NICF - TEXT ANALYTICS

MODULE 6: TEXT CATEGORIZATION

Dr. Wang Aobo

Email: isswan@nus.edu.sg

Objectives of this module

At the end of this module, you can:

- Describe what is text categorization and how text categorization systems work
- Evaluate a text categorization system with respect to a business scenario
- Understand how supervised and unsupervised text categorization works
- Understand what is topic modeling

Outline for this module

- **What is text categorization?**
- **How does supervised text categorization work?**
  - Document data set
  - Building a classifier
  - Evaluation (*quiz*)
  - Running the classifier (*workshop*)
- **Text categorization application examples**
- **Unsupervised text categorization**
  - Document clustering
- **Topic Modeling**
  - LDA (*workshop*)

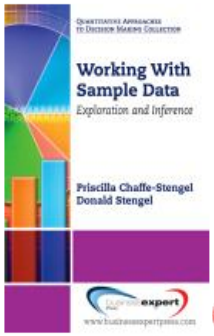
WHAT IS TEXT CATEGORIZATION?

Contrast with a library catalog

- **Example to right**
  - Subject: Statistics
- **Assigned by a cataloger**
- **Slow, tedious**
- **May be inconsistent**

Record 1 of 381 in National Library Board **Search type:** Search by Subjects
Search was: Statistics

Next



Annotation

Title Working with sample data : exploration and inference / Priscilla Chaffe-Stengel, Donald N. Stengel.
Author Chaffe-Stengel, Priscilla M.
Publisher New York : Business Expert Press, c2012.
Physical Description 151 p. : ill. ; 23 cm.
Notes Includes index.
"The quantitative approaches to decision making collection"--Cover.
Originally published in 2011.
Other Contributors Stengel, Donald N.
Search by Subjects Commercial statistics.
Statistics.

MESH index of a single journal paper

Below is an example of a **complete reference** in Medline (OvidSP) showing the journal article details and the list of MeSH headings (some with subheadings) assigned to it by the NLM Indexers:

Unique Identifier	20980007
Record Owner	From MEDLINE, a database of the U.S. National Library of Medicine.
Status	MEDLINE
Authors	Asejczyk-Widlicka M. Srodka W. Schachar RA. Pierscionek BK.
Authors Full Name	Asejczyk-Widlicka, M. Srodka, W. Schachar, R A. Pierscionek, B K.
Institution	Institute of Physics, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wrocl Poland.
Title	Material properties of the cornea and sclera: a modelling approach to test experimental analysis
Source	Journal of Biomechanics. 44(3):543-6, 2011 Feb 3.
Abbreviated Source	J Biomech. 44(3):543-6, 2011 Feb 3.
NLM Journal Name	Journal of biomechanics
Publishing Model	Journal available in: Print-Electronic Citation processed from: Internet
NLM Journal Code	0157375, hjf
Country of Publication	United States
MeSH Subject Headings	Computer Simulation *Cornea / ph [Physiology] Finite Element Analysis Humans *Intraocular Pressure / ph [Physiology] Muscle Rigidity *Sclera / ph [Physiology] Visual Acuity / ph [Physiology]
Abstract	Material prope es of cornea and sclera are important for maintaining the shape of the eye and the requisite ce curvatures for optics. They also need to withstand the forces of external and inte sculature and fluctuations in intraocular pressure (IOP). These properties are difficult to iable results have been reported. A previously published experimental procedure, material properties of the eyeball coats were obtained, has been modelled in this te Element Analysis, in order to test the accuracy of the experiment. Material e calculated from the model and the resulting relationships between stress and rnea and sclera compared to their experimentally obtained counterparts. The ween model and experiment was close for the sclera but more varied for the cornea The pressure vessel model can be applied for measuring the material properties of the sclera but is less accurate for the cornea. Copyright Copyright 2010 Elsevier Ltd. All rights reserved.

Journal article details

MeSH descriptors assigned by indexers- taken from the list of preferred terms used to describe topics

The indexers have assigned the subheading Physiology to this MeSH descriptor

From: http://library.ulster.ac.uk/sci/mesh/

Automatic text categorization (also known as “classification”)

- **Hard Classification**

  The process of assigning text documents uniquely into two or more categories (a document cannot be in more than one category)

  E.g., spam filtering – binary decision: “spam” or “not spam”

- **Soft Classification**

  The process of assigning one or more category labels to a text document (a document may have more than one category)

  E.g., news filtering – which category to assign to news articles:

  - Sports, Olympics, Football (natural class)
  - Political, Business, Home,… (news sections)
  - Asian, Europe, Middle-East, …(geographical)

Some Examples of Text Classification

- Assigning subject categories to documents
- Email spam detection
- Medical diagnosis
- Identifying a language (before further processing)
- Identifying fraud (anomaly detection)
- Sentiment analysis (e.g., positive/negative reviews)
- Monitoring a news feed (e.g., news about 737-MAX 8)
- Etc.

HOW DOES AUTOMATIC TEXT CATEGORIZATION WORK?

