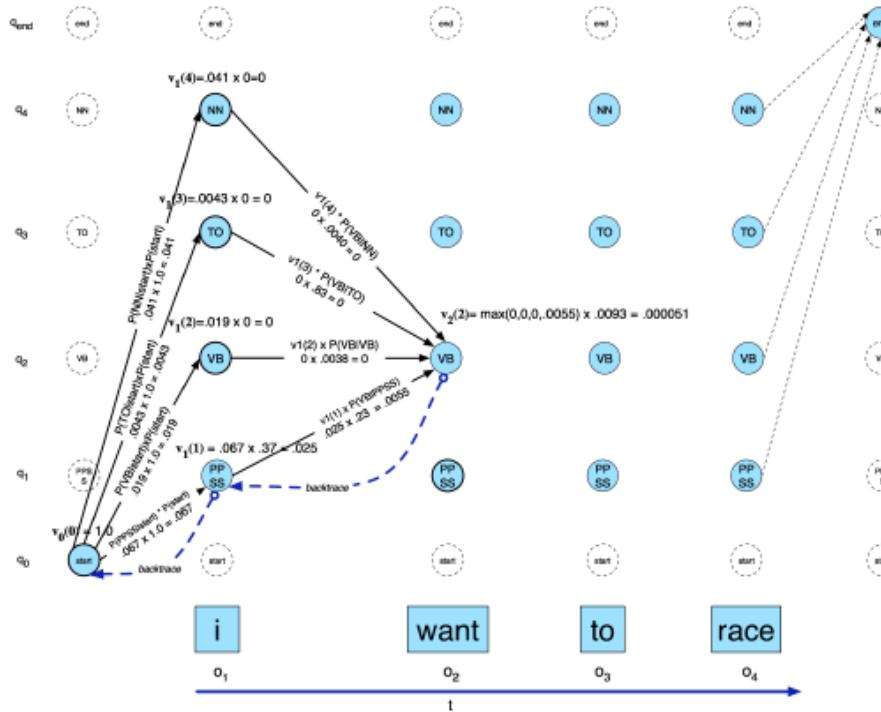
HMM transitions and emissions

Transitions and Emissions



TRELLIS

HMM Trellis



BASELINES

Overview: Accuracies

- Roadmap of (known / unknown) accuracies:
 - Most freq tag: ~90% / ~50%
 - Trigram HMM: ~95% / ~55%
 - Maxent $P(t|w)$: 93.7% / 82.6%
 - TnT (HMM++): 96.2% / 86.0%
 - MEMM tagger: 96.9% / 86.9%
 - Cyclic tagger: 97.2% / 89.0%
 - Upper bound: ~98%

Most errors
on unknown
words

Courtesy of Dan Klein

ERROR ANALYSIS

Common Errors

- Common errors [from Toutanova & Manning 00]

	JJ	NN	NNP	NNPS	RB	RP	IN	VB	VBD	VBN	VBP	Total
JJ	0	177	56	0	61	2	5	10	15	108	0	488
NN	244	0	103	0	12	1	1	29	5	6	19	525
NNP	107	106	0	132	5	0	7	5	1	2	0	427
NNPS	1	0	110	0	0	0	0	0	0	0	0	142
RB	72	21	7	0	0	16	138	1	0	0	0	295
RP	0	0	0	0	39	0	65	0	0	0	0	104
IN	11	0	1	0	169	103	0	1	0	0	0	323
VB	17	64	9	0	2	0	1	0	4	7	85	189
VBD	10	5	3	0	0	0	0	3	0	143	2	166
VBN	101	3	3	0	0	0	0	3	108	0	1	221
VBP	5	34	3	1	1	0	2	49	6	3	0	104
Total	626	536	348	144	317	122	279	102	140	269	108	3651

NN/JJ NN

official knowledge

VBD RP/IN DT NN

made up the story

RB VBD/VBN NNS

recently sold shares

Courtesy of Dan Klein

HMMs as a Decoder for NER

Sentence:

John bought a Maruti in Bangalore

Decoded hidden states:

Person Other Other Vehicle Other Place

HMM – Evaluation

Can you calculate $P(s)$ using an HMM trained as a POS tagger?

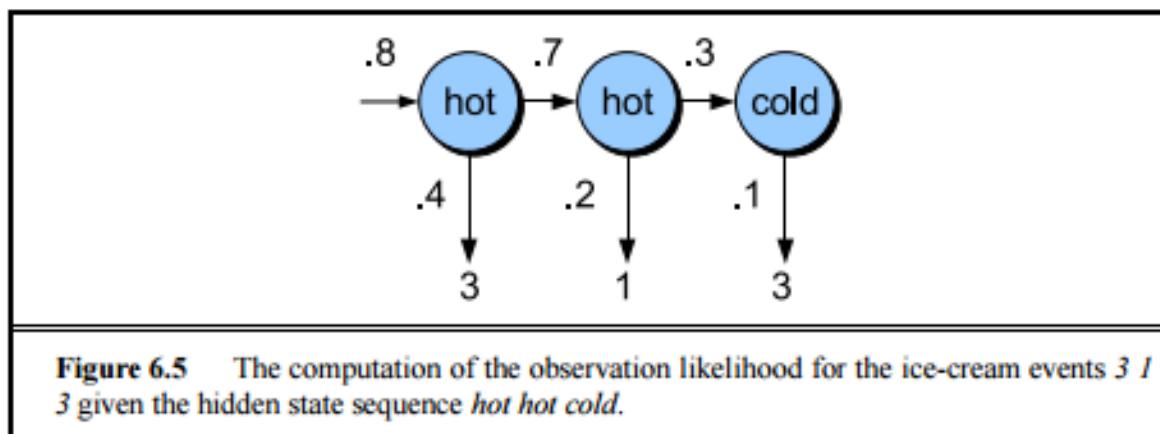
For evaluation (computing the probability of a sequence of observations) you need to use the ‘Forward algorithm’.

HMM – If you knew the hidden states

$$P(O|Q) = \prod_{i=1}^n P(o_i|q_i) \times \prod_{i=1}^n P(q_i|q_{i-1})$$

The computation of the forward probability for our ice-cream observation $3 \ 1 \ 3$ from one possible hidden state sequence *hot hot cold* is as follows (Fig. 6.5 shows a graphic representation of this):

$$\begin{aligned} P(3 \ 1 \ 3 | \text{hot hot cold}) &= P(\text{hot}|\text{start}) \times P(\text{hot}|\text{hot}) \times P(\text{cold}|\text{hot}) \\ &\quad \times P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold}) \end{aligned}$$



Courtesy the textbook Jurafsky and Martin

HMM – Forward Algorithm

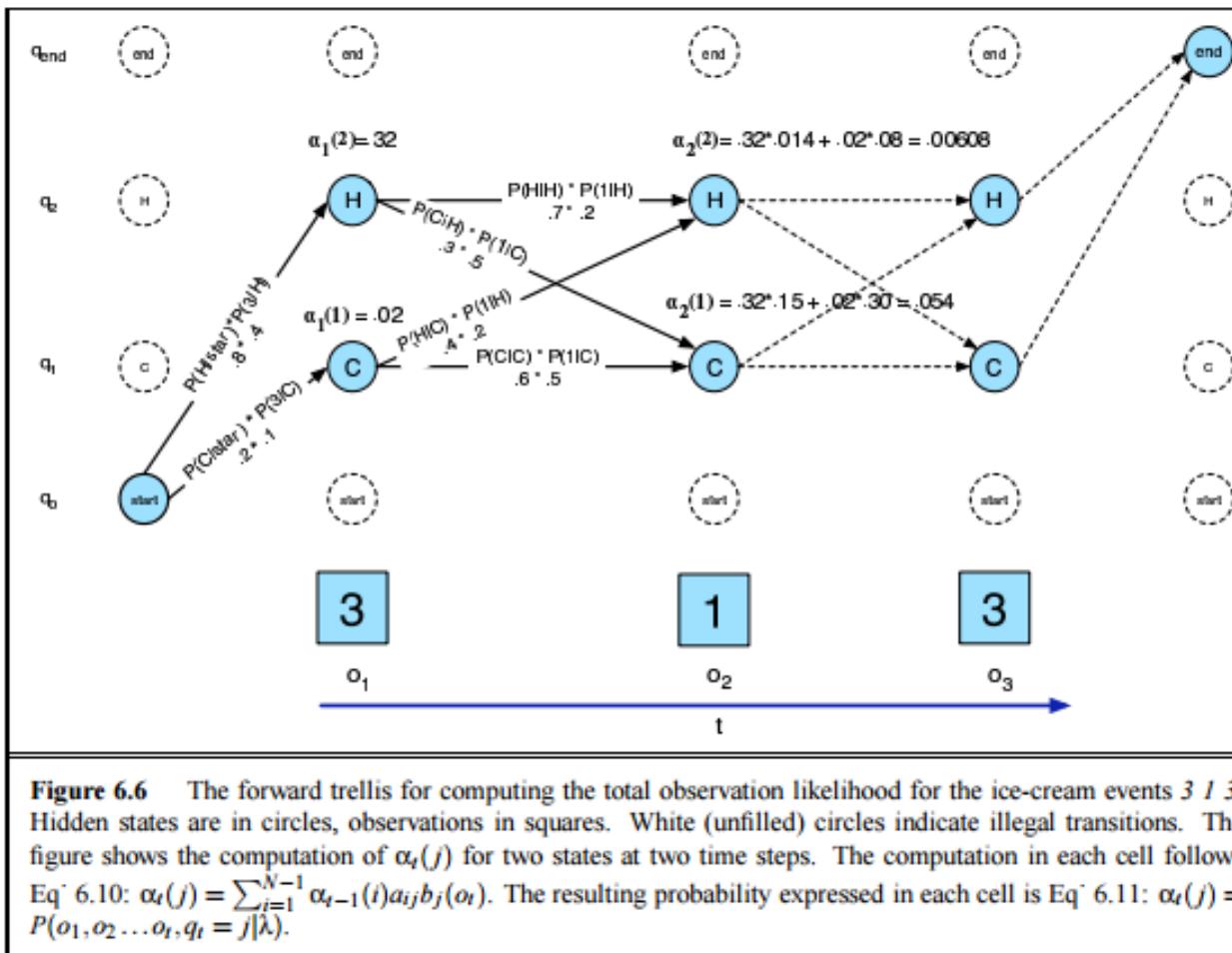


Figure 6.6 The forward trellis for computing the total observation likelihood for the ice-cream events 3 1 3. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $\alpha_t(j)$ for two states at two time steps. The computation in each cell follows Eq. 6.10: $\alpha_t(j) = \sum_{i=1}^{N-1} \alpha_{t-1}(i) a_{ij} b_j(o_t)$. The resulting probability expressed in each cell is Eq. 6.11: $\alpha_t(j) = P(o_1, o_2, \dots, o_t, q_t = j | \lambda)$.

Courtesy the textbook Jurafsky and Martin

HMM Model – Challenge

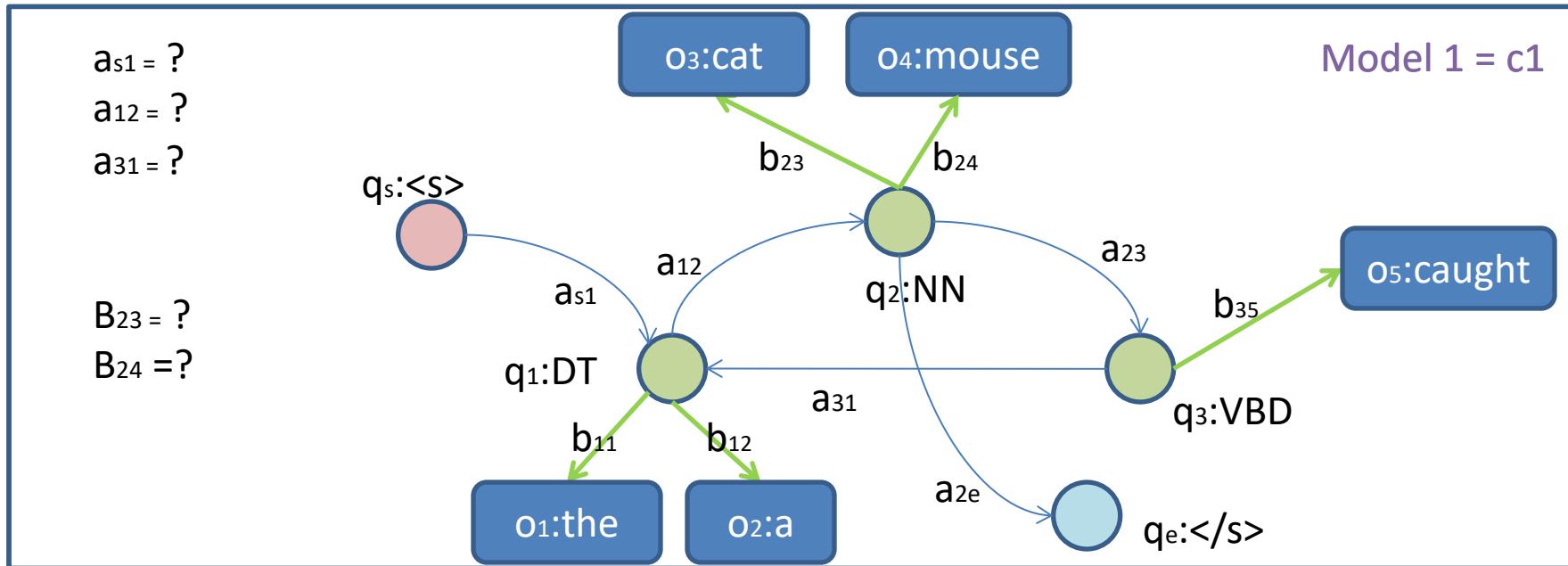
Can you build a classifier out of this?

You just calculated $P(s)$

**For classification you need to
compute $P(c|s)$**

Use Bayesian Inversion!!!

HMM – Challenge



Transition Probabilities: $A = \{a_{s1} \ a_{12} \ a_{23} \ a_{31} \dots\}$ Emission Probabilities: $B = \{b_{ij} \dots\}$

$$P(c|s) = P(s|c) * P(c) / Z$$

$$P(c|s) = \text{forward_algorithm}(s, \text{hmm_for_c}) * \\ P(c) / Z$$

HMM – Challenge

Can you build a relation extractor out of HMMs?

HMMs as a Classifier for RelEx

Train HMM1 on Sentences:

Tim/P Cook/P is the CEO of Apple/O

Vishal/P Sikka/P is the CEO of Infosys/O

Train HMM2 on Sentences:

Bangalore/L is the capital of Karnataka/L

Delhi/L is the capital of India/L

HMMs as a Classifier for RelEx

Now, given a new sentence:
“Frank is the CEO of Cognizant”

See if $P(C|S)$ is higher with HMM1 or HMM2

Ignoring $P(C)$, you can just see if $P(\text{"Frank is the CEO of Cognizant"})$ is higher for HMM1 or HMM2.

Sequential Pattern Mining for Text Analytics

Cohan Sujay Carlos

Aiaioo Labs

Bangalore, India

cohan@aiaioo.com

Feature Level Sentiment Analysis

- M. Hu and B. Liu, “Mining opinion features in customer reviews,” in *Proceedings of AAAI*, pp. 755–760, 2004.
- A.-M. Popescu and O. Etzioni, “Extracting product features and opinions from reviews,” in *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005.

Sentiment in Comparative Sentences

- N. Jindal and B. Liu, “Identifying comparative sentences in text documents,” in *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*, 2006.
- N. Jindal and B. Liu, “Mining comparative sentences and relations,” in Proceedings of AAAI, 2006.
- Murthy Ganapathibhotla and Bing Liu. "Mining Opinions in Comparative Sentences." *Proceedings of the 22nd International Conference on Computational Linguistics (Coling-2008)*, Manchester, 18-22 August, 2008.

Identifying intentions might help sentiment analysis?

Inquire Intention

Example 1: “Is the Canon EOS 5 a good camera?”

What clues do you have that this is an **inquiry**?

Identifying intentions might help sentiment analysis?

Inquire Intention

Example 1: “Is the Canon EOS 5 a good camera?”

What clues do you have that this is an **inquiry**?

Is the Canon EOS 5 a good camera?

Identifying intentions might help sentiment analysis?

Purchase Intention

Example 2: “I want a good Canon camera”

What are the clues that there is buy intention?

Identifying intentions might help sentiment analysis?

Purchase Intention

Example 2: “I want a good Canon camera”

What are the clues that there is buy intention?

I want a good Canon camera

Identifying intentions might help sentiment analysis?

Direct Intention

Example 3: “Take me to a good movie.”

What tell us that this is a directive?

Take me to a good movie

The kind of features we need

Is the Canon EOS 5 a good camera?

I want a good Canon camera

Take me to a good movie

Unigrams – ‘?’

Bigrams – “Is it” instead of “It is”

At the end of a sentence ... (How do you encode whether a word is at the end of a sentence?)

The kind of features we need

Is the Canon EOS 5 a good camera?

I want a good Canon camera

Take me to a good movie

Unigrams – ‘?’

Bigrams – “Is it” instead of “It is”

At the end of a sentence ... (How do you encode whether a word is at the end of a sentence?)

The kind of features we need

At the end of a sentence ... (How do you encode whether a word is at the end of a sentence?)

< s > Is the Canon EOS 5 a good camera? < /s >

< s > I want a good Canon camera < /s >

< s > Take me to a good movie < /s >

Class Sequential Rule

Sequence of symbols $i_1 - i_n$ that matches text in which the symbols appear in that order

Example, $I = < i_1 i_2 i_3 >$ where $i_1 i_2 i_3$ are unigrams

Matches:

A i_1 B i_2 C D i_3 E

i_1 i_2 R S i_3

i_1 i_2 i_3 M N

Does not match:

A B i_2 i_1 C D i_3 E

A B i_2 C D i_3 E

Class Sequential Rule

Placeholders $x_i - x_n$ among the symbols $i_1 - i_n$ that match text between the symbols on either side

Example, $I = < i_1 i_2 x_1 i_3 >$ where $i_1 i_2 i_3$ are unigrams

When I matches $A \ i_1 \ B \ i_2 \ C \ D \ i_3 \ E$ --- x_1 equals 'C D'

When I matches $i_1 \ i_2 \ R \ S \ i_3$ --- x_1 equals 'R S'

When I matches $i_1 \ i_2 \ i_3 \ M \ N$ --- x_1 equals "

Class Sequential Rules

Capabilities of the formalism:

- Identify **types** of sentences
- Extract **entities** from those sentences

Mapping to **Natural Language Programming**:

- **Types** = programming primitives
- **Entities** = variables, literals and expressions

Example

S = increment the value of x by 2 * 3

I₁ = <increment of VARIABLE by EXPRESSION>

I₁ matches S: increment the value of x by 2 * 3

VARIABLE = 'x'

EXPRESSION = '2 * 3'

Example Continued

$$E = 2 * 3 \xrightarrow{\hspace{10em}} (* 2 3)$$

$I_2 = < \text{EXPRESSION} * \text{EXPRESSION} >$

I_2 matches E : $2 * 3$

$\text{EXPRESSION} = '2'$

$\text{EXPRESSION} = '3'$

Example Continued

Mapping to **Natural Language Programming**:

- **Types** = programming primitives
- **Entities** = variables, literals and expressions

increment the value of x by 2 * 3

↓
Class Sequential Rules

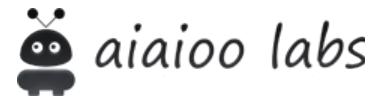
 $(\underline{+} = \underline{x} (\underline{*} \underline{2} \underline{3}))$

How do you learn CSRs?