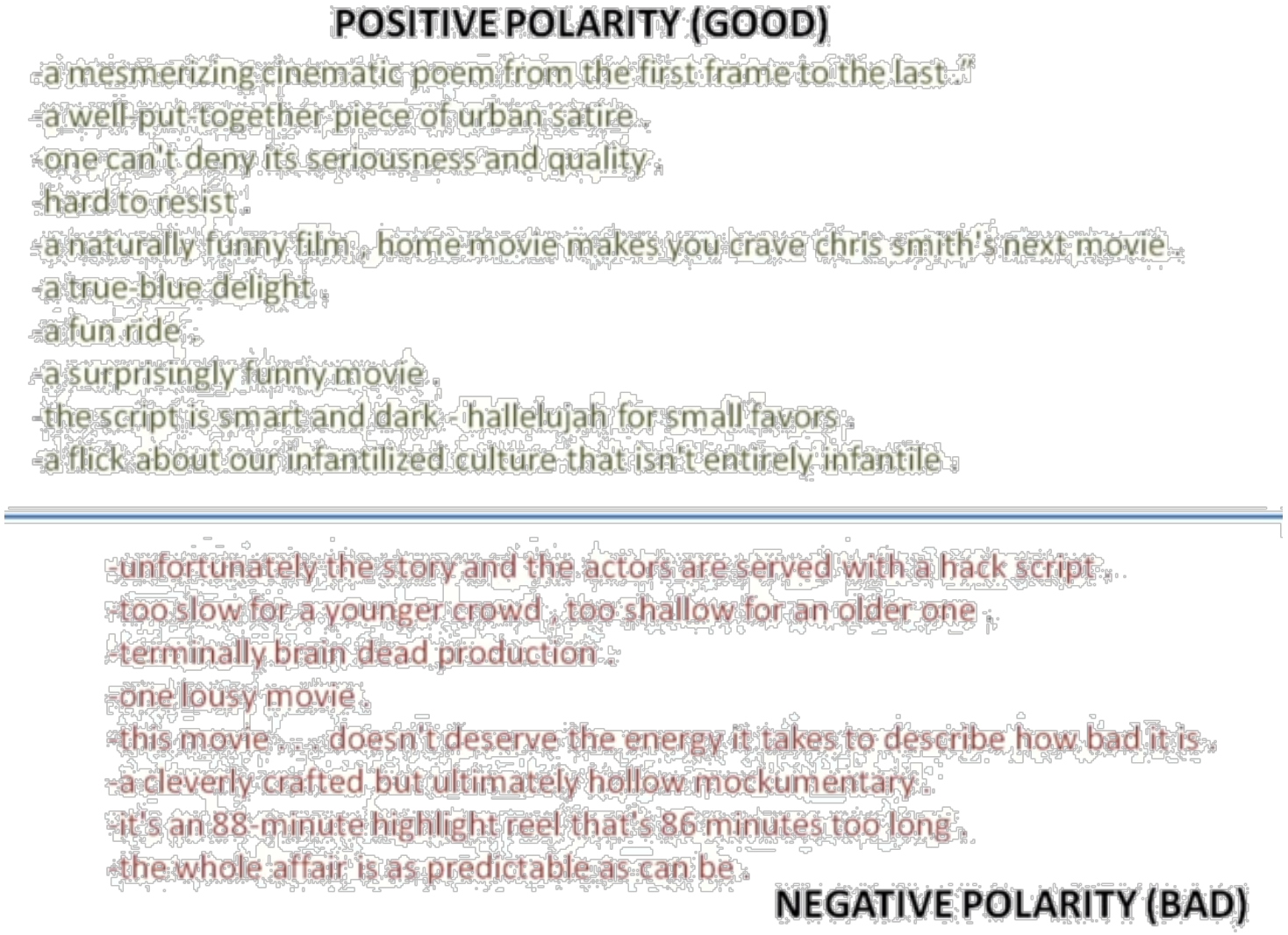
Text Categorization Phases

Two Phases for supervised method

1. **Training – creating the text "classifier" (automatic categorization engine)**
   - You need a set of documents, already categorized
   - Divide the set into training (typically 70%) and testing (30%)
   - Train your classifier such that it's able to accurately classify the training set of documents to your level of comfort
     - "level of comfort" depends on how hard is the task! ☺
   - Evaluate your classifier on the test set; ensure sufficient accuracy
2. **Running – using your classifier on new sets of documents**
   - You will not know how well it performs
   - Need to "audit" the results occasionally (use an assessor)
     - Assess random sample of the documents against the predicted categories

DOCUMENT DATA SET

Movie reviews classified as “good” and “bad”



From: http://karpathy.ca/mlsite/lecture2.php

Movie reviews classified as "good" and "bad"

POSITIVE POLARITY (GOOD)

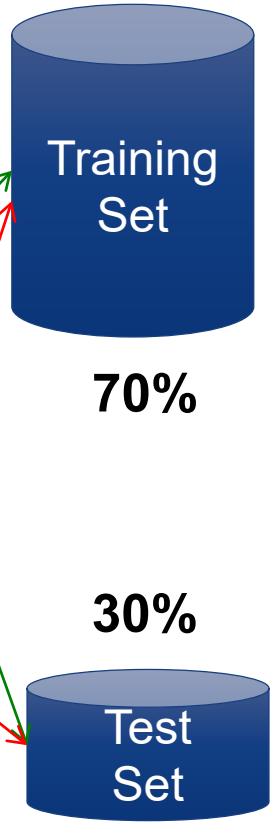
- a mesmerizing cinematic poem from the first frame to the last .
- a well-put-together piece of urban satire .
- one can't deny its seriousness and quality .
- hard to resist .
- a naturally funny film , home movie makes you crave chris smith's next movie .
- a true-blue delight .
- a fun ride .
- a surprisingly funny movie .
- the script is smart and dark - hallelujah for small favors .
- a flick about our infantilized culture that isn't entirely infantile .

5000 reviews

- unfortunately the story and the actors are served with a hack script .
- too slow for a younger crowd , too shallow for an older one .
- terminally brain dead production .
- one lousy movie .
- this movie . . . doesn't deserve the energy it takes to describe how bad it is .
- a cleverly crafted but ultimately hollow mockumentary .
- it's an 88-minute highlight reel that's 86 minutes too long .
- the whole affair is as predictable as can be .

5000 reviews

NEGATIVE POLARITY (BAD)



From: http://karpathy.ca/mlsite/lecture2.php

BUILDING A CLASSIFIER

JUST SOME EXAMPLES (NOT EXHAUSTIVE)

Creating classifiers

- **Hand-coded classifiers (the "good old days!")**
  - If <conditions> then <category> else NOT<category>, where conditions are normally in disjunctive normal form

If	((wheat & farm)	**or**	
	(wheat & commodity)	**or**	
	(bushels & export)	**or**	
	(wheat & tonnes)	**or**	
	(wheat & winter & ¬soft))	**then**	WHEAT else ¬ WHEAT

From: F. Aiolli, *Text Categorization*, http://www.math.unipd.it/~aiolli/corsi/SI-0607/Lez09.251006.pdf

Inducing classifiers (1)

Doc A belong to Cat A =
(Doc X belong to Cat A) (Cat A contains Doc A) / (Doc X contains Doc A)

di is Doc A
cj is Cat A

- **Probabilistic Classifiers**
  - Represent the probability that a document d_i belongs to category c_j by

$$P(c_j|d_i) = P(c_j)P(d_i|c_j)/P(d_i)$$

where $P(c_j)$ is the probability that a randomly selected document belongs to category c_j, $P(d_i)$ is the probability that a randomly selected document has d_i as a vector representation and $P(d_i|c_j)$ is the probability that category c_j contains d_i.

Naïve Bayes Model

Probabilistic Classifiers

Represent the probability that a document d_i belongs to category c_j by

$$P(c_j|d_i) = P(c_j)P(d_i|c_j) / \cancel{P(d_i)}$$

Doc A belong to Cat A =
(Doc X belong to Cat A) (Cat A contains Doc A) / (Doc X contains Doc A)

di is Doc A
cj is Cat A

$$class_{MAP} \approx \boldsymbol{argmax}_{class \in C} P(doc|class) * \mathrm{P}(class)$$

$$\approx \boldsymbol{argmax}_{class \in C} P(w_1, w_2, \ldots w_n|class) * \mathrm{P}(class)$$

$$\approx \boldsymbol{argmax}_{class \in C} P(w_1|class) * P(w_2|class) * \cdots * P(w_n|class) * \mathrm{P}(class)$$