Bangalore, India

Frequent Itemset Mining

Bangalore, India



Discovering Patterns for Relation Extraction



Corpus of inquiries:

- 1: Is the Canon 5 a good camera?
- 2: Is Pune close to Mumbai?
- 3: Is the moon made of cheese?

Find the frequent patterns in all these inquiries.

Corpus of inquiries:

- 1: Is the Canon 5 a good camera?**
- 2: Is Pune close to Mumbai?**
- 3: Is the moon made of cheese?**

Find the frequent patterns in all these inquiries.

Corpus of inquiries:

- 1: Is the Canon 5 a good camera?**
- 2: Is Pune close to Mumbai?**
- 3: Is the moon made of cheese?**

Find the frequent patterns in all these inquiries.

Is the ?

Is ?

Corpus of inquiries:

- 1: Is the Canon 5 a good camera?**
- 2: Is Pune close to Mumbai?**
- 3: Is the moon made of cheese?**

Frequent Patterns:

Is the ?

Is ?

Why are these the frequent patterns?

Support

Support is the number of times a pattern appears in a corpus:

- 1: Is the Canon 5 a good camera?**
- 2: Is Pune close to Mumbai?**
- 3: Is the moon made of cheese?**

Frequent Patterns are patterns whose support exceeds a threshold (minsup) = 2 in this case:

Is the ?

Is ?

Support for a pattern in this corpus:

- 1: Is the Canon 5 a good camera?
- 2: Is Pune close to Mumbai?
- 3: Is the moon made of cheese?

Is the ? (support = 2)

Is ? (support = 3)

Algorithm for extracting frequent patterns

1: Generate patterns

2: Throw out all patterns that do not have the required support

Algorithm for extracting frequent patterns

1: Generate patterns

But there are a lot of patterns

2: Throw out all patterns that do not have the required support

But this requires evaluating every single pattern by counting all the documents in the corpus that it matches, which is expensive.

Brute-force algorithm for extracting frequent patterns

- 1:** Generate all patterns of a given length
- 2:** Discard those with less support than minsup in the corpus

Patterns in this corpus:

- 1: Is the Canon 5 a good camera?**
- 2: Is Pune close to Mumbai?**
- 3: Is the moon made of cheese?**

- 1: Generate all patterns of length 1**

Is, the, Canon, 5, a, good, camera, ?, Pune, close,
to, Mumbai, moon, made, of, cheese

Patterns in this corpus:

1: Is the Canon 5 a good camera?

2: Is Pune close to Mumbai?

3: Is the moon made of cheese?

2: Generate all patterns of length 2 from the patterns of length 1

<Is, the>, <is, Canon>, <is,5>, <is, a> ... <the, Canon>, <the, 5> ... <Canon, 5>, <Canon, a>

Patterns in this corpus:

- 1: Is the Canon 5 a good camera?**
 - 2: Is Pune close to Mumbai?**
 - 3: Is the moon made of cheese?**
-
- 3: Generate all patterns of length 3 from the patterns of length 2**

<Is, the, Canon>, <Is, the, 5>, <Is, the, a>, <is, Canon, 5>, <is, Canon, a> <is, 5, a>

Patterns in this corpus:

- 1: Is the Canon 5 a good camera?
- 2: Is Pune close to Mumbai?
- 3: Is the moon made of cheese?
- 4: Generate all patterns of length 4 from the patterns of length 3

<Is, the, Canon, 5>, <Is, the, Canon, a>, <Is, the, Canon, the>, <the, Canon, 5, a>, <the, Canon, a, good>

Patterns in this corpus:

- 1: Is the Canon 5 a good camera?**
- 2: Is Pune close to Mumbai?**
- 3: Is the moon made of cheese?**

So if you have 10000 possible patterns and a corpus of 10000 documents, you end up having to do 100 million matches in order to calculate support and select only the frequent patterns whose counts exceed the minsup value.

Patterns in this corpus:

- 1: Is the Canon 5 a good camera?**
- 2: Is Pune close to Mumbai?**
- 3: Is the moon made of cheese?**

So you use a variant of the A-priori algorithm which was originally developed for frequent itemset mining.

Retail transactions:

- 1: <milk, eggs, bread, sugar>
- 2: <coffee, milk, eggs, magazine>
- 3: <milk, bread, eggs>

Frequent itemsets:

milk, eggs, bread

milk, eggs

How do you find the Frequent Itemsets

Retail transactions:

- 1: <milk, eggs, bread, sugar>
- 2: <coffee, milk, eggs, magazine>
- 3: <milk, bread, eggs>

Generate all itemsets

Throw out all itemsets that don't have the minsup (minimum support) in the corpus.

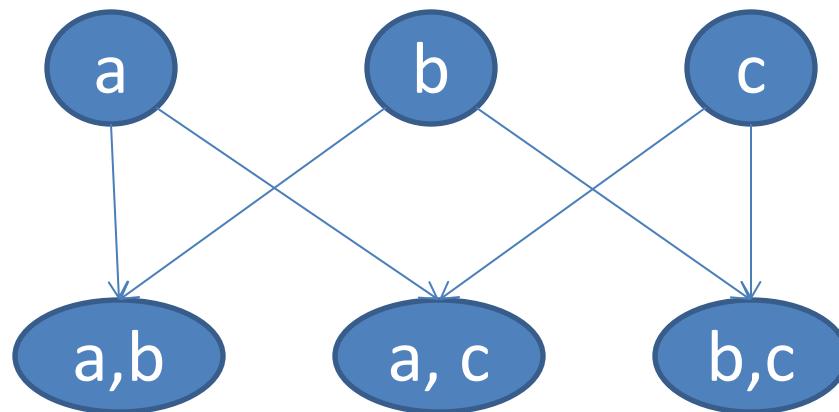
Generating the Frequent Itemsets

Generate itemsets of size 1



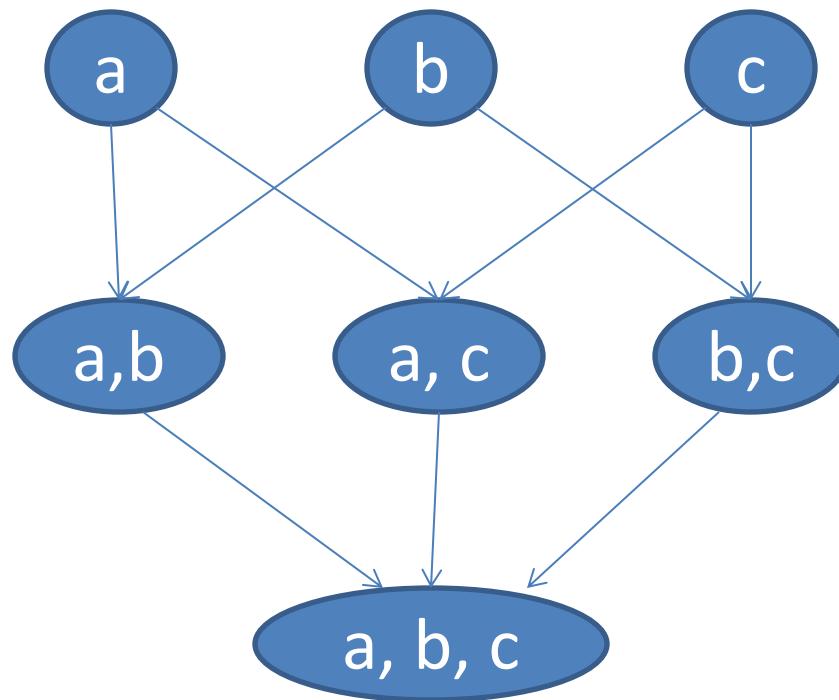
Generating the Frequent Itemsets

Generate itemsets of size 2



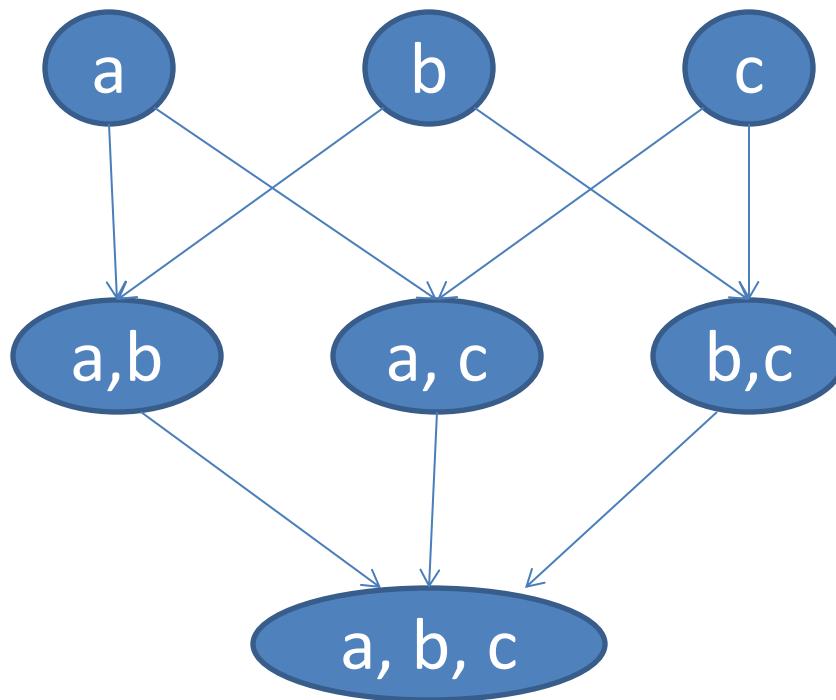
Generating the Frequent Itemsets

Generate itemsets of size 3



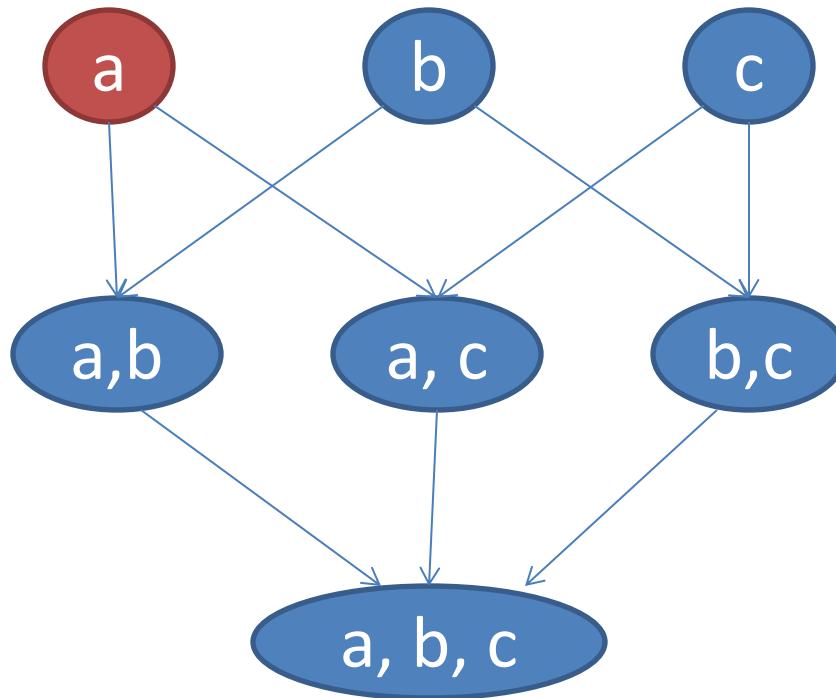
Generating the Frequent Itemsets

You can consider a smaller set of candidates if you prune this graph



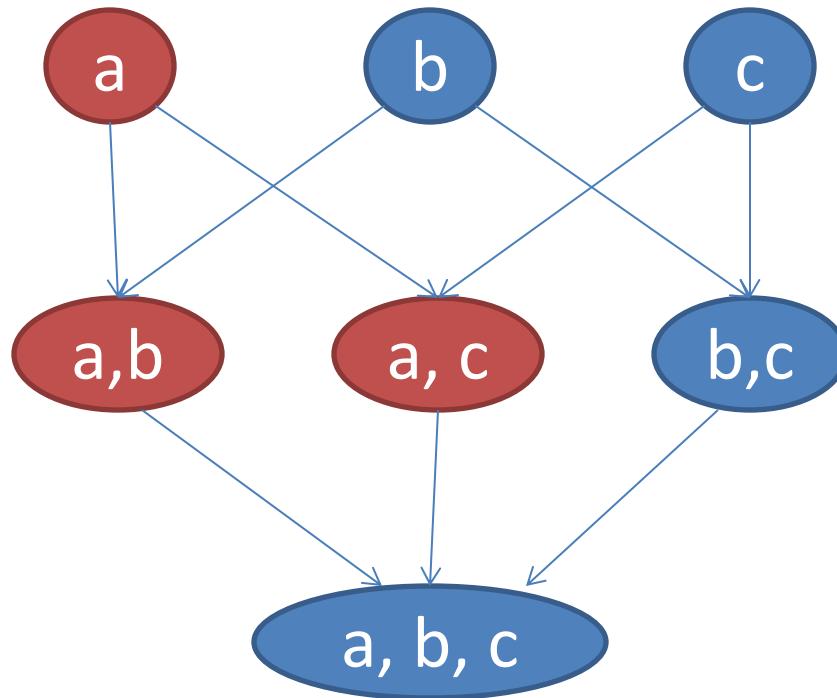
Generating the Frequent Itemsets

Let's say “a” does not have the requisite support.



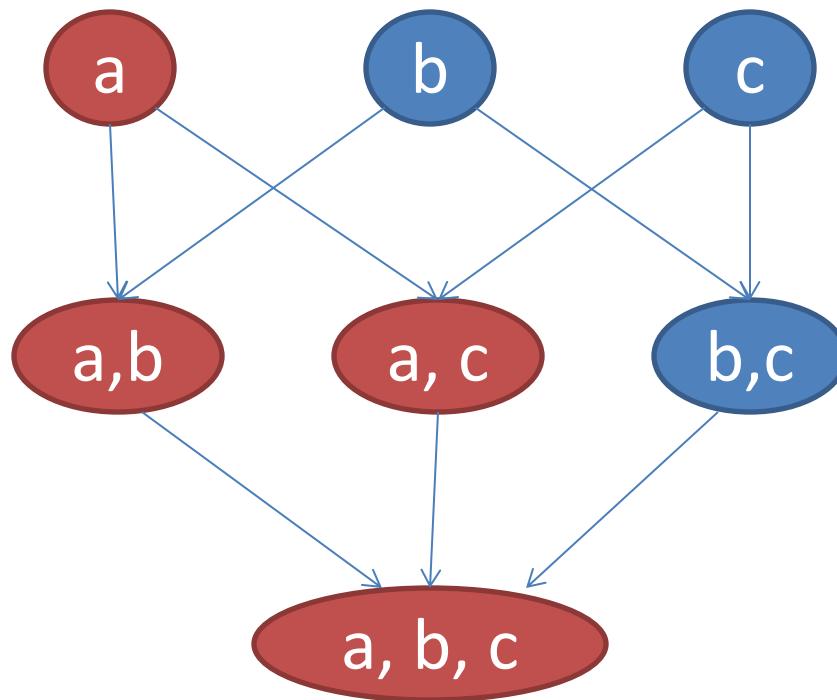
Generating the Frequent Itemsets

Then all length 2 itemsets containing “a” also do not have the requisite support.



Generating the Frequent Itemsets

All length 3 itemsets containing “a” also do not have the requisite support.



The Frequent Itemsets

Now we have the frequent itemsets.



The Frequent Itemsets

These can be used to generate the rule:

Now test the confidence of the rule (the % of the time the rule holds in the corpus) to decide if the rule is a good one.

$$b \rightarrow c$$

The Frequent Itemsets

You can also generate:

Age > 20 and Gender Male -> Online Buyer

b -> c

Common Pattern Specification Language

Cohan Sujay Carlos

Aiaioo Labs

Bangalore, India

cohan@aiaioo.com

Rule-Based Information Extraction

- THE COMMON PATTERN SPECIFICATION LANGUAGE - Douglas E. Appelt

What is CPSL?

Designed as a language for specifying **finite-state grammars** for specifying **information extraction rules**

Why CPSL?

Meant to “enable the creation of **shareable resources** for the development of **rule-based information extraction systems**”

Where CPSL?

Designed as a language for “systems based on the **knowledge engineering** approach of developing sets of rules more or less by hand, targeted toward specific subjects.”

Who designed CPSL?

The CPSL was designed by a committee consisting of a number of researchers from the Government and all of the TIPSTER research sites involved in Information Extraction.

How CPSL?

Assume a reference interpreter ...



It has a reference pointer that starts out pointed at the first word in a document then runs through the document as matching is done, and ends at the last word.

How CPSL?

Assume a reference interpreter ...



It has a reference pointer that starts out pointed at the first word in a document then runs through the document as matching is done, and ends at the last word.

How CPSL?

Assume a reference interpreter ...



It has a reference pointer that starts out pointed at the first word in a document then runs through the document as matching is done, and ends at the last word.

How CPSL?

Assume a reference interpreter ...



Rule: <rule_name>

Priority: <integer>

<rule_pattern part> --> <rule_action_part>

How CPSL?

Assume a reference interpreter has been implemented by the ...

Rule: RULE 1

Priority: 1

**< prefix_pattern > body_pattern < postfix_pattern > -->
<rule_action_part>**

When pattern matching begins, the reference interpreter assumes that the initial cursor position is between the prefix pattern and the body pattern.



How CPSL?

Assume a reference interpreter has been implemented by the ...

Rule: RULE 1

Priority: 1

< prefix_pattern > body_pattern < postfix_pattern > -->
<rule_action_part>

If the annotations to the immediate left of the cursor match the prefix pattern, then the body pattern is matched.



How CPSL?

Assume a reference interpreter has been implemented by the ...

Rule: RULE 1

Priority: 1

<prefix_pattern> body_pattern <postfix_pattern>
 > --> <rule_action_part>

If that match is successful, then the postfix pattern is matched.

Example of CPSL?

Assume a reference interpreter has been implemented by the ...

Rule: RULE 1

Priority: 1

<a> (reference | interpreter)+ <by> -->
<rule_action_part>

will match “reference interpreter has been implemented”.

Pattern Combination?