Naïve Bayes Model

Doc A belong to Cat A =
(Doc X belong to Cat A) (Cat A contains Doc A) / (Doc X contains Doc A)

di is Doc A
cj is Cat A

$$class_{MAP} \approx \boldsymbol{argmax}_{class \in C}\, P(doc|class) * \mathrm{P}(class)$$
$$\approx \boldsymbol{argmax}_{class \in C}\, P(w_1, w_2, \ldots w_n|class) * \mathrm{P}(class)$$
$$\approx \boldsymbol{argmax}_{class \in C}\, P(w_1|class) * P(w_2|class) * \cdots * P(w_n|class) * \mathrm{P}(class)$$

- **Example**
  - For sentiment detection problem: $\boldsymbol{class} \in \{+, -\}$
  - **Training**: calculate and keep all the likelihood of vocabulary words *wrt.* classes: $P(w|-)$, $P(w|+)$, $\mathrm{P}(+)$, $\mathrm{P}(-)$

$P(w|-)$ = *count (neg docs having w) / count (neg docs)*

$P(-)$ = ***count (neg doc) / count(total docs)***

  - **Testing**: calculate, compare and select the class offering bigger result

$$P(w_1|-) * P(w_2|-) * \cdots * P(w_n|-) * \mathrm{P}(-)$$
$$P(w_1|+) * P(w_2|+) * \cdots * P(w_n|+) * \mathrm{P}(+)$$

Naïve Bayes Model

$$class_{MAP} \approx \boldsymbol{argmax}_{class \in C}\, P(doc|class) * \mathrm{P}(class)$$
$$\approx \boldsymbol{argmax}_{class \in C}\, P(w_1, w_2, \ldots w_n|class) * \mathrm{P}(class)$$
$$\approx \boldsymbol{argmax}_{class \in C}\, P(w_1|class) * P(w_2|class) * \cdots * P(w_n|class) * \mathrm{P}(class)$$

- **Example**
  - For sentiment detection problem: $\boldsymbol{class} \in \{+, -\}$
  - **Training**: calculate and keep all the likelihood of vocabulary words *wrt.* classes: $P(w|-)$, $P(w|+)$, $\mathrm{P}(+)$, $\mathrm{P}(-)$
  - **Testing**: calculate, compare and select the class offering bigger result

$$P(w_1|-) * P(w_2|-) * \cdots * P(w_n|-) * \mathrm{P}(-)$$
$$P(w_1|+) * P(w_2|+) * \cdots * P(w_n|+) * \mathrm{P}(+)$$

$$Dnew = "I\ hated\ the\ poor\ acting"$$

$$P(+|D_{new}) = P(I|+) * P(hated|+) * \sout{P(the|+)} * P(poor|+) * P(acting|+) * P(+)$$
$$= \mathrm{a}$$

$$P(-|D_{new}) = P(I|-) * P(hated|-) * \sout{P(the|-)} * P(poor|-) * P(acting|-) * P(-)$$
$$= \mathrm{b}$$

Inducing classifiers (1)

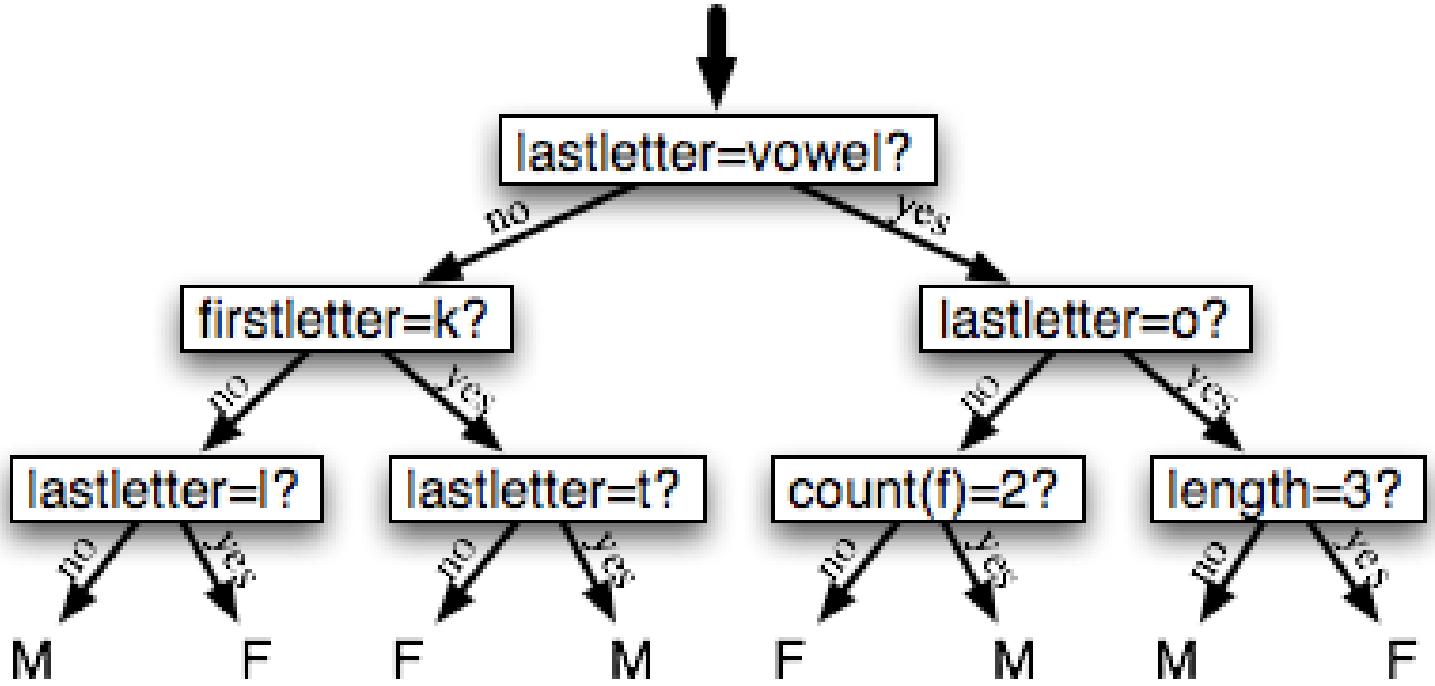
- **Decision Tree Classifiers**
  - Uses symbolic representation in a tree network
  - Internal nodes are terms, edges are tests on the weights, and leaf nodes are categories
  - The classifier works by traversing a path to the appropriate leaf node

Exercise

- **List of boys names**
  - Alan
  - Barry
  - Colin
  - Dexter
  - Edward
  - Frederick
  - Howard
  - ….
- **List of girls names**
  - Anna
  - Betty
  - Chelsea
  - Doris
  - Elizabeth
  - Fanny
  - Hortense
  - ….