Alternation: (arg1 | arg2 | "" | arg n)

Iteration: (arg1 arg2 ... arg n) *

or (arg1 arg2 ... arg n) +

Optionality: (arg1 arg2 ... arg n)?

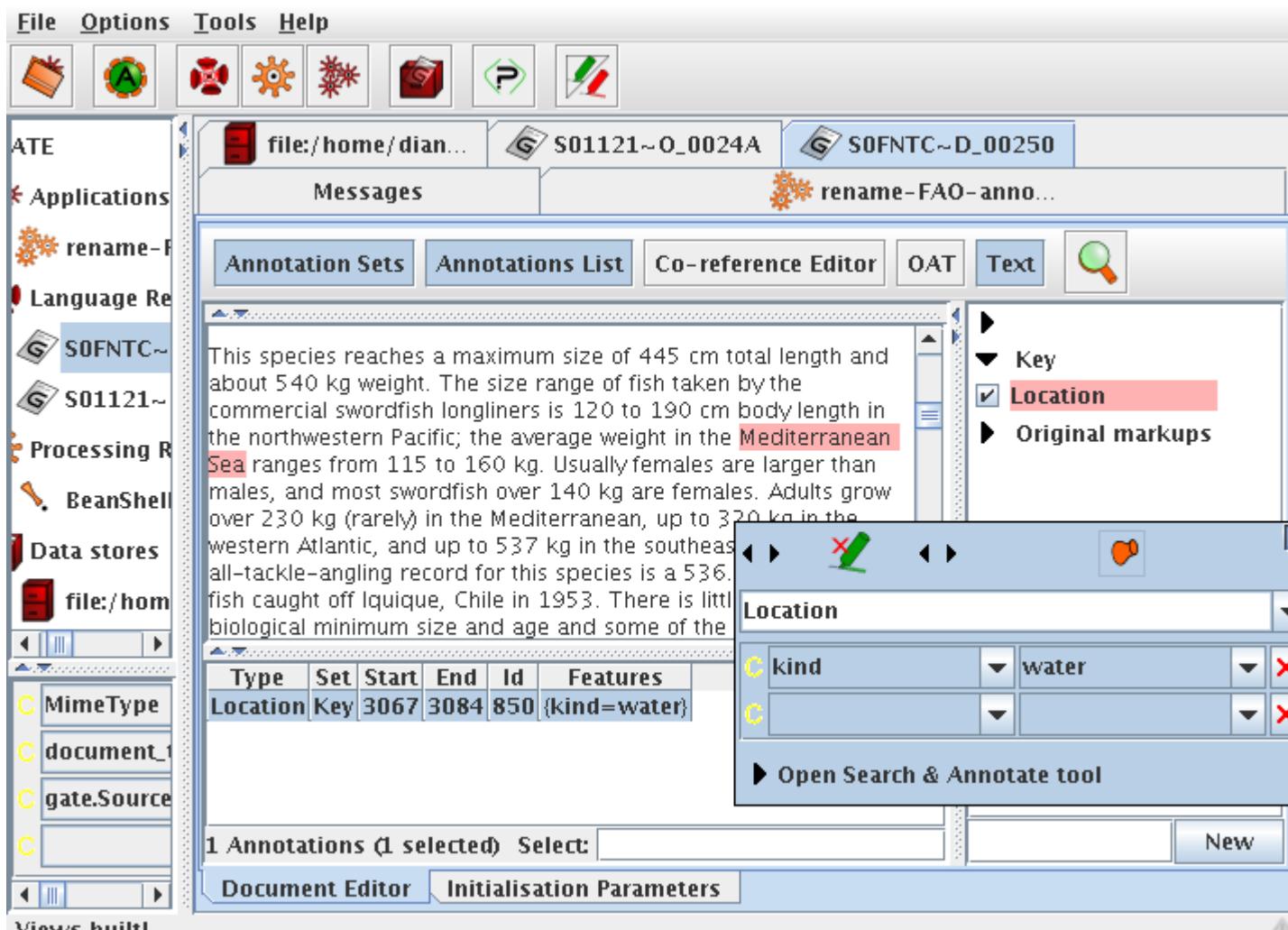
The rule action part

```
<a> (reference | interpreter)+ <by> -->  
<rule_action_part>
```

CPSL includes two assignment operators: "=" and "+=". The former operator is the basic assignment operator. The latter operator assumes that the left hand operand represents a set, and the right hand element is added to the set by the assignment.

A popular CPSL implementation

GATE = General Architecture for Text Engineering



Rule-Based Sentiment Classification

matches_gazetteer_of_positive_words

--> **label = positive**

matches_gazetteer_of_negative_words

--> **label = negative**

IF count(positive labels) > count(negative labels)
THEN this document has positive sentiment

Rule-Based Sentiment Classification

matches_gazetteer_of_positive_words

--> **label = positive**

matches_gazetteer_of_negative_words

--> **label = negative**

But if your rules are going to be this simple,
why do you even need GATE?

Advantages of Rule-Based Approaches

- It often takes less time to build rules than to collect data for training a good Machine Learning system
- Rules sometimes work better than machine learning on short texts because short texts don't allow the accurate estimation of $P(*)$
- You get instant results with rules
- Writing rules is more fun for engineers than annotating data

What to Watch Out for When Using Rule-Based Approaches

- Since a lot of time is spent building rules, you often end up collecting very little data for testing the rule-based system. This can result in over-fitted rules – rules that only work for the data in your test set.
- There is a tendency to let the rules pick the examples to test on
- Development bias – you use the test corpus to guide rule-writing, and as a result you end up tuning the rules to the test corpus

How to avoid development bias

- Break your test set up into a development set and a test set
- Lock up your test set till you have developed all the rules
- Use the development set to develop your rules
- Then, measure the performance on the test set.
- Don't use the test set more than 3 times. After three measurements, make it a part of the development set and collect a new test set.

Exercise 1

Build a rule-based sentiment classifier based on the AFINN gazetteer.

Measure its performance on the polarity dataset.

The datasets were downloaded from <http://www.cs.cornell.edu/people/pabo/movie-review-data/>

The MPQA corpus seems to be more GATE-oriented: <http://mpqa.cs.pitt.edu/>

Exercise 2

Build a rule-based sentiment classifier based on the AFINN gazetteer.

Measure its performance on the subjectivity dataset.

Exercise 3

Build a naïve bayesian sentiment classifier trained on 80% of the polarity dataset.

Measure its performance on the remaining 20% of the polarity dataset.

Exercise 4

Build a naïve bayesian sentiment classifier trained on 80% of the subjectivity dataset.

Measure its performance on the remaining 20% of the subjectivity dataset.

Rule-based vs Statistical Approaches

As children, we are taught this spelling rule:

“I before E except after C”

-- an example of a linguistic insight

[Courtesy of http://norvig.com/chomsky.html](http://norvig.com/chomsky.html)

Rule-based / Statistical Approaches

Learning to choose between rule-based
and statistical approaches

[Courtesy of http://norvig.com/chomsky.html](http://norvig.com/chomsky.html)

Problem 7: Deciding Between Spelling Alternatives

1. **F**ield
2. **W**ield
3. **Sh**ield
4. De**c**eive
5. Rec**e**ive
6. Ce**il**ing

[Courtesy of http://norvig.com/chomsky.html](http://norvig.com/chomsky.html)

A question

Why use data-oriented methods when we can ask an expert to come up with some rules?

Rule-based Approach

As children, we are taught this spelling rule:

“I before E except after C”

-- an example of a linguistic insight

[Courtesy of http://norvig.com/chomsky.html](http://norvig.com/chomsky.html)

Probabilistic Statistical Model:

- Count the occurrences of ‘ie’ and ‘ei’ and ‘cie’ and ‘cei’ in a large **corpus**

$$P(IE) = 0.0177$$

$$P(EI) = 0.0046$$

$$P(CIE) = 0.0014$$

$$P(CEI) = 0.0005$$

The Saying:
“I before E except after C”

Courtesy of <http://norvig.com/chomsky.html>

Probabilistic Statistical Model:

- Count the occurrences of ‘ie’ and ‘ei’ and ‘cie’ and ‘cei’ in a large **corpus**

$$P(IE) = 0.0177$$

$$P(EI) = 0.0046$$

$$P(CIE) = 0.0014$$

$$P(CEI) = 0.0005$$

The Saying:
“I before E except after C”
I normally occurs before E!!
So the first part of the saying is true!

Probabilistic Statistical Model:

- Count the occurrences of ‘ie’ and ‘ei’ and ‘cie’ and ‘cei’ in a large **corpus**

$$P(IE) = 0.0177$$

$$P(EI) = 0.0046$$

$$P(CIE) = 0.0014$$

$$P(CEI) = 0.0005$$

The Saying:

“I before E except after C”

After C, I still comes before E
75% of the time!! So the
second part is incorrect!

Words where ie occur after c

- science
- society
- ancient
- species

[Courtesy of http://norvig.com/chomsky.html](http://norvig.com/chomsky.html)

Statistical Approach

Language is **more complicated** than humans realize ...

People underestimate the complexity of the rules governing even simple things related to language ...

So, we use probabilities and statistics to aid in decision making about text ... instead of rules like “I before E except after C”

A question

If data-oriented methods can be so much better, why do people still use rule-based methods?

It turns out ... finding labelled data is hard

Can you use otherwise labelled data?

Distance Learning for Sentiment Classification

- **Twitter Sentiment Classification using Distant Supervision** by: [Alec Go](#), [Richa Bhayani](#), [Lei Huang](#)
- Dmitry Davidiv, Oren Tsur and Ari Rappoport. 2010. Enhanced Sentiment Learning Using Twitter Hashtags and Smileys. Coling 2010.

Distance Learning for Sentiment Classification

- **Twitter Sentiment Classification using Distant Supervision** by: [Alec Go](#), [Richa Bhayani](#), [Lei Huang](#)

Say there is a tweet:

“Havng yummy foodz @ café XYZ ☺”

Use the smileys ☺ as sentiment labels

But finding even that sort of data is hard

Can you use unlabelled data?

Machine Learning on Unlabelled Data

Now, we'll learn about learning on
unlabelled data.

It's like learning without a teacher &
figuring stuff out!

Unsupervised Sentiment Classification

- P. Turney, “Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews,” in *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 417–424, 2002.

Problem 8: Can you train a classifier on unlabelled data

Can unlabelled data help?

Training a classifier on partly labelled data

Text Classification from Labeled and Unlabeled
Documents using EM

by KAMAL NIGAM, ANDREW KACHITES, SEBASTIAN
THRUN, TOM MITCHELL

Training a Classifier

How do you train a classifier on very little labelled data?

Can you train a classifier on partly labelled data?

Can you change the precision and recall?

Problem 8: Training a classifier using EM

**Train classifier 1 with a few labelled documents
labelled with each of N categories.**

**Classify a large number of unlabelled documents
using classifier 1.**

**Train classifier 2 using the all the documents
classified by the previous classifier (1) and the
original manually labelled documents.**

**Classify the large number of originally unlabelled
documents using classifier 2.**

Repeat K times ...

Classifier K will be better than classifier 1.

Training on Partly Labelled Data

We have our third design pattern:

Training on Partly Labelled Data

When do you use it?

When there is a lot of unlabelled data available and very little labelled data.

Time for Exercises!

Let's practise training a classifier with partly labelled data!

Problem 9: Can you train a classifier to reject a category?

Can partly labelled data help?

Problem 9: Training a classifier to Reject a Category

Train classifier 1 with a few documents labelled with the category to be rejected R + the category Other comprising all other labelled documents.

Classify a large number of unlabelled documents using classifier 1.

Train classifier 2 using the all the documents classified by the previous classifier (1) as R and the remaining labelled documents marked as other.

Classify the large number of originally unlabelled documents using classifier 2.

Repeat K times ...

Classifier K will be better than classifier 1.

Training to Reject a Category

We have our fourth design pattern:

Training to Reject

When do you use it?

When you need to change the precision and recall of a classifier.

Problem 10: Clustering

What is Clustering?

Grouping data without labels in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters)

Data With Labels

Data with corresponding **Human Scores**

- "...is writing a paper"
- "... has flu 😞"
- "... is happy, yankees won!"

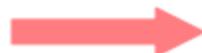


0.5
0.1
0.87

Regression

Data with corresponding **Human Class Labels**

- "...is writing a paper"
- "... has flu 😞"
- "... is happy, yankees won!"



SAD
SAD
HAPPY

Perceptron

Naïve Bayes

Fisher's Linear Discriminant

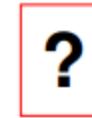
Data Without Labels

Data with NO corresponding Labels

- "...is writing a paper"
- "... has flu ☹"
- "... is happy, yankees won!"



$$\begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix}$$



Document Clustering

Previously we classified Documents into:
Politics(Class1) and Sports (Class2)

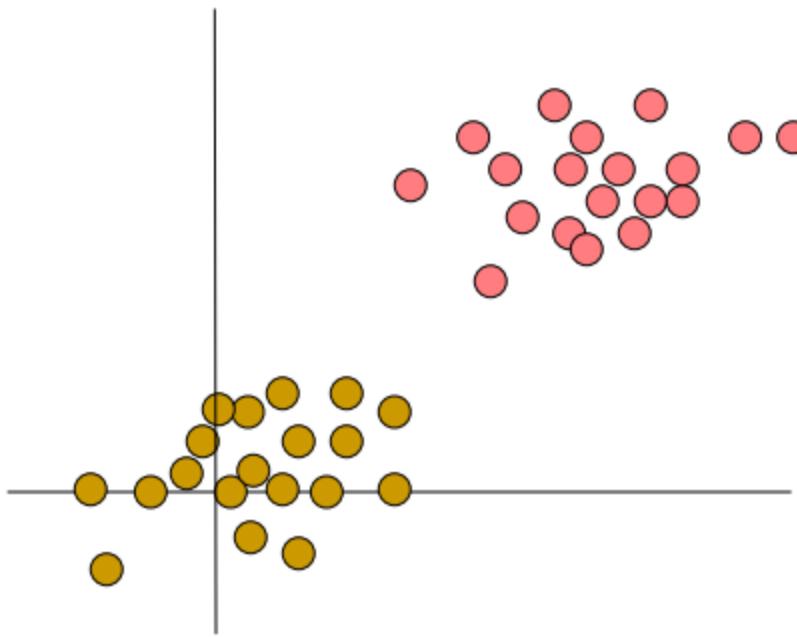
We had labelled data

- Supervised learning (classification)

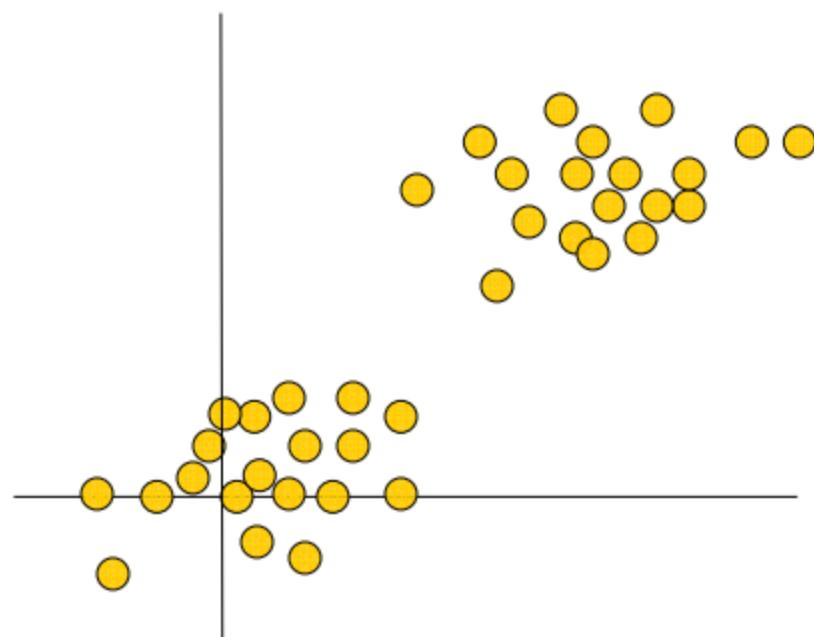
If we have no labelled data

- Unsupervised learning (clustering)

Document Clustering



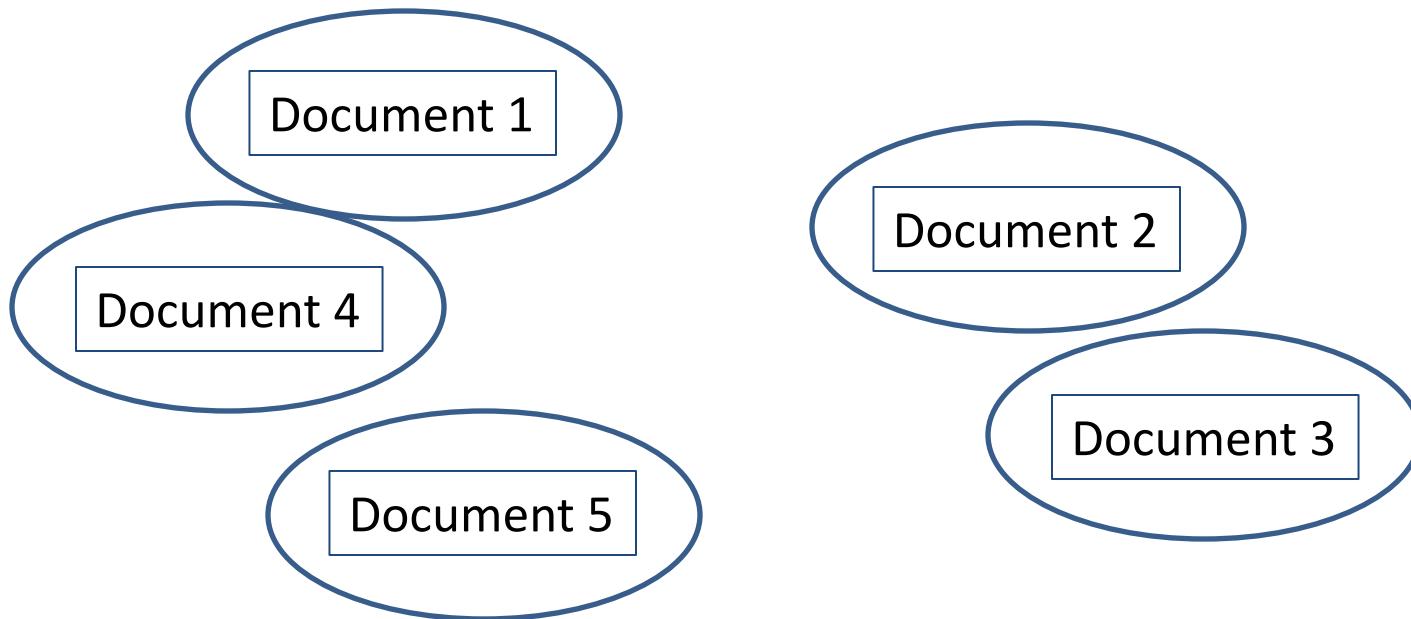
Supervised Training
of Classification Algorithm



Unsupervised Training
of Clustering Algorithm

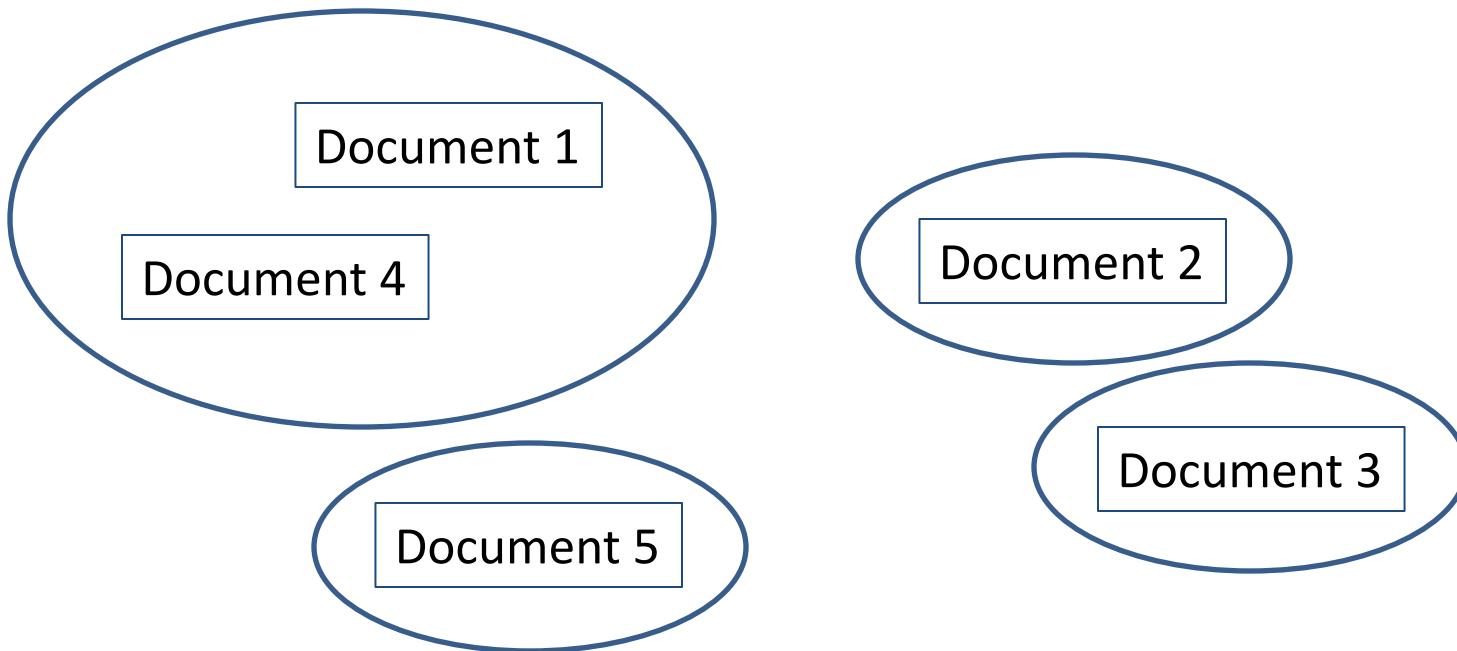
Method 1: Agglomerative Hierarchical Clustering

- Start with each document in its own cluster



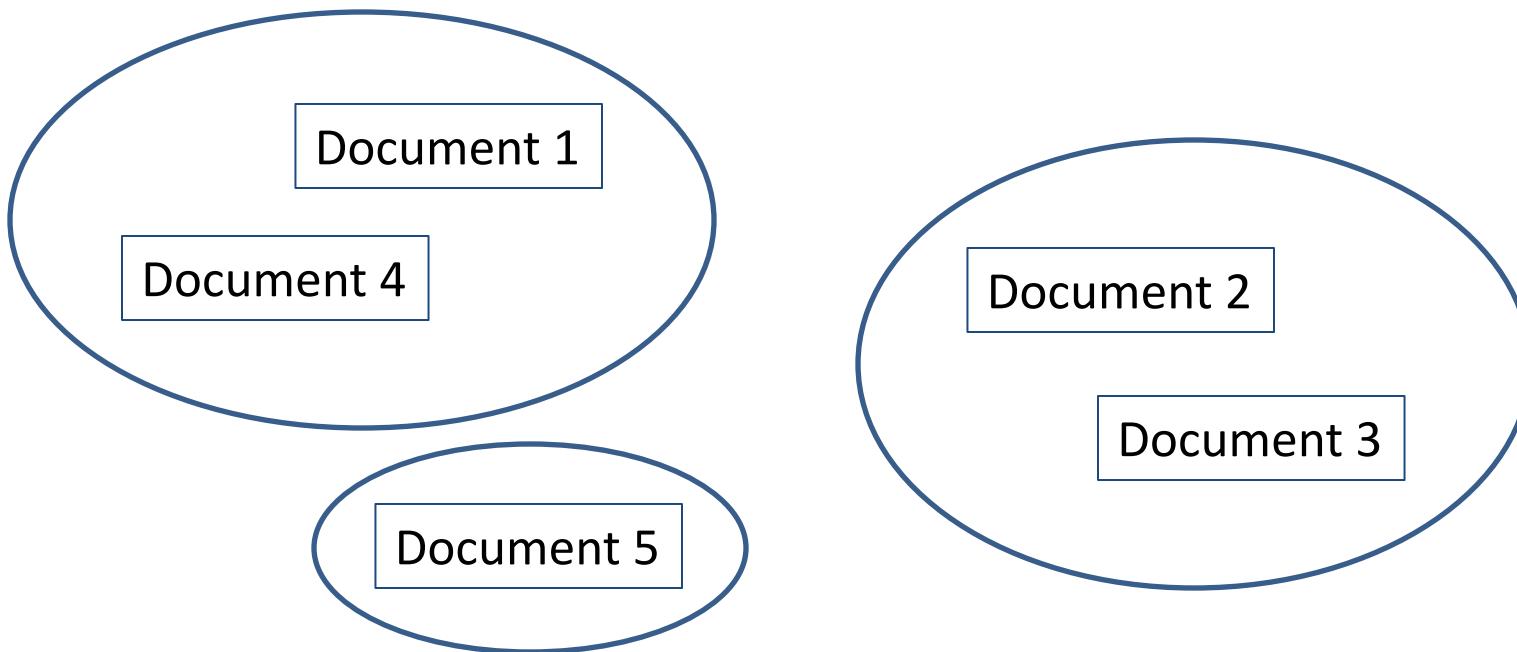
Method 1: Agglomerative Hierarchical Clustering

- Calculate the distance between all pairs of clusters and merge the nearest ones



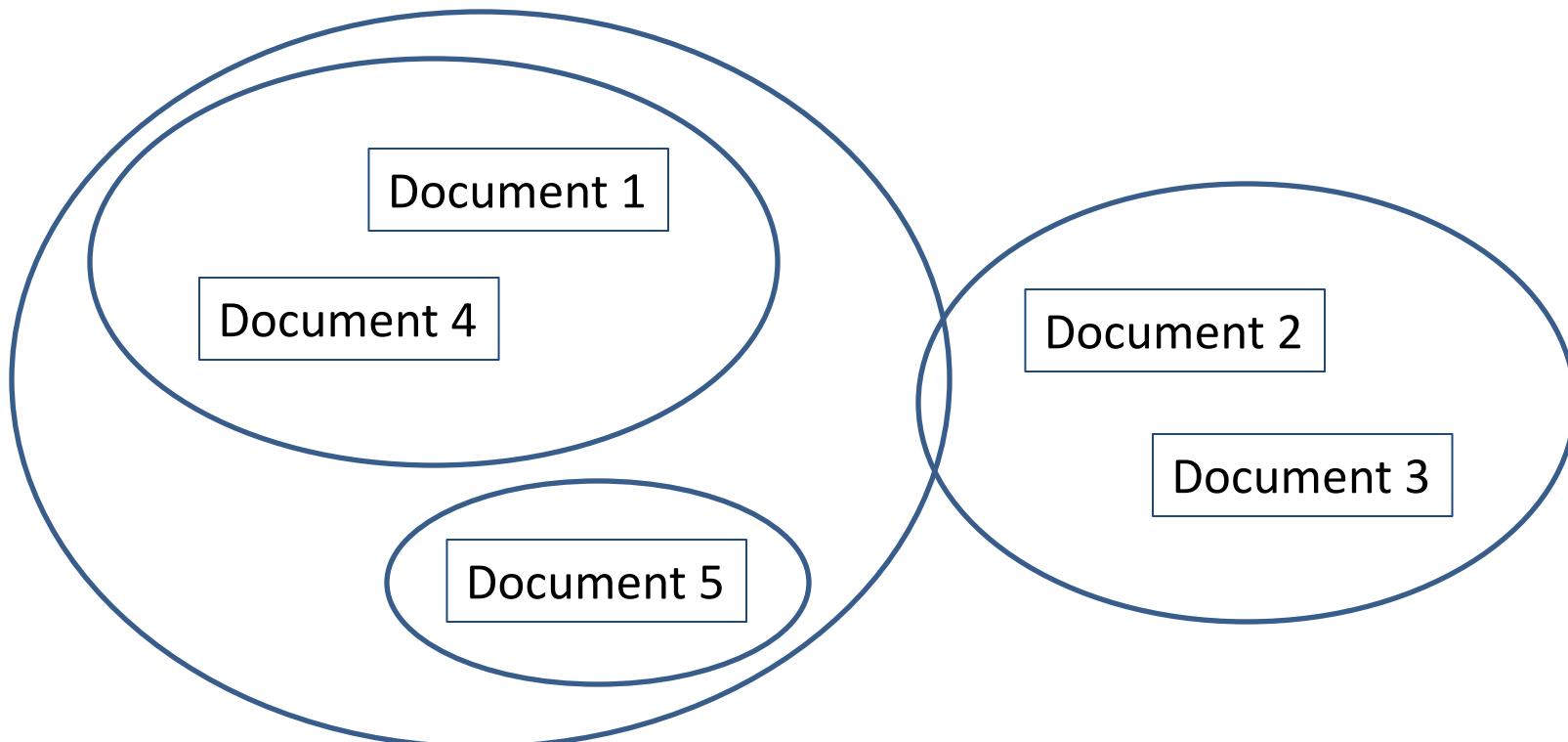
Method 1: Agglomerative Hierarchical Clustering

- Calculate the distance between the new cluster and all other clusters and merge the nearest pair



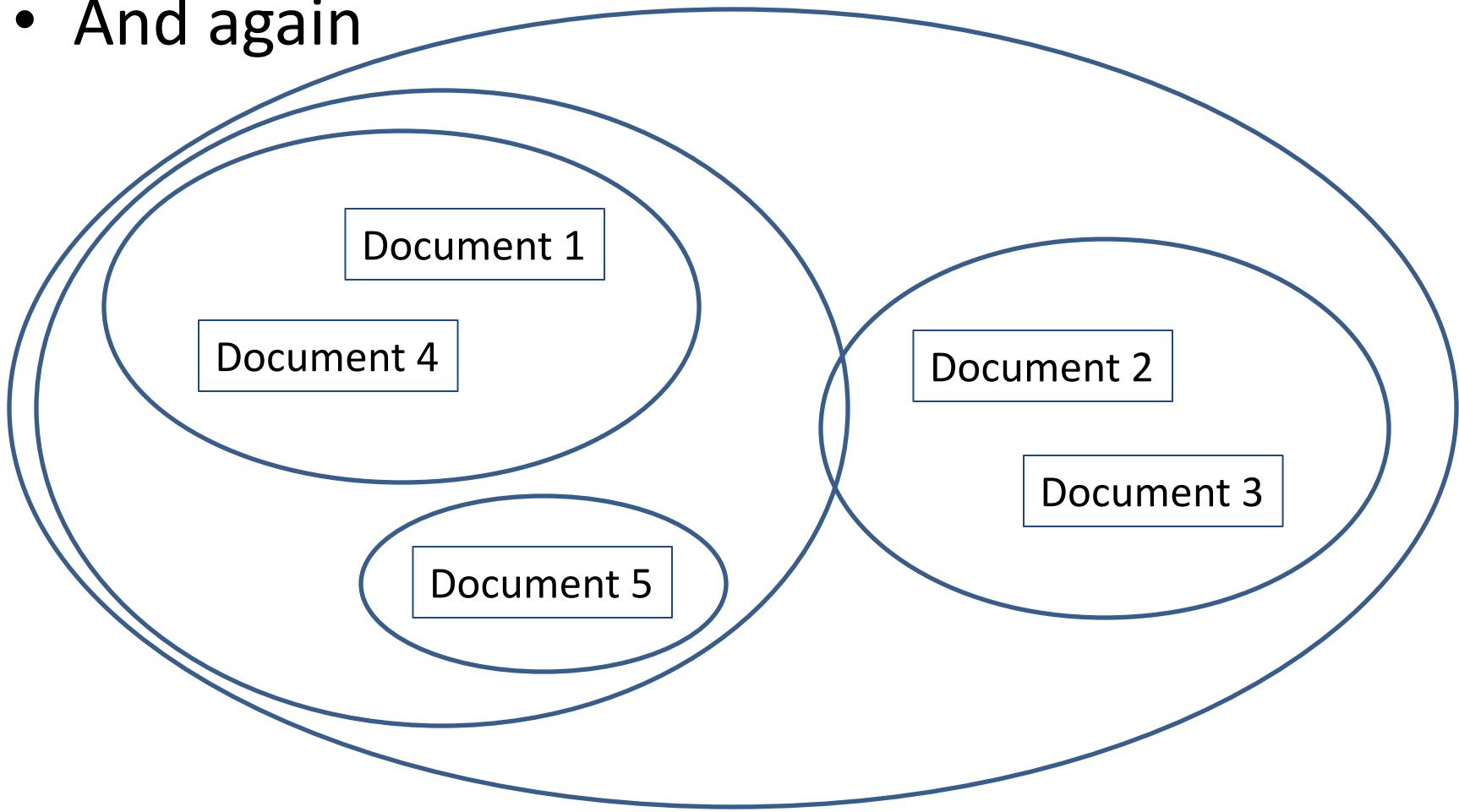
Method 1: Agglomerative Hierarchical Clustering

- Again



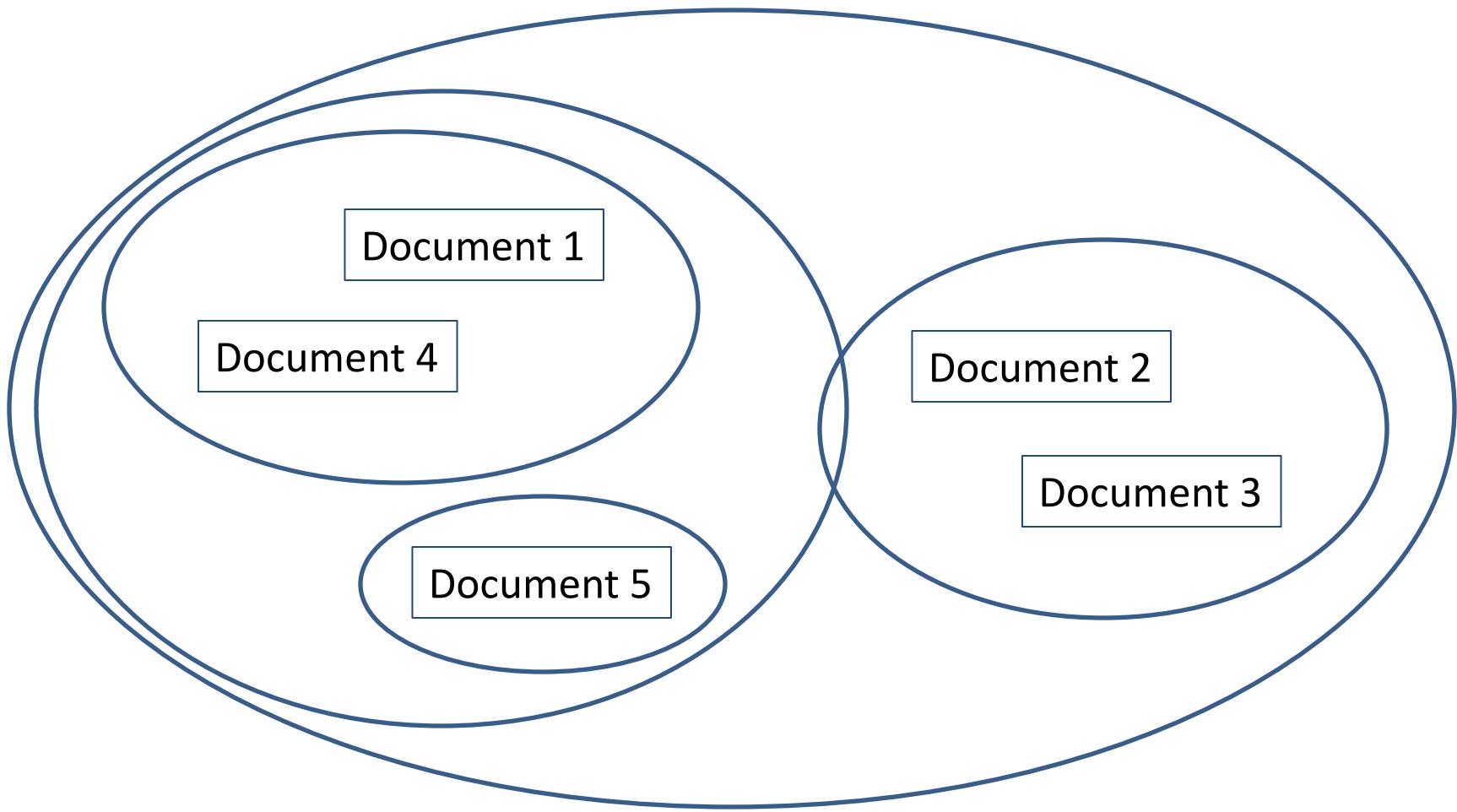
Method 1: Agglomerative Hierarchical Clustering

- And again



Method 1: Agglomerative Hierarchical Clustering

- And you get a hierarchy of clusters



Measure the distance between 2 documents?