Given a list of boys' names and a list of girls' names, build a simple classifier to distinguish boys' names from girls' names

Example of a decision tree to decide if a name is male or female



From: http://nltk.googlecode.com/svn/trunk/doc/book/ch06.html

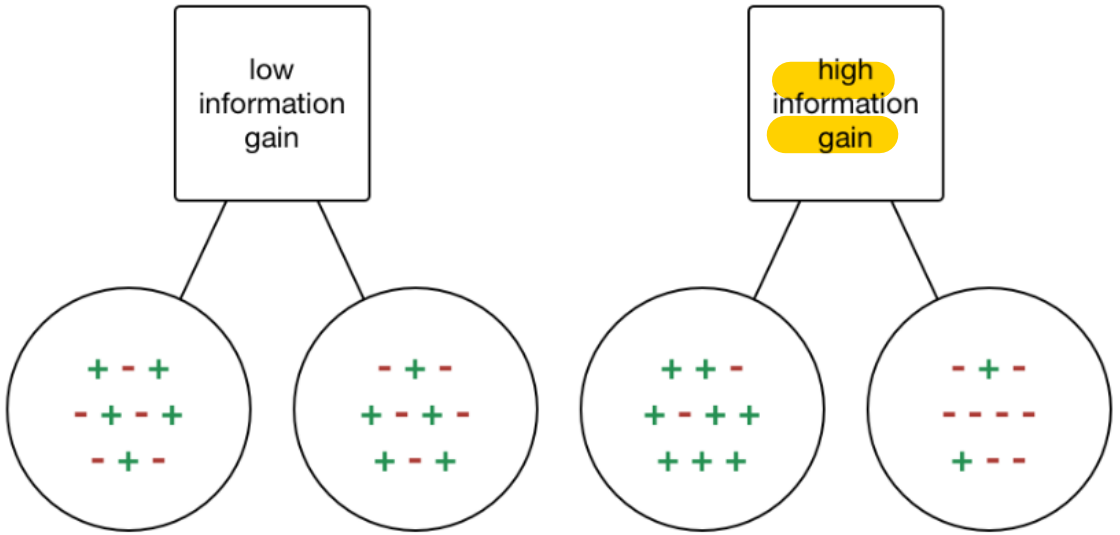
Information Gain

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N} I(D_{left}) - \frac{N_{right}}{N} I(D_{right})$$

f: feature split on
D_p: dataset of the parent node
D_{left}: dataset of the left child node
D_{right}: dataset of the right child node
I: impurity criterion (Gini Index or Entropy)
N: total number of samples
N_{left}: number of samples at left child node
N_{right}: number of samples at right child node

$$\mathrm{H}(T) = \mathrm{I}_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

https://towardsdatascience.com/entropy-and-information-gain-in-decision-trees-c7db67a3a293



Information Gain

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N} I(D_{left}) - \frac{N_{right}}{N} I(D_{right})$$

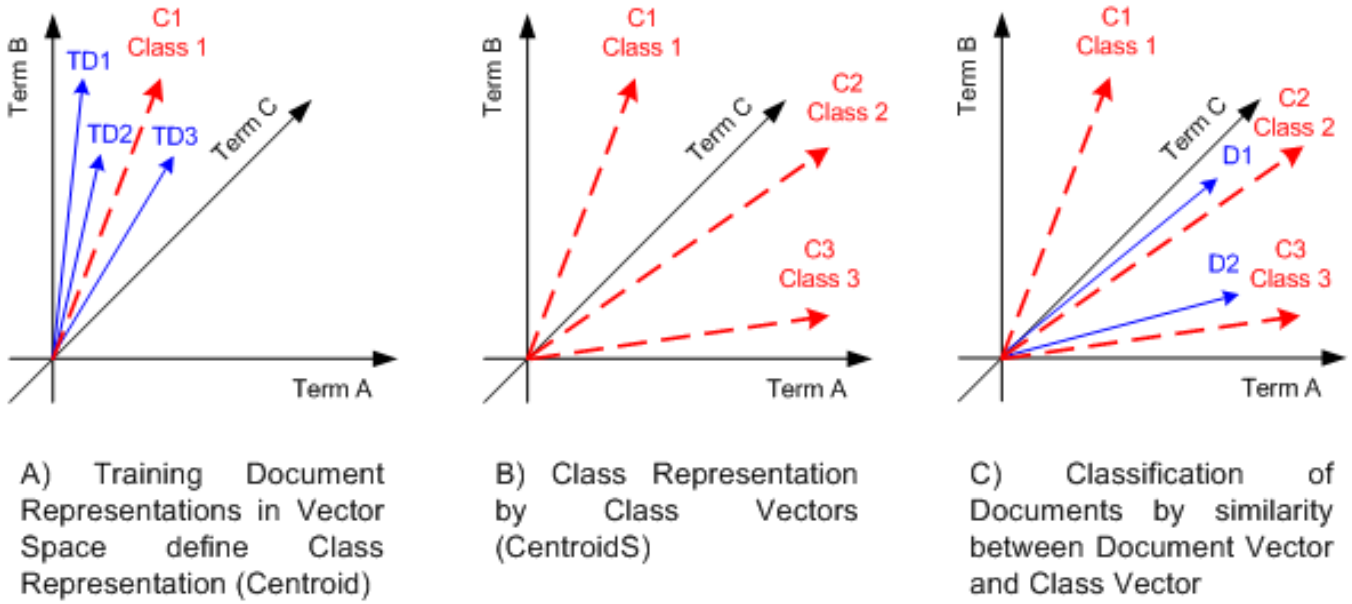
f: feature split on
D_p: dataset of the parent node
D_{left}: dataset of the left child node
D_{right}: dataset of the right child node
I: impurity criterion (Gini Index or Entropy)
N: total number of samples
N_{left}: number of samples at left child node
N_{right}: number of samples at right child node

Inducing classifiers (2)

- **The Rocchio Classifiers**
  - Each category is represented by a prototypical document, i.e., profile vector

denoting the first, original, or typical form of something.

  - Documents are classified by similarity to the profile vector



A) Training Document Representations in Vector Space define Class Representation (Centroid)

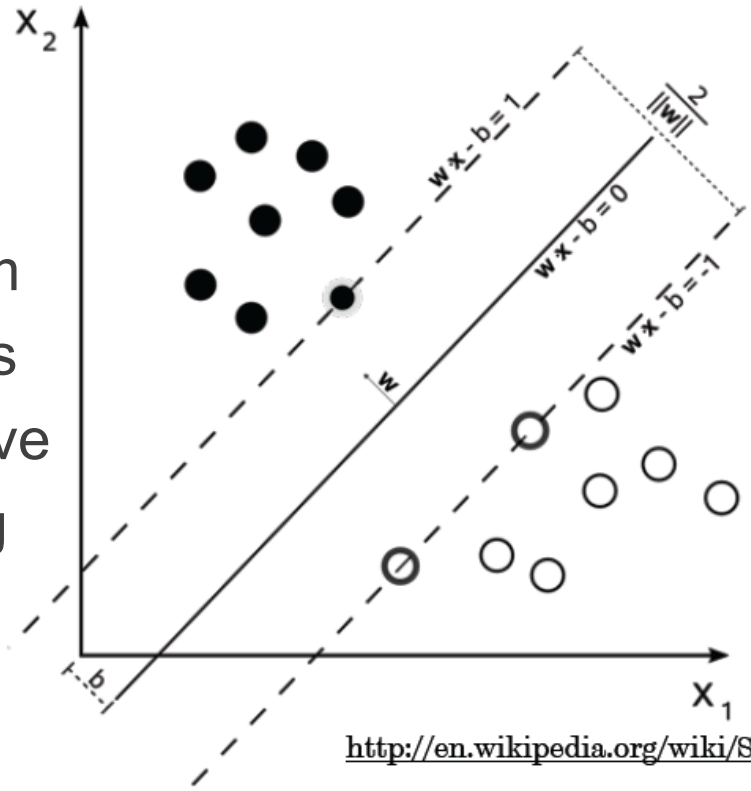
B) Class Representation by Class Vectors (CentroidS)