Model the document as:

- a) Vector Space
- b) Probabilistic Distribution

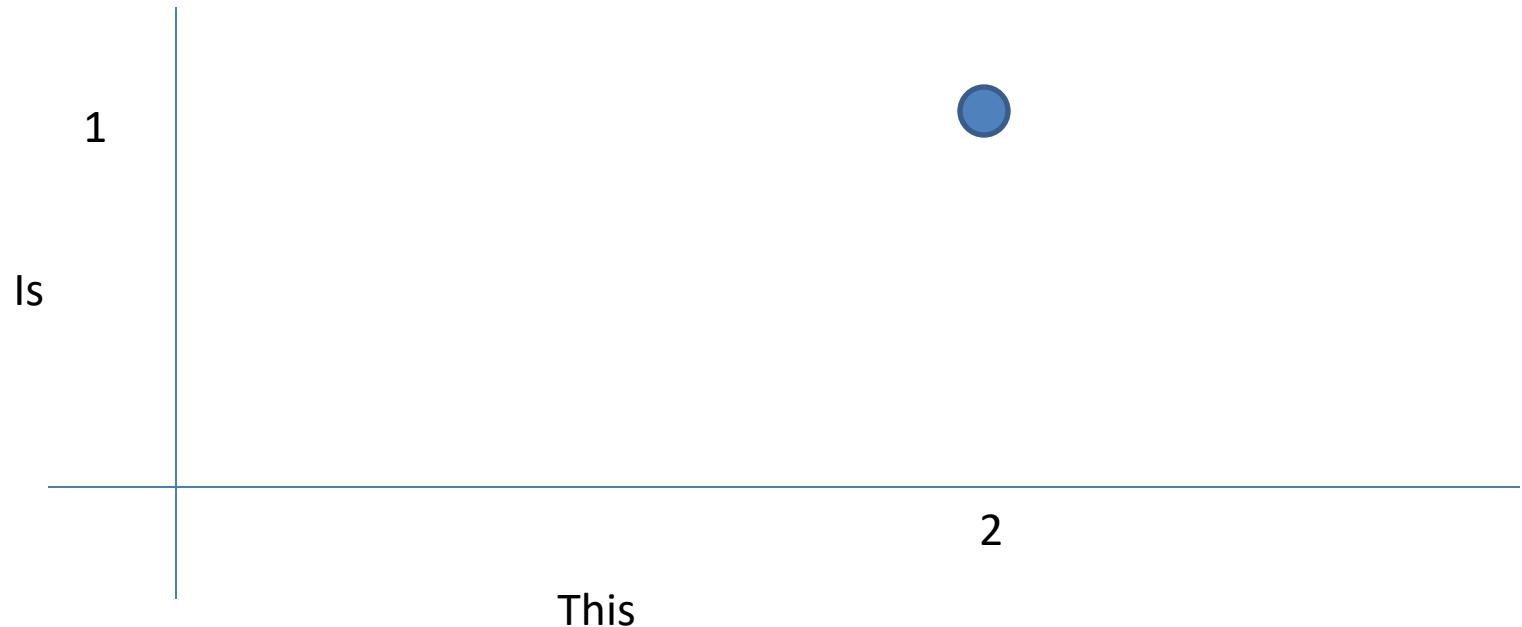
Then apply a distance measure or divergence:

- a) Cosine Distance (vector space)
- b) Manhattan Distance (vector space)
- c) KL Divergence (probability distribution)

Document 1

{“This”, 2}, {“Is”, 1}

Vocabulary: 2 (can be represented in a 2-dimensional space)



Document 2

{"That", 2}, {"Is", 1}

Vocabulary: 2 (can be represented in a 2-dimensional space)

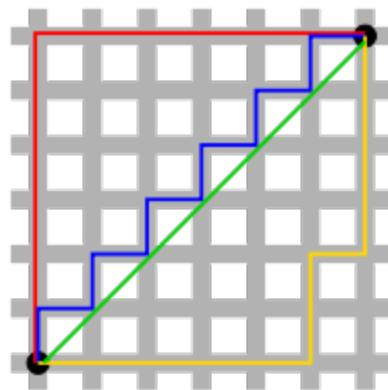


Manhattan Distance

{"This",2}, {"Is",1}, {"That",0}

{"This",0}, {"Is",1}, {"That",2}

Vocabulary: 3 (can be represented in a 3-dimensional space)



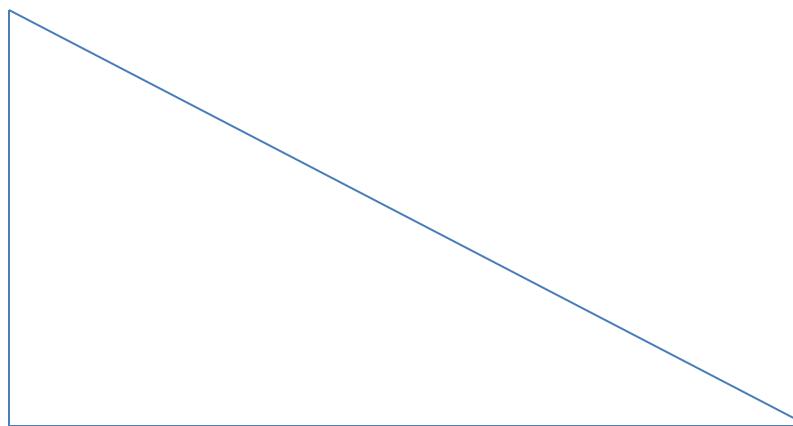
$$2 + 0 + 2$$

Euclidean Distance

{"This",2}, {"Is",1}, {"That",0}

{"This",0}, {"Is",1}, {"That",2}

Vocabulary: 3 (can be represented in a 3-dimensional space)



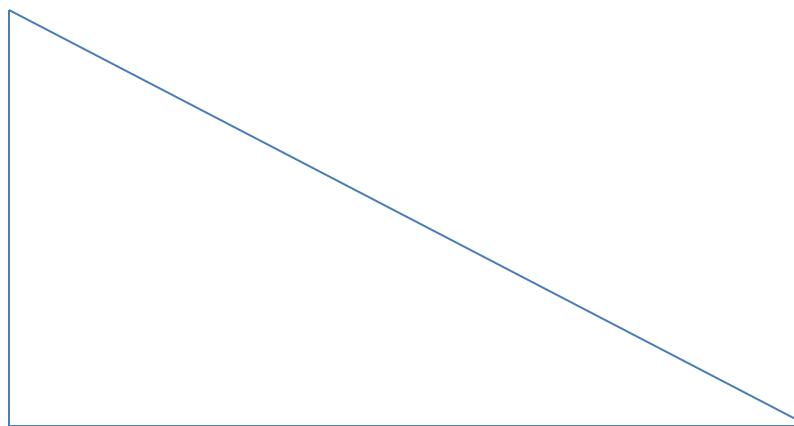
$$\text{Sqrt}(2^2 + 0 + 2^2)$$

Cosine Distance

{"This",2}, {"Is",1}, {"That",0}

{"This",0}, {"Is",1}, {"That",2}

Vocabulary: 3 (can be represented in a 3-dimensional space)



$$(0 + 1 + 0) / \sqrt{4+ 1+0} * \sqrt{0+1+4} = 1/5$$

Did you know

... that you can use a classifier ...

... to do clustering?

The exercise is ... think about it ...

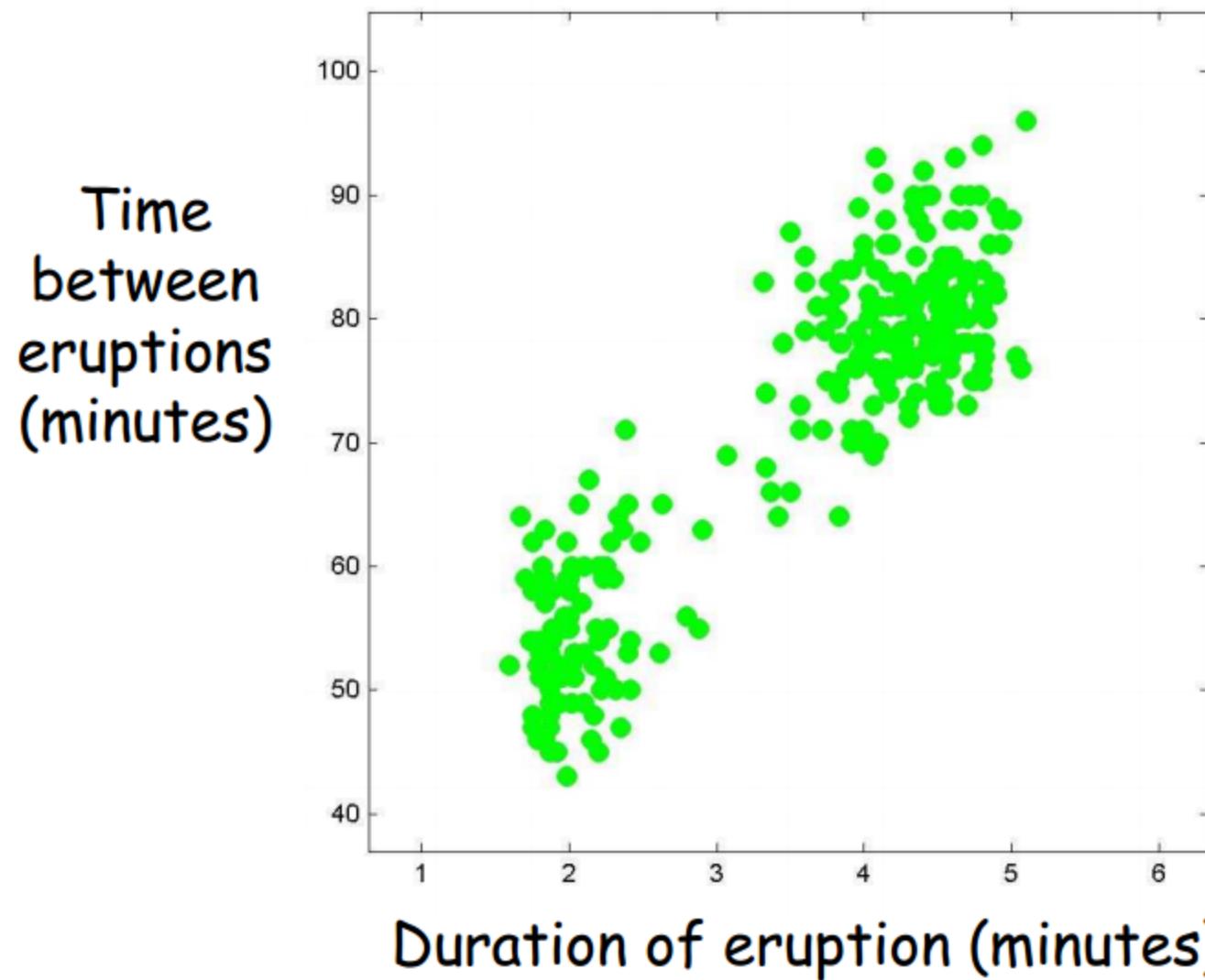
Method 2: K-Means

Initialize prototypes, then iterate between:

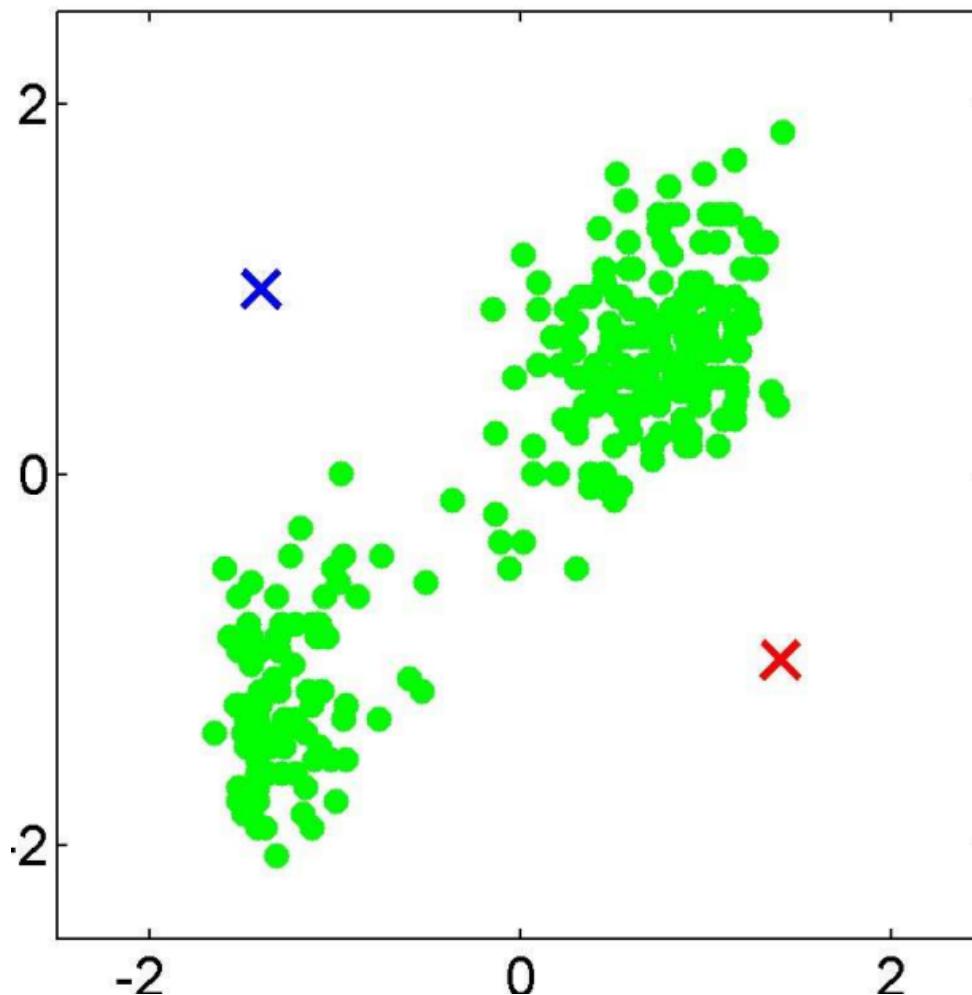
- E-step: assign each data point to nearest prototype
- M-step: update prototypes to be the cluster means

The simplest version is based on Euclidean distance

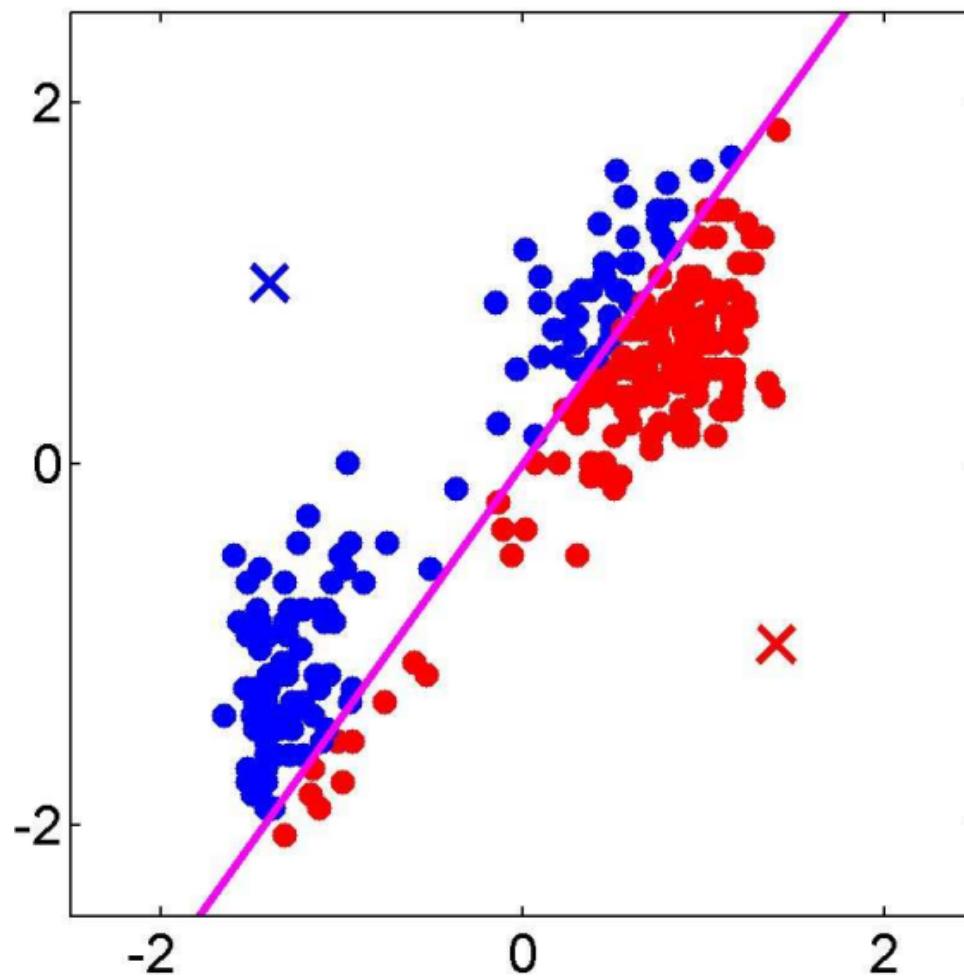
Method 2: K-Means



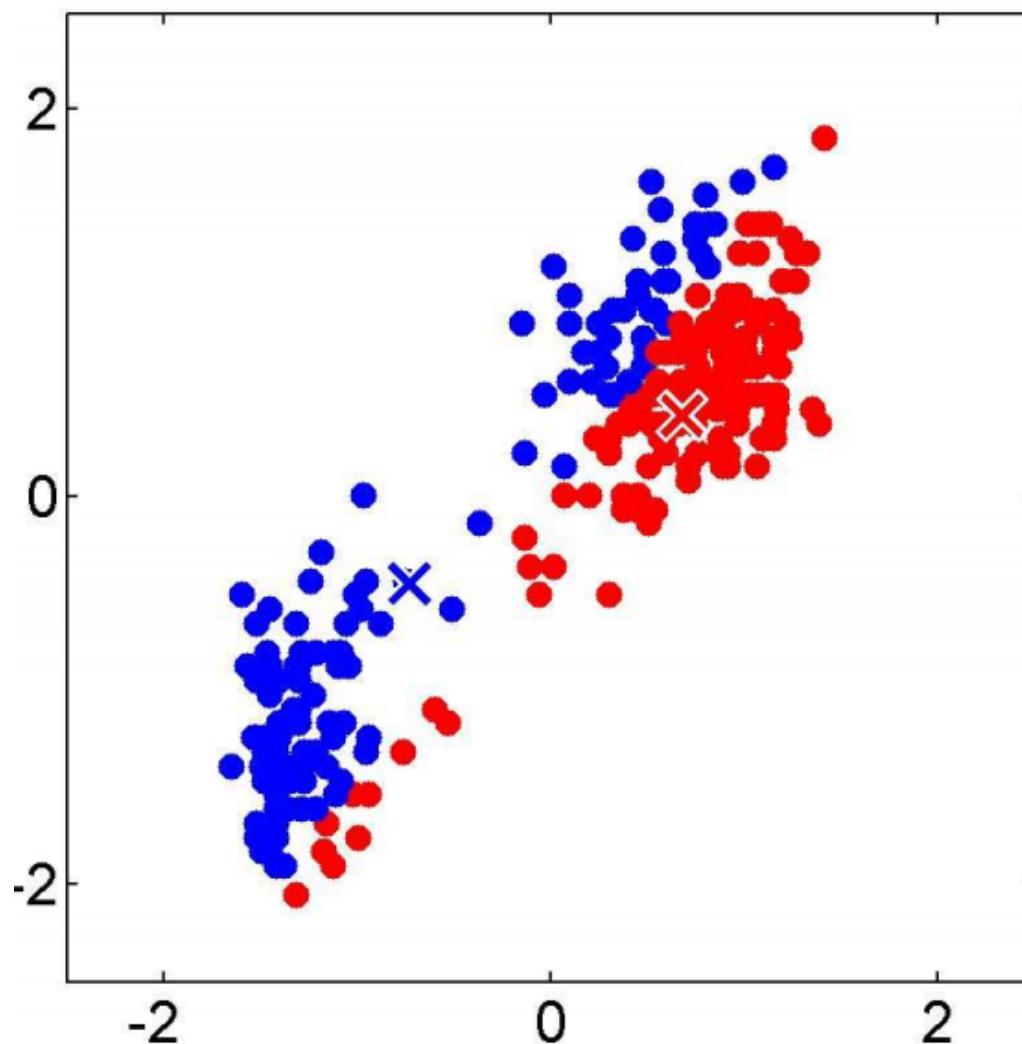
Method 2: K-Means



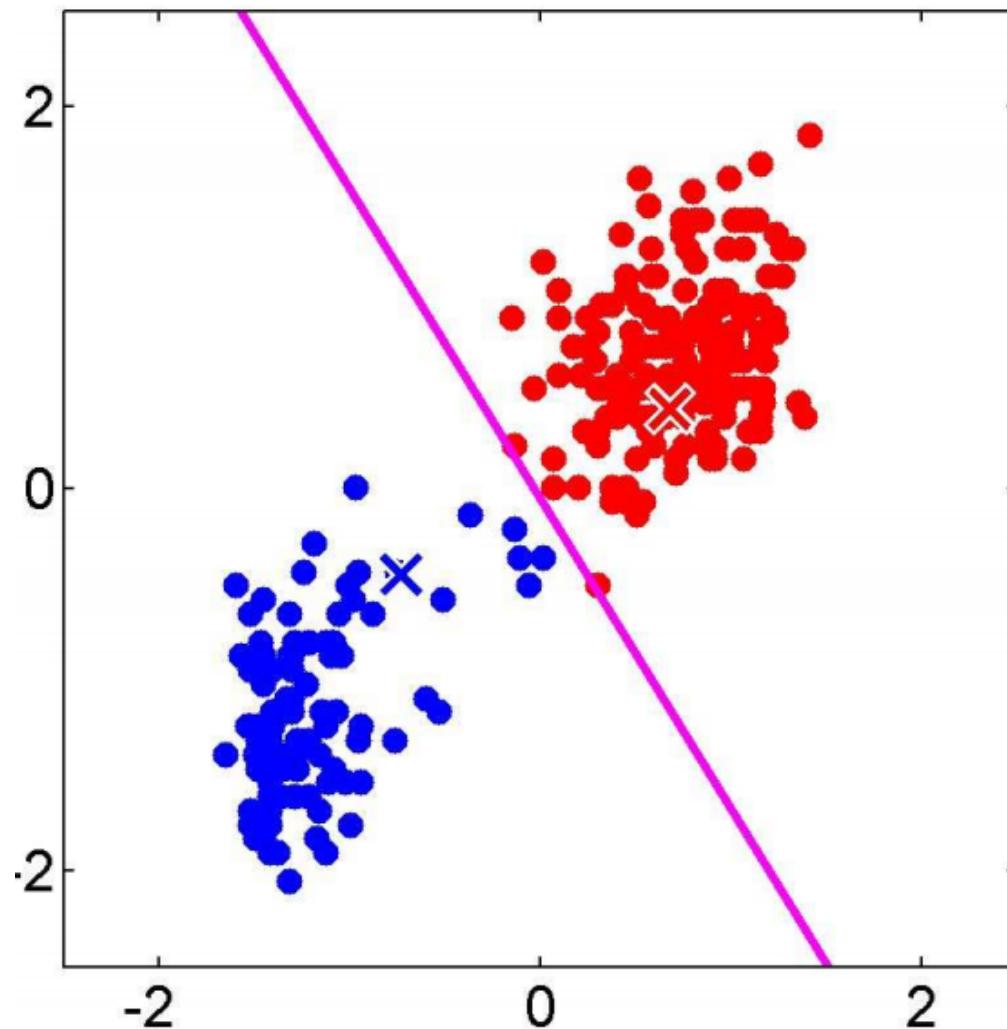
Method 2: K-Means



Method 2: K-Means

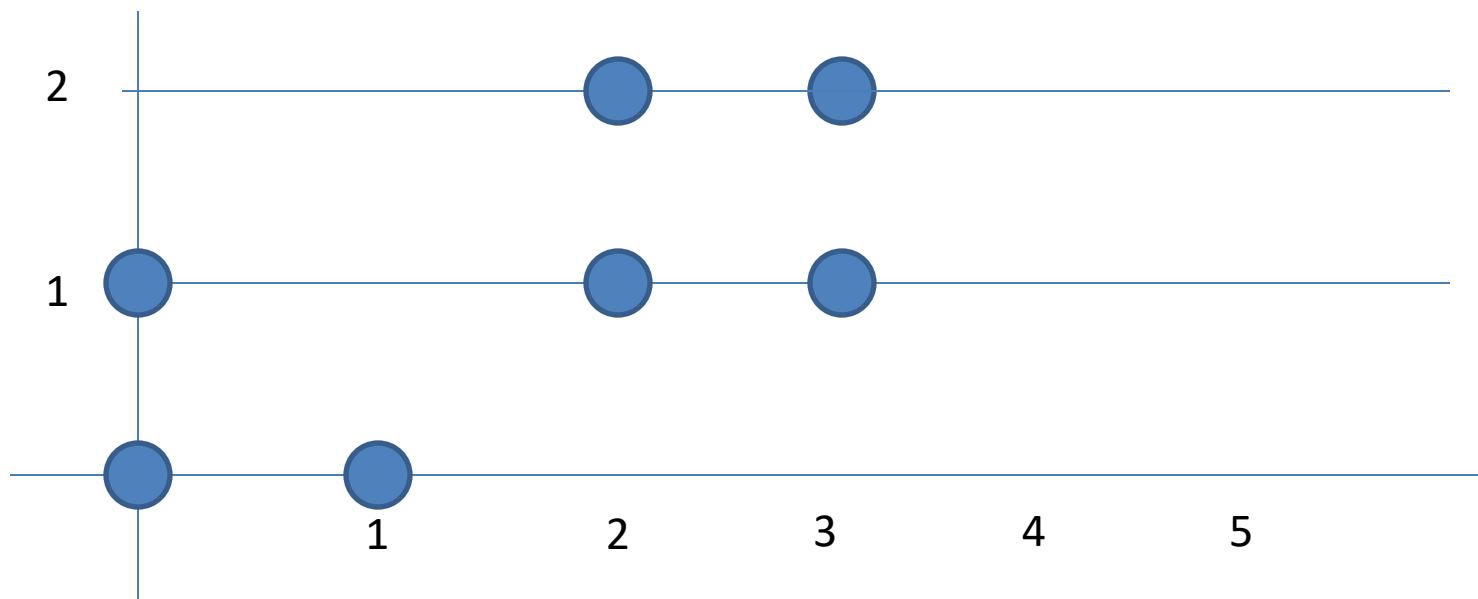


Method 2: K-Means



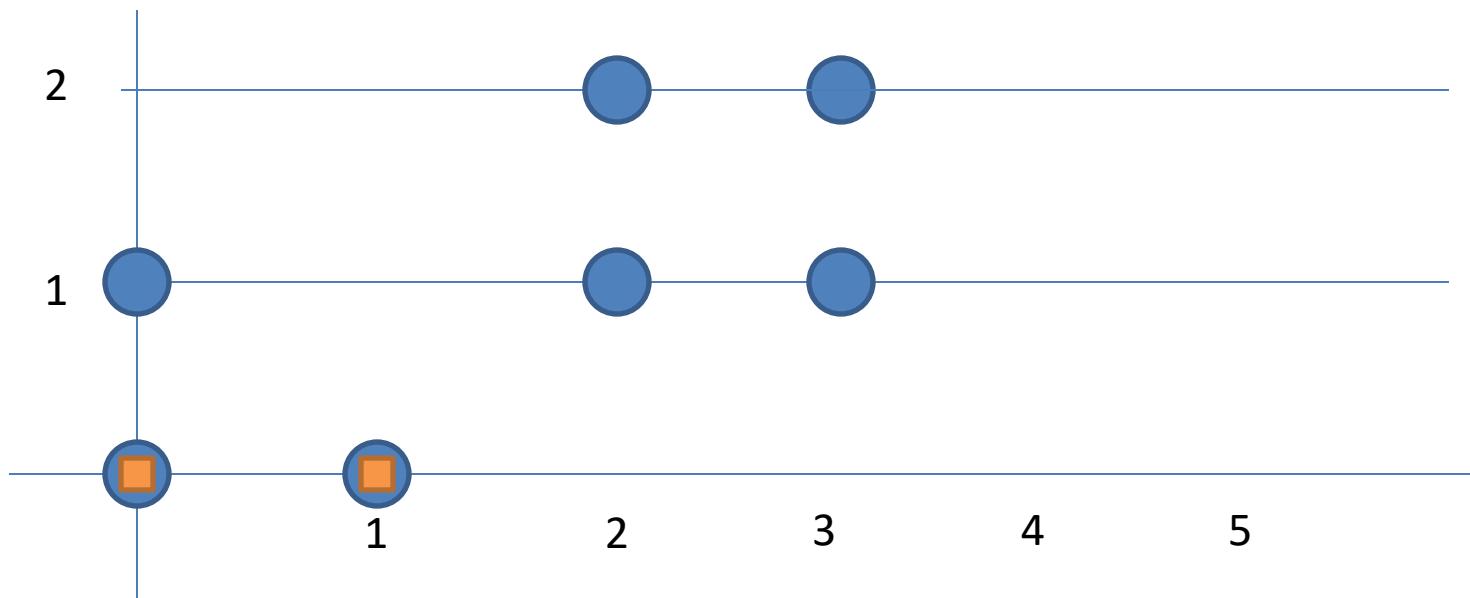
Example: K-Means

A 2-dimensional space



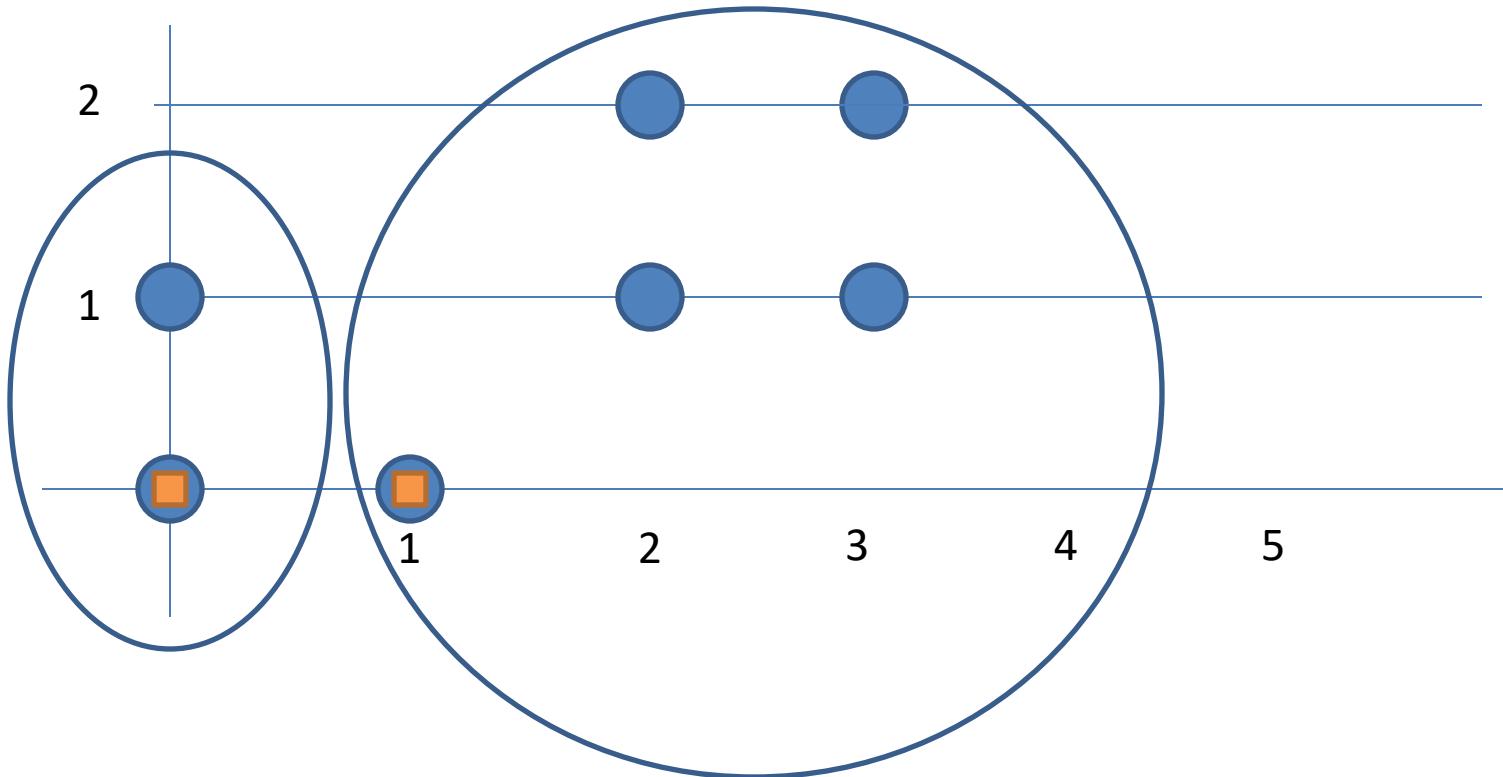
Example: K-Means

A 2-dimensional space – you want 2 clusters –
Pick 2 centroids at random



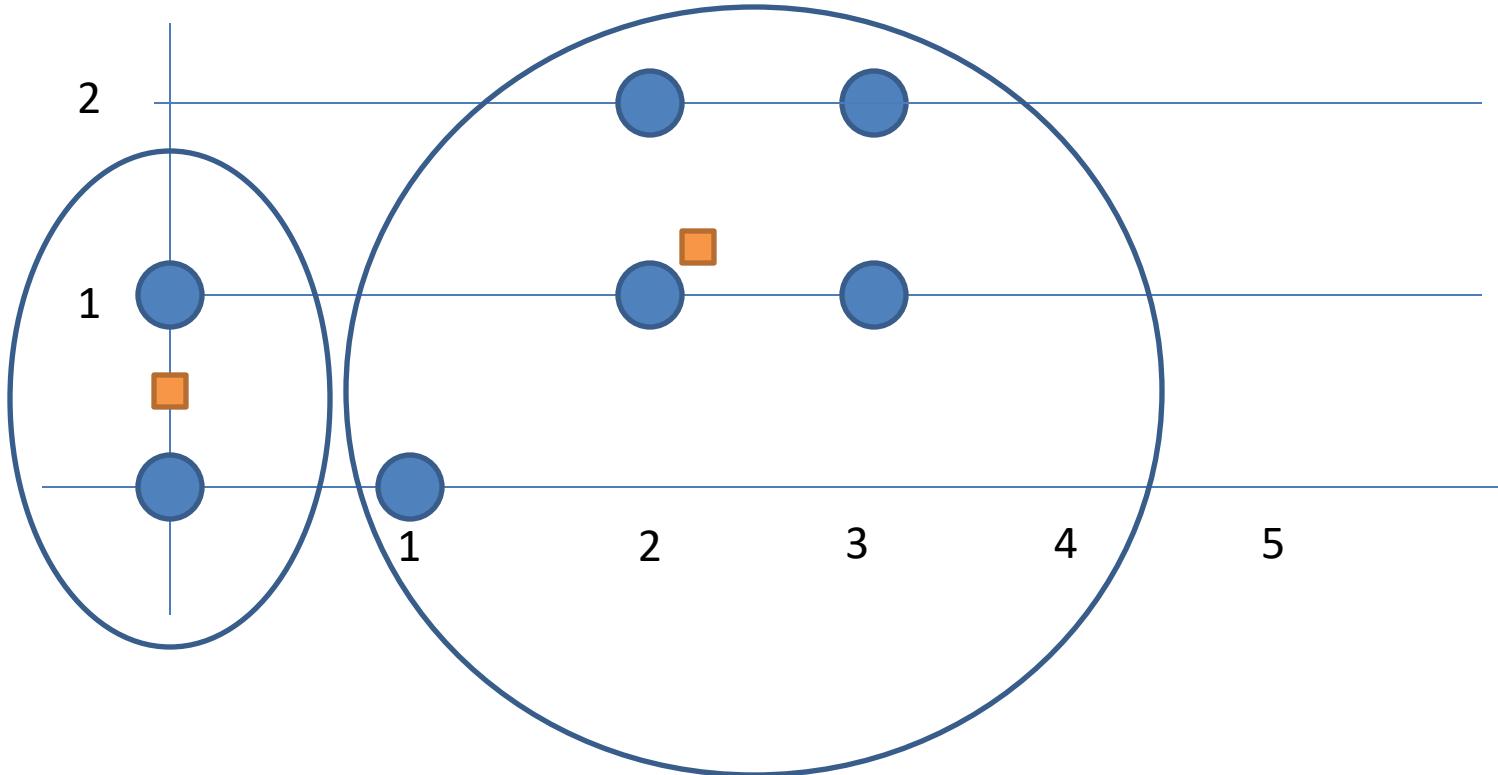
Example: K-Means

E-Step: Assign each of the points to the nearest centroid



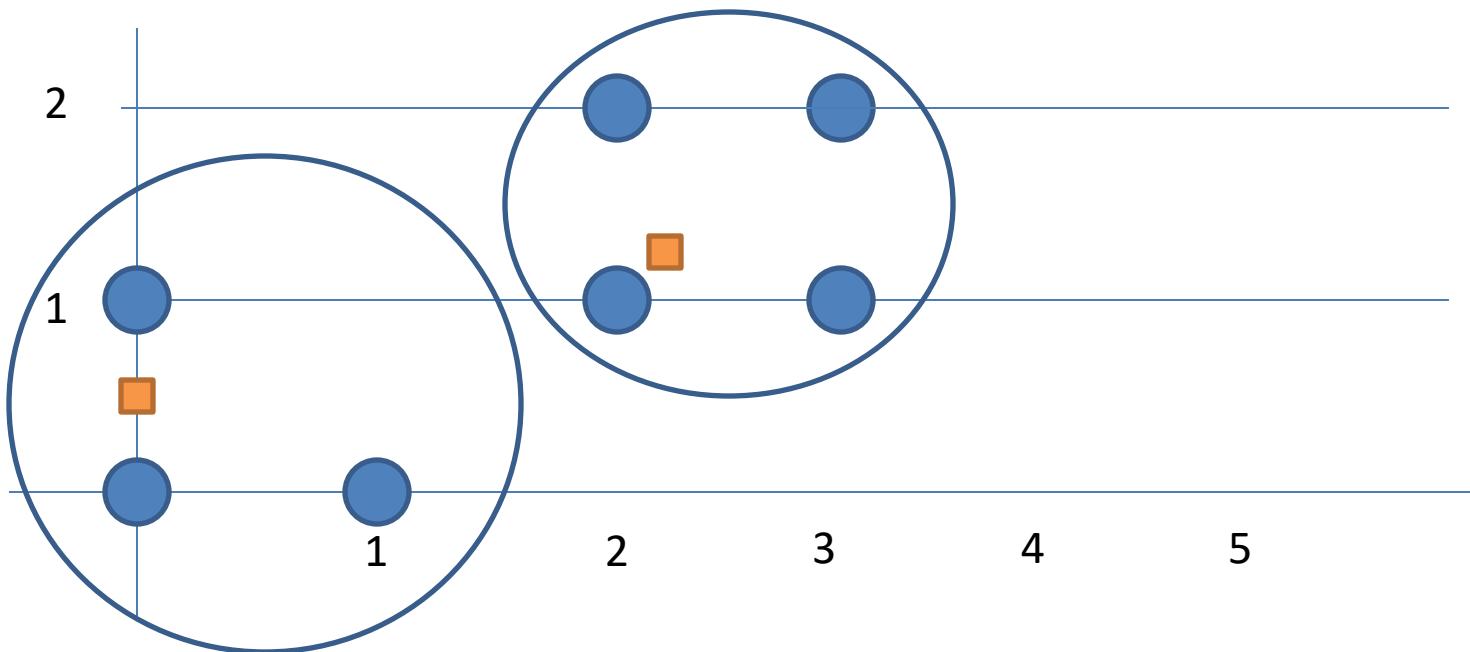
Example: K-Means

M-Step: Recompute the centroid as the center of all its points



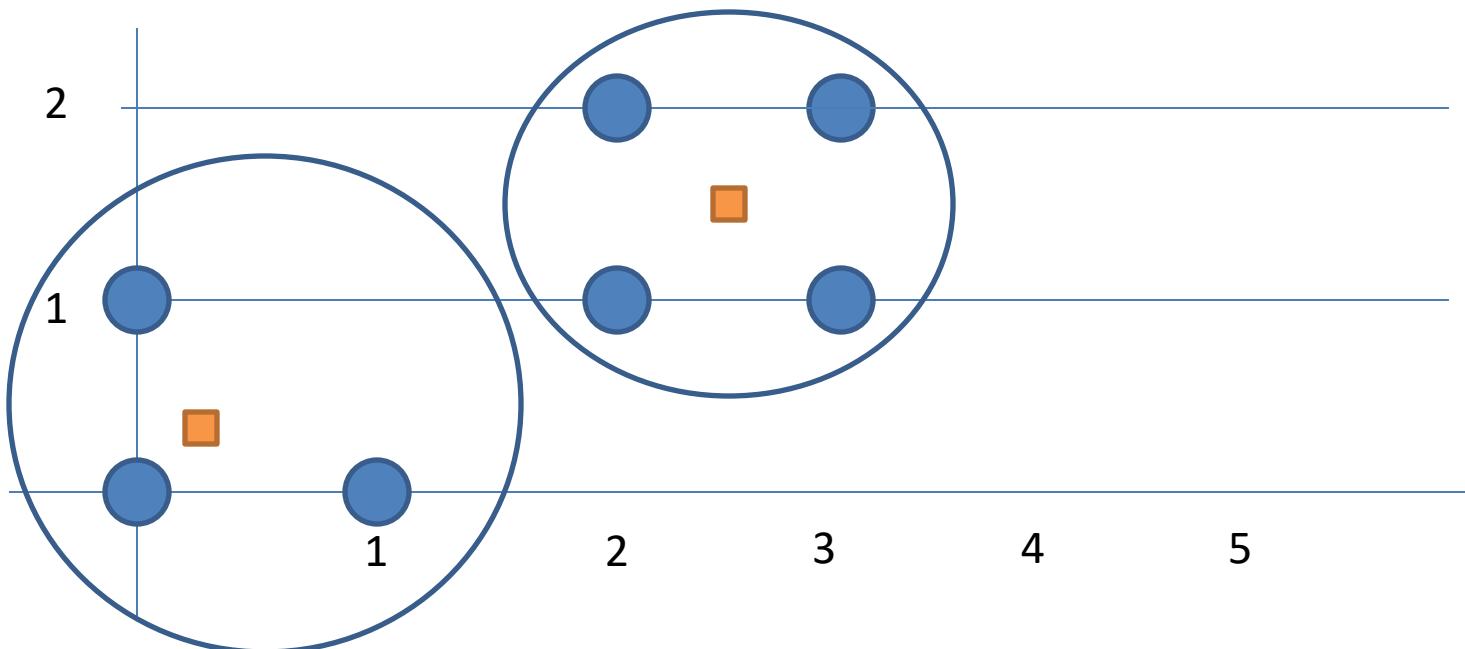
Example: K-Means

E-Step: Assign each of the points to the nearest centroid



Example: K-Means

M-Step: Recompute the centroid as the center of all its points



Expectation Maximization

K-means Cost Function

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \| \mathbf{x}_n - \boldsymbol{\mu}_k \|^2$$

Diagram illustrating the K-means cost function components:

- responsibilities**: Points to the term r_{nk} .
- data**: Points to the term \mathbf{x}_n .
- prototypes**: Points to the term $\boldsymbol{\mu}_k$.

Expectation Maximization

Responsibilities

- *Responsibilities* assign data points to clusters

$$r_{nk} \in \{0, 1\}$$

such that

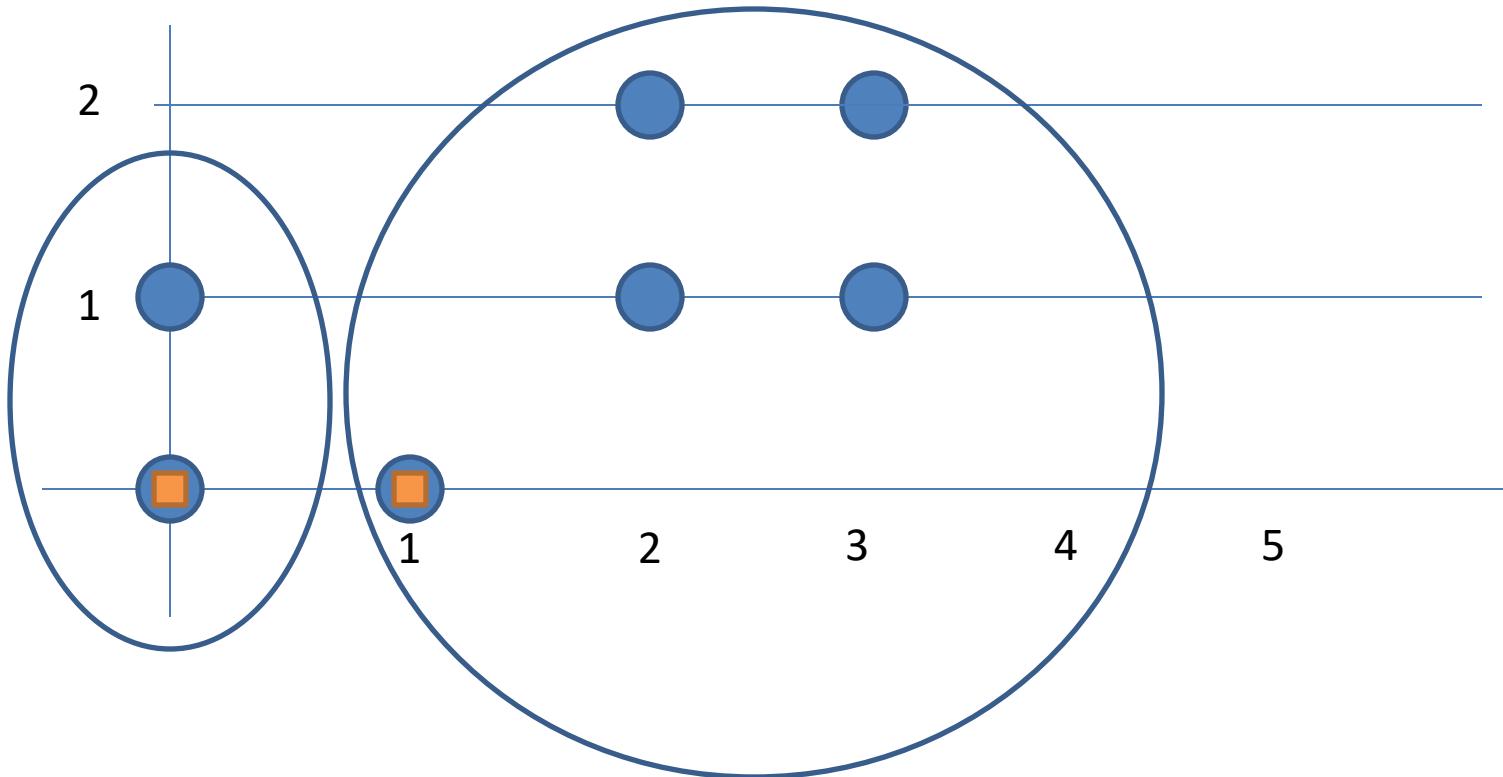
$$\sum_k r_{nk} = 1$$

- Example: 5 data points and 3 clusters

$$(r_{nk}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

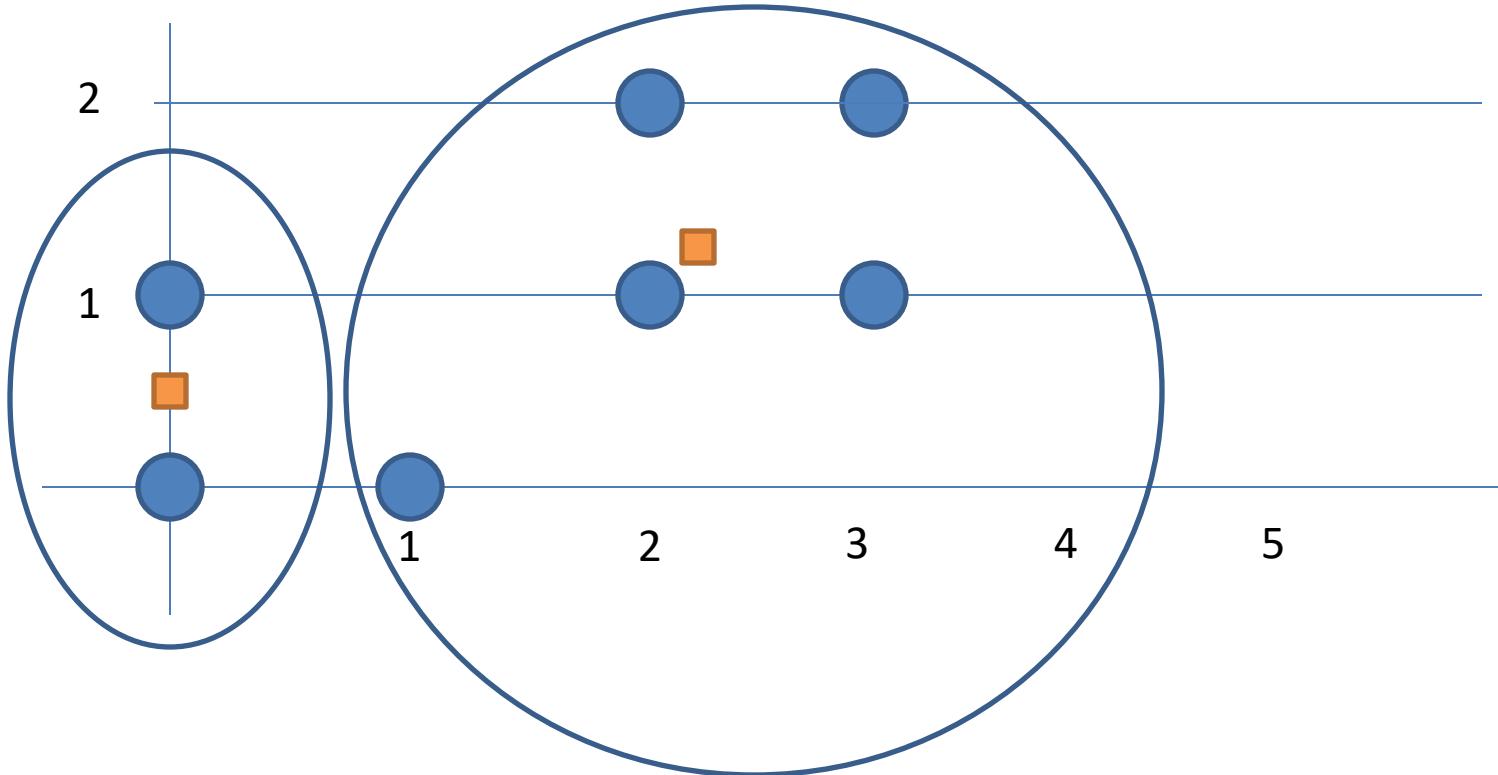
Example: K-Means

E-Step: Assign each of the points to the nearest centroid



Example: K-Means

M-Step: Recompute the centroid as the center of all its points



Problem 10: Clustering

Clustering documents into N clusters

Train classifier 1 with documents randomly assigned to one of N categories.

Classify documents using classifier 1.

Train classifier 2 using the documents classified by the previous classifier (1).

Classify documents using classifier 2.

Repeat K times ...

The output of classifier K will be in N clusters.

Time for Exercises!

Let's construct a clustering tool using a classifier!

Extraction with Unlabelled Data

Can we do extraction with unlabelled data?

Key-phrase Extraction

Bangalore, India

Discovering Key-phrases for Entity Extraction



Discovering Keyphrases

Corpus:

- 1:** Is the Canon 5 a good camera?
- 2:** Is the Canon factory close to Mumbai?
- 3:** Is the Canon camera made in Mumbai?

Find the words that stand out.

Canon, Mumbai

Why not: is, the, ?

Supervised Algorithm for extracting key-phrases

1: Generate ngrams

2: Classify some as keyphrases

Unsupervised Algorithm for extracting key-phrases

1: Generate ngrams

2: The ones that stand out statistically from a background corpus are the keyphrases

Discovering Keyphrases

Corpus:

- 1: Is the Canon 5 a good camera?
- 2: Is the Canon factory close to Mumbai?
- 3: Is the Canon camera made in Mumbai?

Find the words that stand out.

Canon, Mumbai

Canon and Mumbai are uncommon words that occur more often than they typically do in here.

Corpus:

- 1:** Is the Canon 5 a good camera?
- 2:** Is the Canon factory close to Mumbai?
- 3:** Is the Canon camera made in Mumbai?

Find the words that stand out.

Is, the, ?

Is, the and **?** are common tokens in any corpus.

Algorithm for extracting key-phrases

1: Generate ngrams