C) Classification of Documents by similarity between Document Vector and Class Vector

From: http://www.iicm.tugraz.at/about/Homepages/cguetl/courses/isr/opt/classification/Vector_Space_Model.html

Inducing classifiers (3)

- **Support Vector Machines (SVMs)**
  - SVMs divide the term space in hyperplanes separating the positive and negative training samples.
  - The surface that provides the widest separation between the support surfaces is selected



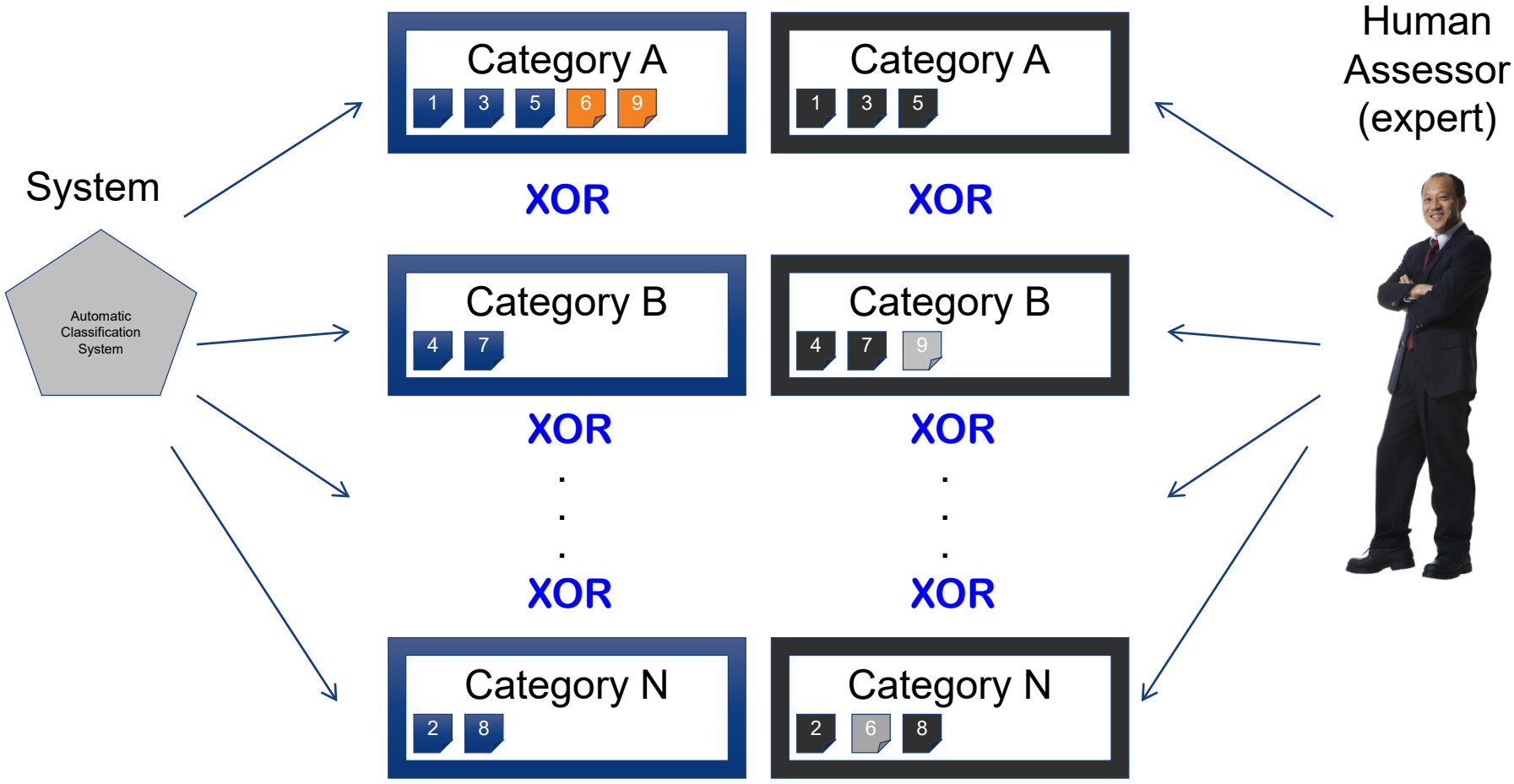
http://en.wikipedia.org/wiki/Support_vector_machines

EVALUATION

ACCURACY

EVALUATION

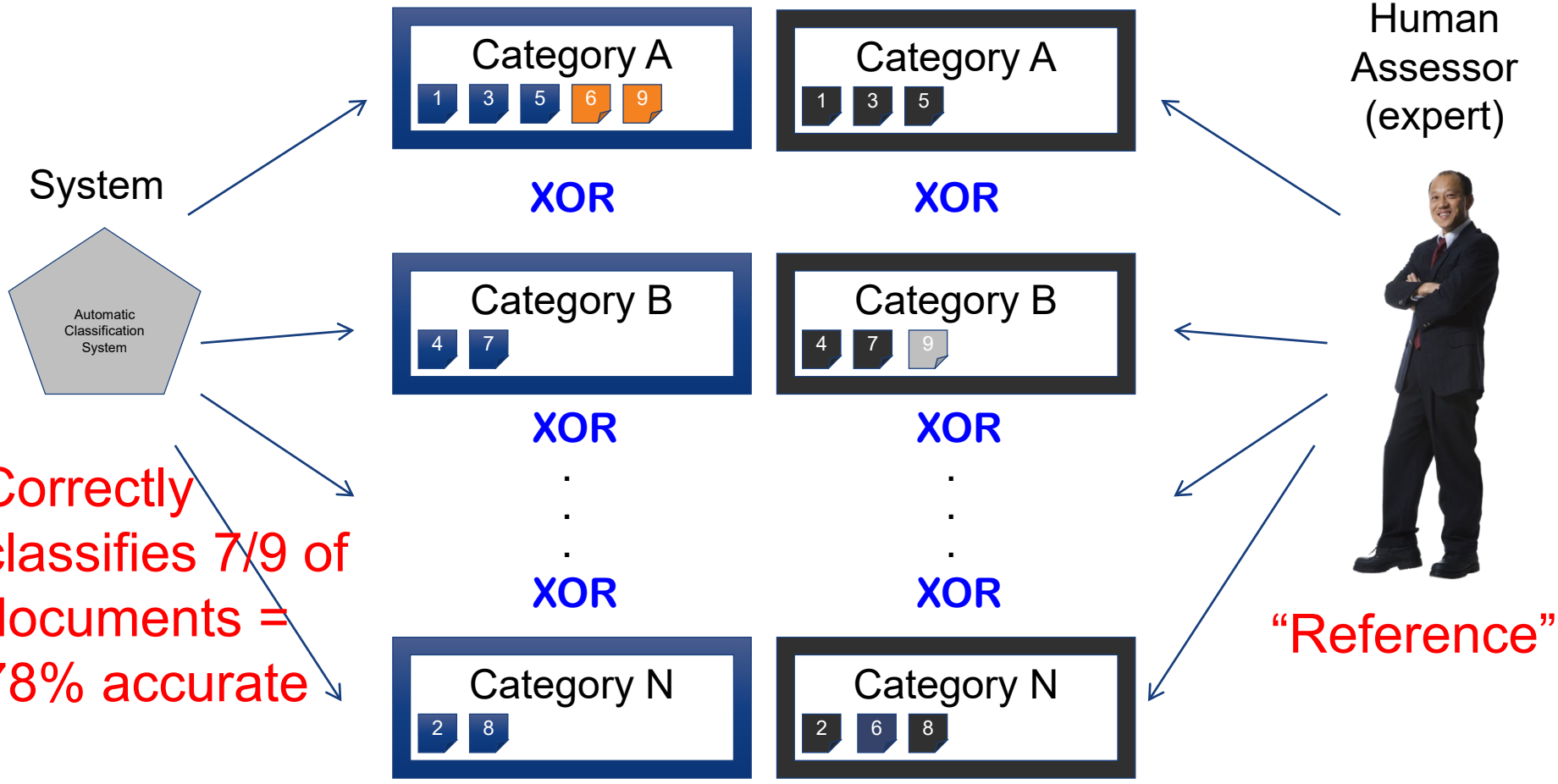
What is “accuracy”?



What do we mean by "Accuracy"

- **You measure an automatic categorization system by:**
  - How well it classifies a set of documents against a "reference"
  - This "reference" is normally a human expert
- **Reference**
  - Gold standard – accepted as being the best available
    - May not be perfect, e.g., tumour board for oncology
  - Good enough
    - Human expert(s), typically 80% agreement, good methodology
  - Better than nothing
    - "your boss tells you to do this, so you recruit your friends, family,…"
- **Most of the time, no such thing as "absolute truth"**

One measure of accuracy



Discussion

- **Weather prediction (system predicts one week in advance)**

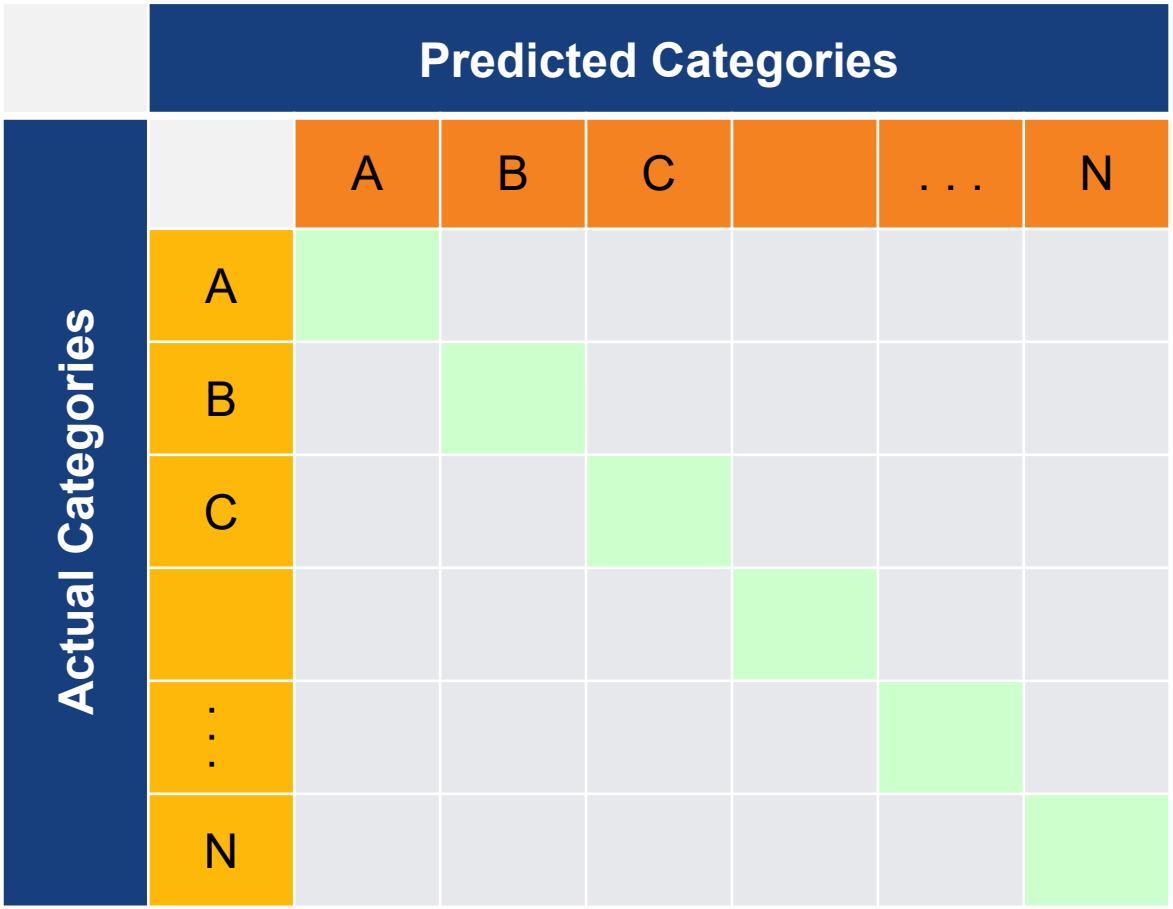
	Mon	Tue	Wed	Thu	Fri	Sat	Sun
System	Sunny	Drizzle	Rain	Sunny	Cloudy	Thunderstorms	Sunny
Actual	Sunny	Rain	Cloudy	Sunny	Drizzle	Thunderstorms	Cloudy

- **Questions:**
  - How “accurate” is the weather prediction system?
  - Do you have a tolerance for error? +/- margin of error?
  - How about outcomes – will I get wet if I use the system to decide whether or not to carry an umbrella? Will I get angry?