Easy to do from a corpus

2: Throw out all patterns that are outstanding

Use a statistical measure

Statistical Measures for Identifying Key-Phrases

1: Probability Ratios

List the words with the highest probability ratios

Probability Ratio: probability in this corpus vs probability in a background corpus

Issues with Probability Ratios:

- 1: Does not work well for rare words (they get probability ratios of 1/1 or 0/0)

1: Probability Ratios

List the words with the highest probability ratios

Probability Ratio: probability in this corpus vs probability in a background corpus

2: Pointwise Mutual Information

Useful for multiple words

PMI: *probability of two words occurring one after the other divided by the product of their individual probabilities*

2: Pointwise Mutual Information

Useful for multiple words

PMI:

$$\text{pmi}(x; y) \equiv \log \frac{p(x, y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}.$$

2: Kullback Leibler Divergence

Useful for individual or multiple words

KL Divergence: Kullback–Leibler divergence of Q from P , denoted $D_{\text{KL}}(P||Q)$, is a measure of the information lost when Q is used to approximate P

2: Kullback Leibler Divergence

Useful for individual or multiple words

KL Divergence:

$$D_{\text{KL}}(P\|Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)}.$$

Real world text analytics problem

Let's see how you might use text analytics to solve a marketing problem!

B2B Marketing!!!
Promoting a B2B Brand

One Text Analytics Project

Applications of Text Analytics to B2B Sales and Marketing

The Problem: Marketing B2B Brands

1. How We Promote B2B Products Today
2. What Can Happen When You Apply Text Analytics

B2C and B2B Products

B2C

Shop for apple iphone on Google



Apple iPhone
5s (Silver, ...
Rs. 62,500.00
Amazon.in

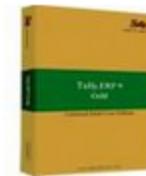


Apple iPhone
5s (Space ...
Rs. 62,500.00
Amazon.in



Apple iPhone
5c (Green, ...
Rs. 37,207.00
Amazon.in

B2B



Tally ERP 9
Accounting ...
Rs. 51,000.00
smesauda

Advertisement

Advertisement

The difference between B2C and B2B marketing

is one of
targeting

If the product is a B2C product

**B2C – A large *segment* of the world is a
Prospective Buyer**



Customer



Customer



Customer



Accountant

If the product is a B2B product

B2B – People in Specific Roles in Firms are
Prospective Buyers



Not a customer



Not a customer



Customer



Accountant

B2B Product Buyer

1. The buyer is a role in a firm
2. There is a limited number of buyers
3. More targeting needed to reach buyers efficiently

How We Promote B2B Products Today

Digital Marketing

1. Search Engine Marketing

Images for enterprise resource planning



Report images

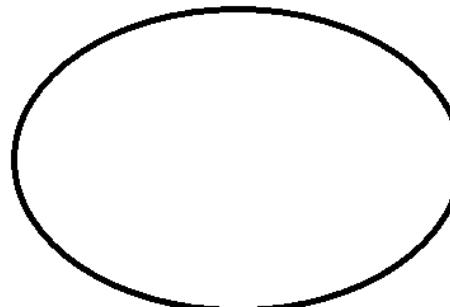


More images for enterprise resource planning

ERP Software | Business Process | Solutions | SAP

www.sap.com/india/pc/bp/erp.html ▾

Support innovation, operational excellence, and end-to-end business processes with our market-leading enterprise resource planning (ERP) software.



2. Social Media Marketing

A screenshot of SAP's official Facebook page. The cover photo features the SAP logo and the text 'SAP Computers/Technology'. The page has a 'Timeline' tab selected. A status update from 'Abhijith Karthikeya' is visible, stating '807k people like this' and 'Abhijith Karthikeya and 3 other friends'. There are 'Watch Video' and 'Like' buttons at the top right. Below the timeline, there are tabs for 'About', 'Photos', 'Events', and 'More'. A 'Post' button is located at the bottom left.