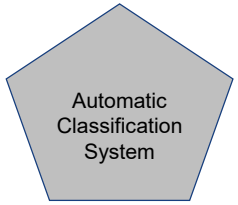
Discussion

- **There is a desert in USA where it only rains 10 days in a year. But when it rains, there are flash floods which kill an average of 20 people per year.**
- **You build a weather prediction system. How would you measure accuracy?**
  - If you said no rain every day, you would automatically be right 355/365 days in a year = 97% “accuracy”
  - If you said rain every day, you would be right only 10/365 = 3% of the time, but you would save 20 lives. Would this be true?
  - You are right half the time = 50%, so you would save 10 lives, but ONLY if people listen to your predictions.
- **What would you propose?**

Confusion Matrix



Example (using #)



		Predicted Categories						
			A	B	C		. . .	N	
Actual Categories	A		143	34	17		. . .	2	= Tot(A) docs
	B		67	1289	44		. . .	239	= Tot(B) docs
	C		980	234	3454		. . .	88	= Tot(C) docs
							. . .		
	⋮						. . .		
	N		87	24	63		. . .	650	= Tot(N) docs

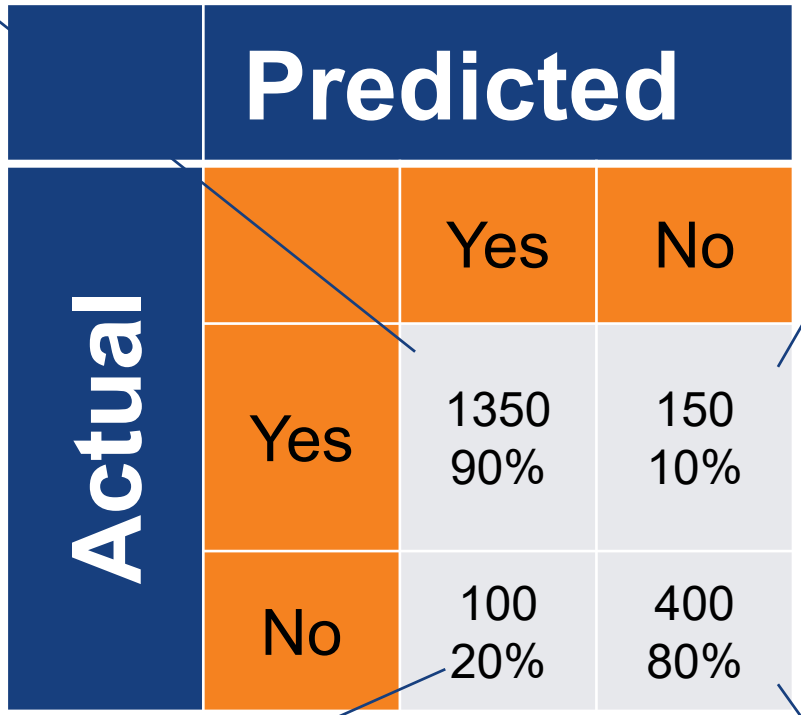
Example (using %)

Automatic Classification System

Actual Categories \ Predicted Categories	A	B	C		. . .	N	
A	87%	2%	5%		. . .	1%	= 100%
B	6%	90%	0%		. . .	2%	= 100%
C	12%	2%	77%		. . .	4%	= 100%
					. . .		
⋮					. . .		
N	21%	0%	4%		. . .	65%	= 100%

Consider the simple 2x2 matrix (2000 documents were classified)

Desired positive prediction



False negative

False positive

Desired negative prediction

EVALUATING MULTIPLE CLASSIFIERS

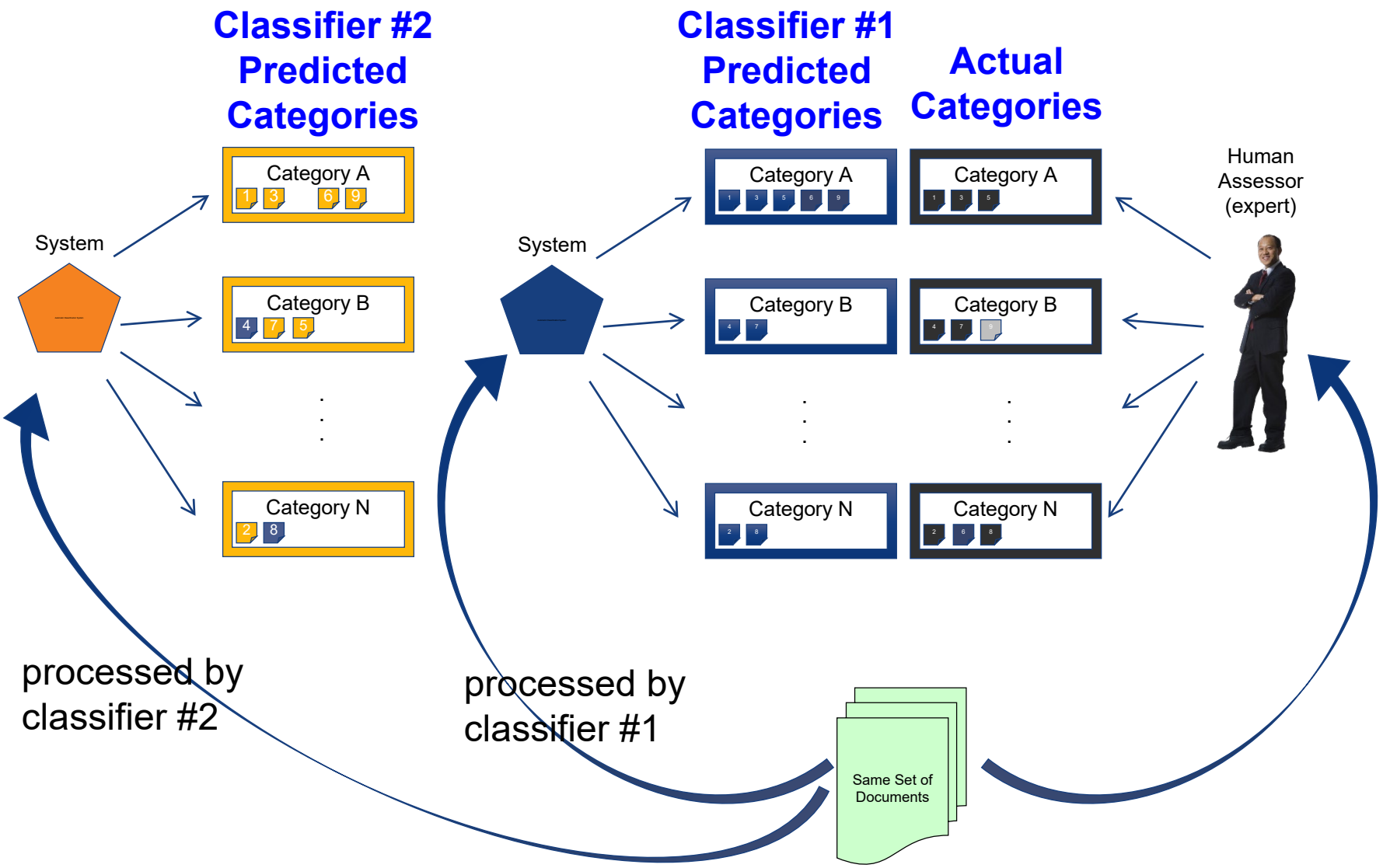
Discussion

- **Weather prediction. You ask two people to predict whether it will rain or not in the coming week:**

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
ZZ	Yes	No	Yes	No	Yes	No	No
MK	No	No	No	Yes	No	No	Yes
Actual	Yes	Yes	Yes	No	Yes	Yes	No

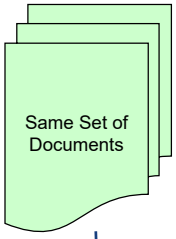
- **Questions:**
  - Who is more “accurate”? ZZ is right 5/7 times. MK is right 0/7 times.
  - Who should you ask in future if you don’t want to get wet?
  - Who is the better “classifier”?