Content Marketing

Marketing over the web, social media, and email by sharing content that helps customers or prospects:

1. Content creation around keywords
2. Content dissemination on digital media
3. Lead capture using forms, polls and gated downloads

This is a bit of a superset of SEO (SEO is about making your website rank high on search engines).

Example of Content Marketing

Let's say I am Tally, and I want to promote tools for financial accounting to Chartered Accountants.

1. Content creation around keywords that potential customers might search for:

accounting, accounting software, accounting tools,
tax preparation, taxation, cost accounting, tax compliance

2. Content dissemination on digital media:

Web pages with articles including those keywords

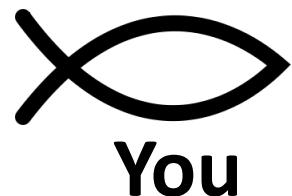
[http://tallsolutions.com/5- accounting -tips-for-small-firms.html](http://tallsolutions.com/5-accounting-tips-for-small-firms.html)

Title: 5 Amazing **Accounting** Tips for Small Firms

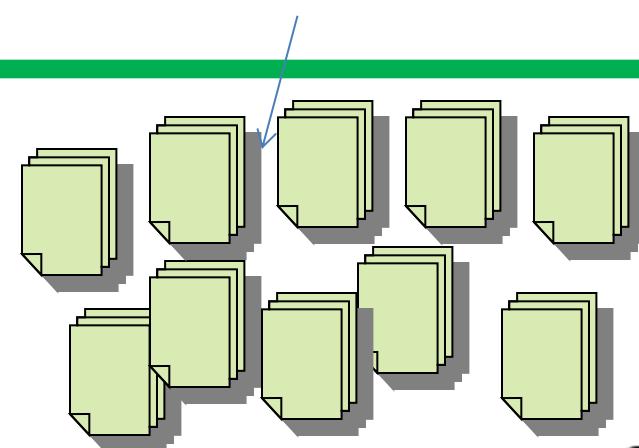
Content: Has your **accounting** given you spent sleepless nights

Where Your Content Marketing Doesn't Work Great

Your content



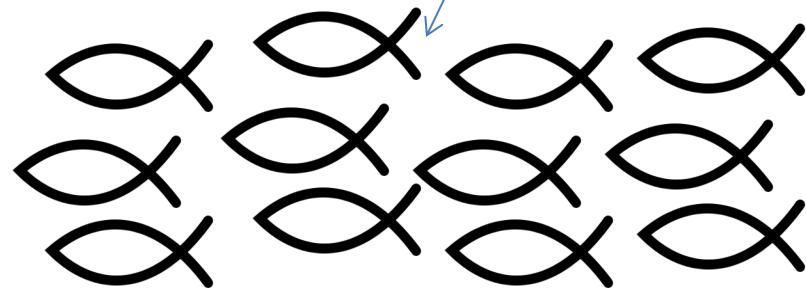
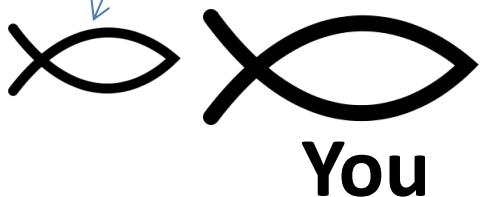
Your competitor's content



Where Your Content Marketing Doesn't Work Great

Your followers

Your competitor's followers



Your competitor

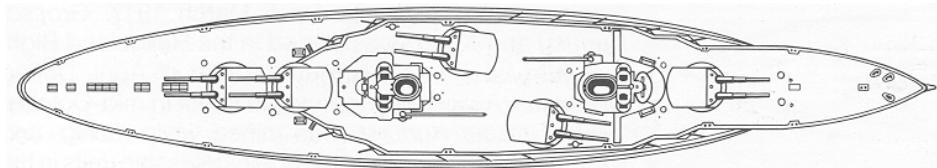
The above methods rely
on “being found”

Traditional Digital Marketing Works Just Like A Tool For Naval Warfare



Naval Mine

How Naval Mines Sink A Ship



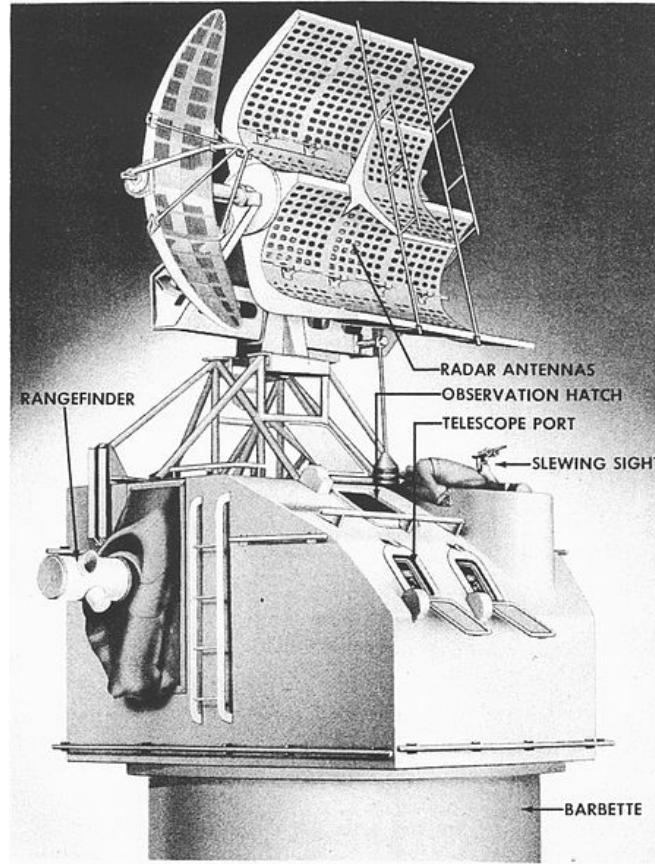
Prospect

Content

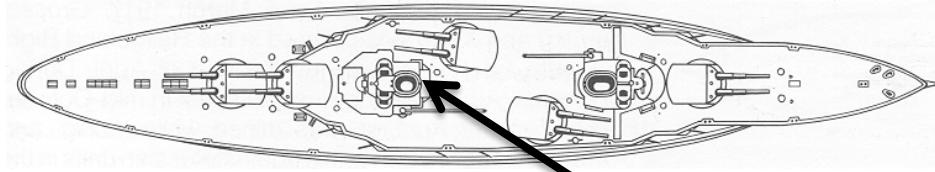
Can Text Analytics Do Things Differently?

Another Way to Sink a Ship

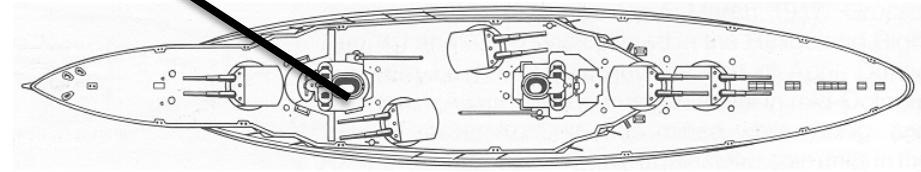
“going out and finding
the customer”



How Gun Fire Sinks A Ship



Prospect



You

Another Way to do B2B Marketing (Sink a Ship) Using Text Analytics

Use Text Analytics To:

1. Identify a Prospect (Spot a Target)
2. Find Contact Information (Location)
3. Find Out Needs and Interests (Find Heading)
4. Interact With Prospect (Fire a Shell)

Our research work over the past 5 years has focused on doing these 4 steps using AI (automatically).

1. Identify a Prospect

Get the identity of prospective buyers from:

- a) Company websites
- b) Social media profiles
- c) News articles

Identifying Prospects – An Example

The screenshot shows the Selasdia software interface for identifying prospects. At the top, there are tabs for Person Search (highlighted in orange), Company Search, Discover Twitter, Email Verifier, and Quick Create. To the right of the tabs are social media icons for LinkedIn, Twitter, Facebook, Google+, and a user profile icon. Below the tabs, there are radio buttons for CEO, CMO, CTO, and Other, with 'CEO' selected. A search bar contains the company name 'Minewhat'. A 'Search' button is located to the right of the search bar. A yellow callout box displays the results for Janakiram (Ram) Ganesan, identified as the founder of Minewhat, with a LinkedIn URL provided. Below the callout are 'Save' and 'Get URL and Email' buttons.

selasdia

Person Search Company Search Discover Twitter Email Verifier Quick Create

CEO CMO CTO Other, please specify

Enter company Minewhat

Search

Janakiram (Ram) Ganesan
founder of Minewhat
<http://in.linkedin.com/in/janakg>

Save

Get URL and Email

Web Information Extraction for List Building

Modern AI tools can search the world wide web
much like a human would.

The screenshot shows a user interface for searching company executives. At the top, there is a horizontal radio button group for selecting a role: CEO (selected), CMO, CTO, and Other, please specify. Below this is a search form with a label 'Enter company' followed by a text input field containing 'Apple'. To the right of the input field is a green 'Search' button. The search results for Apple show a yellow box containing the name 'Tim Cook' and the title 'Apple CEO'. Below this, another search form is shown for Microsoft, with the company name entered into the input field. The search results for Microsoft show a yellow box containing the name 'Satya Nadella' and the title 'CEO of Microsoft'. At the bottom of the interface are two buttons: a green 'Save' button and a green 'Get URL and Email' button.

CEO CMO CTO Other, please specify

Enter company Apple

Search

Tim Cook
Apple CEO

CEO CMO CTO Other, please specify

Enter company Microsoft

Search

Satya Nadella
CEO of Microsoft

Save

Get URL and Email

Results: B2B Prospect List Culled from Twitter

Person

 **CP Unterberger**
CMO International Business, Fujitsu

RT @carte1s: "@fujitsu_uk: Delighted win the @holmesreport SABRE Award for B2B Marketing with @Harvard_PR - so #Proc 2014-05-21 +

 2014-05-21 08:59:57.0 142 followers

 **Mark Sarbiewski**
CMO at Anaplan, the leading SaaS enterprise planning company. Tech junkie, long suffering Browns fan, locked in a room with my kids.

RT @AnaplanCareers: .@Anaplan is #HIRING! Enterprise Sales Executive - Benelux. #saas #EPM #job <http://t.co/zCkmYJF>

 2014-05-20 19:30:16.0 238 followers

 **Evelina Šaulytė**
Family #1 // positive psychology addict // food lover // Lithuanian // CMO of Invoiceberry

How to Write a Blog Post Outline: A Simple Formula to Follow: <http://t.co/u7D1il2Tqo> via @hubspot - 2014-05-20 +

 2014-05-20 13:04:07.0 66 followers

Text Analytics Steps

Two Steps:

1. Named Entity Extraction/Classification
2. Relationship Extraction

Text Analytics Steps

Step 1: Named Entity Extraction / Classification:

- a) ... said **Janakiram**, the CEO of **Minewhat**, while ...
- b) Attendees included **Janakiram**, CEO, **Minewhat** ...

Output:

Person: Janakiram

Company: Minewhat

Text Analytics Steps

Step 2: Relationship Extraction:

- a) ... said **Janakiram**, the CEO of **Minewhat**, while ...
- b) Attendees included **Janakiram**, CEO, **Minewhat** ...

Output:

CEO_Of {Janakiram, Minewhat}

2. Obtain Contact Details

Get the contact details for prospective buyers from:

- a) Company websites
- b) Crunchbase
- c) Email servers
- d) Social media

Obtaining Contact Details

Person Search Company Search Discover Twitter **Email Verifier** Quick Create 

Enter email address Eg: cohan@aiaioo.com **Verify**

OR

Enter name Cohan Sujay Carlos

Enter company url aiaioo.com

Discover

Email address is **cohan@aiaioo.com**

Save

Tried email addresses

cohan@aiaioo.com

Text Analytics Task

Record Linkage:

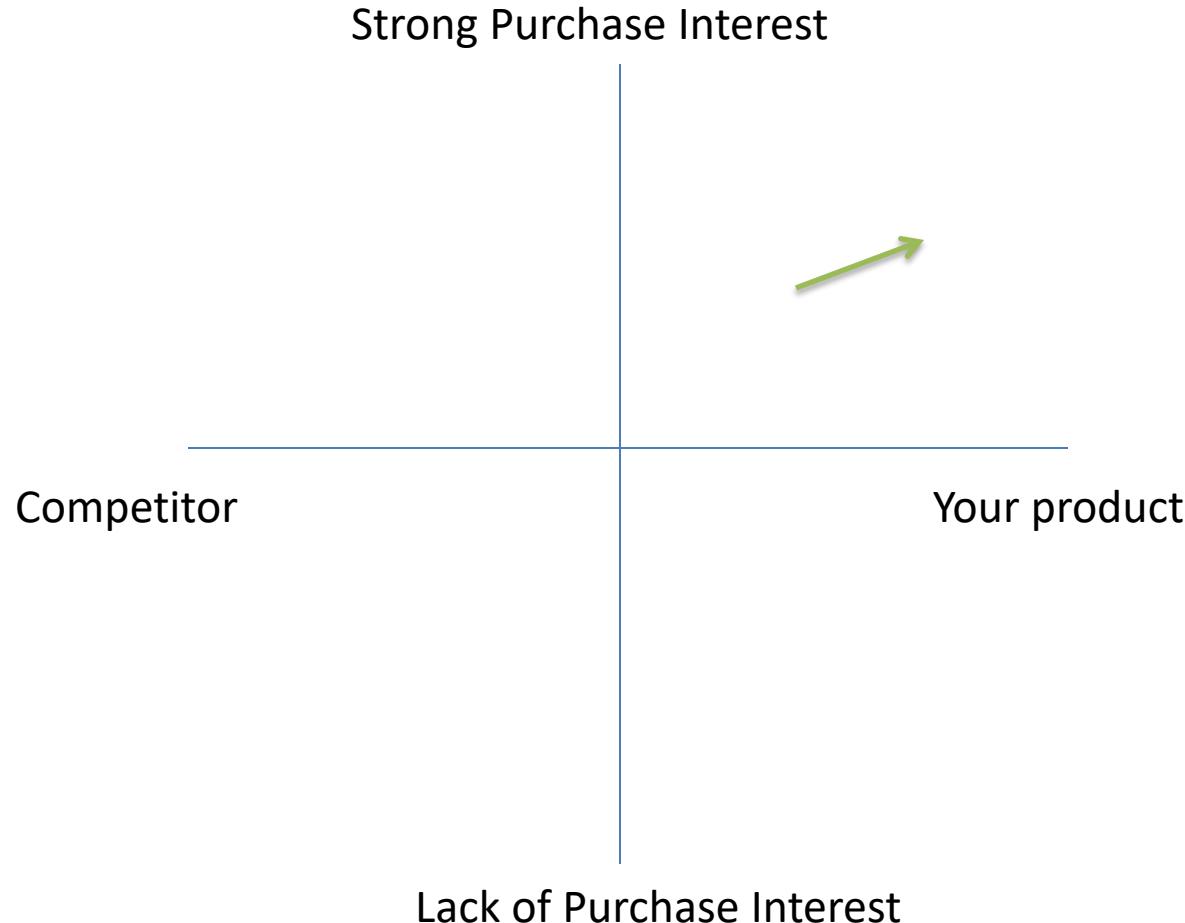
cohan@aiaioo.com \Leftrightarrow Cohan Sujay Carlos

@cohansujay \Leftrightarrow Cohan Sujay Carlos

facebook.com/cohan.sujay \Leftrightarrow Cohan Sujay Carlos

3. Determine Heading

With a prospect, you want to know shifts in interest:



Text Analytics Steps

Intention Analysis – Purchase Intention

Enter a sentence into the box below and click on the demo button.

I would like to buy a car like that beautiful VW we saw near San Mateo.

Run Vakintent Intention Analyzer Demo

Sentence 1: I would like to buy a car like that beautiful VW we saw near San Mateo. = [the intention is to purchase]

Object of Intention	Intention
a car like that beautiful VW we saw near San Mateo	purchase
Holders	Negated?
I	false

Intention Analysis – Sell Intention

Enter a sentence into the box below and click on the demo button.

For sale: a new Gibson electric with amplifier.

Run Vakintent Intention Analyzer Demo

Sentence 1: For sale: a new Gibson electric with amplifier. = [the intention is to sell]

Object of Intention

a new Gibson electric with amplifier .

Intention

sell

Intention Analysis – Categories of Intention

Categories	Parent Category	Department	Urgency	Source
Purchase		Sales	High	CRM
Sell		Procurement	Medium	ERP
Inquire		Help/Sales	High	CRM
Direct		Operations	High	CRM
Compare		Market Research	Low	CRM
Suggest		Market Research	Low	Social
Opine		Design	Low	Social
Praise	Opine	Design	Low	Social
Criticize	Opine	Design	Low	Social
Complain		Customer Service	High	CRM/Social
Accuse		Customer Service	Critical	CRM
Quit		Customer Service	Critical	CRM/Social
Express		Call Center Training	Low	Transcripts
Thank	Express	Call Center Training	Low	Transcripts
Apologize	Express	Call Center Training	Medium	Transcripts
Empathize	Express	Call Center Training	Medium	Transcripts

The following give you clues to the level of interest

I just read this article about B2B marketing!

How do people go about doing B2B marketing?

I am looking for good tools for B2B marketing.

Is tool XYZ for B2B marketing better than tool ABC?

Examples of Leads at the Moment of Purchase Found using *Intention Analysis*

A message has been received: 'ExactTarget or YesMail and why? Go!'. Respond to it if that will help you build a

A message has been received: 'Looking for a SM management platform for a client. Give me thoughts between @hootsuite and @Vocus. [\[View original\]](#)'. Respond to it if it is a lead.

A message has been received: 'Looking for a digital marketing agency in Mumbai for a campaign. Please send profile and contact details to [uPzk7H01RB](#)

A message has been received: 'What are everyone's thoughts on the best marketing automation tools to use?'. Respond to it if that will help you build a relationship with the sender.

4. Engage

Find ways to establish some kind of interaction typically using:

- a) Safe likes
- b) Safe follows
- c) Responding to inquiries or concerns

Text Analytics Tasks

Safety checks:

- a) Ensure that any message that you like is not offensive or inappropriate.
- b) If you're sharing a message, ensure that the message being shared is not about or by your competitor.
- c) Ensure that you don't follow somebody who is politico-religious hazardous material.

What I've covered in this use-case

1. Solving a business problem end to end
2. Reducing business processes to text analytics steps

Are we done?

... kinda ...

The more training you have, the better you will
get at text analysis ... **so keep learning!**

How to Learn More?

Grab the UC Berkeley Natural Language Processing Course's slides. The course's name is CS 294.

Start reading the ACL conference's research papers (ACL = Association of Computational Linguistics)

THE END

Text Analytics Course – 2 Day

Cohan Sujay Carlos
Aiaooo Labs
Bangalore

The bottom