What happens with 2 classifiers?



Comparing models: e.g. 2x2 matrix (actual numbers of documents)

Classifier #1



Classifier #2

		Predicted	
		Y	N
Actual	Y	900	100
	N	40	410

Seems quite good for both predictions

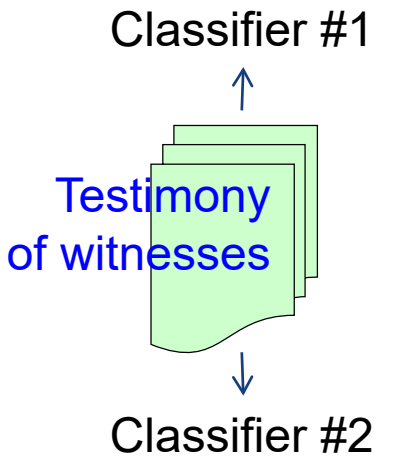
		Predicted	
		Y	N
Actual	Y	700	300
	N	2	448

Reduced the false positives but false negatives increased

? Which classifier is better?

Adding the semantics – the courtroom

Semantics is the study of meaning in language. It can be applied to entire texts or to single words. For example, "destination" and "last stop" technically mean the same thing, but students of semantics analyze their subtle shades of meaning.



		Predicted	
Actual		Guilty	Innocent
	Guilty	900	100
	Innocent	40	410

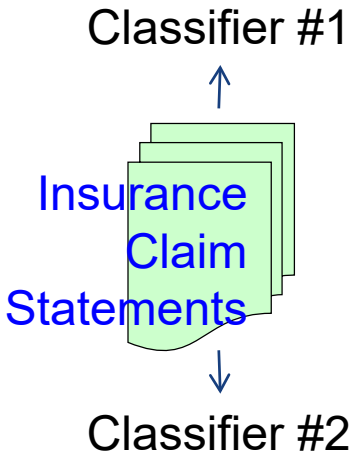
Let 100 guilty go free
Convict 40 innocent persons

		Predicted	
Actual		Guilty	Innocent
	Guilty	700	300
	Innocent	2	448

Let 300 guilty go free
Convict 2 innocent persons

? Which classifier is better?

Adding a cost function – fraud investigation



Actual	Predicted	
	Honest	Fraud
Honest	900	100
Fraud	40	410

Actual	Predicted	
	Honest	Fraud
Honest	700	300
Fraud	2	448

? Which classifier is better?

The average fraud costs the company $2000
It costs the company $500 to investigate each suspected fraud

Adding a cost function – fraud investigation

- The average fraud costs the company $2000
- It costs the company $500 to investigate each suspected fraud

Company loses $80k in fraud
Company pays $255k in costs

		Predicted	
		Honest	Fraud
Actual	Honest	900	100
	Fraud	40	410

Company loses $4k in fraud
Company pays $374k in costs

		Predicted	
		Honest	Fraud
Actual	Honest	700	300
	Fraud	2	448

- ***Consider Doing nothing (don't act to identify fraud):***
  - *Predicted fraud = 0 cases @$500 per case costs $0k for investigation.*
  - *Undetected fraud is 450 cases @$2k/fraud loses $900k.*
  - *Overall -$0k -$900k = -$900k*
- ***Analysis for classifier #1:***
  - *Predicted fraud = 510 cases @$500 per case costs $255k for investigation.*
  - *Undetected fraud is 40 cases @$2k/fraud loses $80k.*
  - *Overall -$255k -$80k = -$335k*
- ***Analysis for classifier #2:***
  - *Predicted fraud = 748 cases @$500 per case costs $374k for investigation.*
  - *Undetected fraud is 2 cases @$2k/fraud loses $4k.*
  - *Overall -$374k -$4k = -$378k*

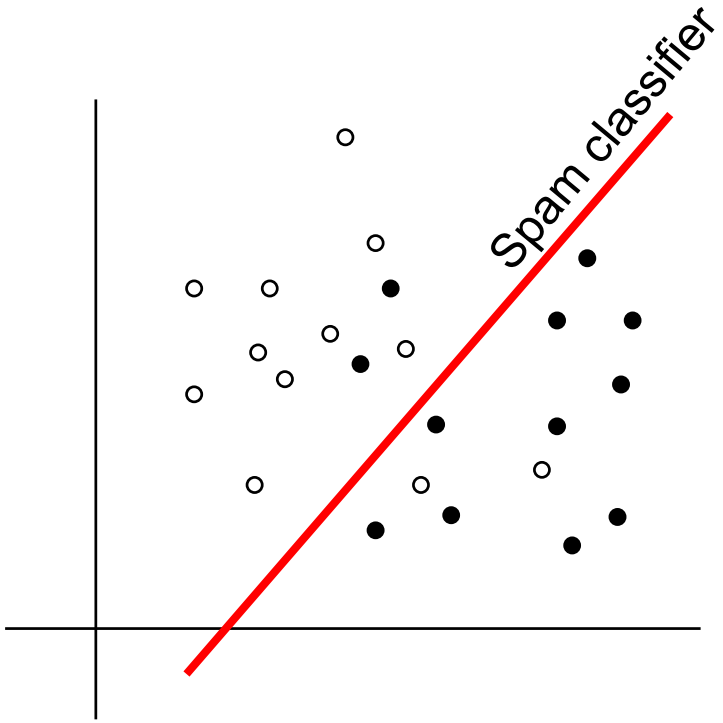
Classifier evaluation

- **Evaluation of classifiers is done with respect to a *business context***
- **Evaluation of classifiers is normally done *empirically***
- **Experimental evaluation focuses on *effectiveness*, i.e., the ability of the classifier to make the right classification decision**
- **Precision & Recall concepts as applied to (multi-class) categorization**
  - Precision is the probability that if a random document d_i is categorized under category c_j, that decision is correct
  - Recall wrt c_j is the probability that if a random document d_i should be categorized under c_j , then the decision is taken

Precision and recall are two extremely important model evaluation metrics. While precision refers to the percentage of your results which are relevant, recall refers to the percentage of total relevant results correctly classified by your algorithm.

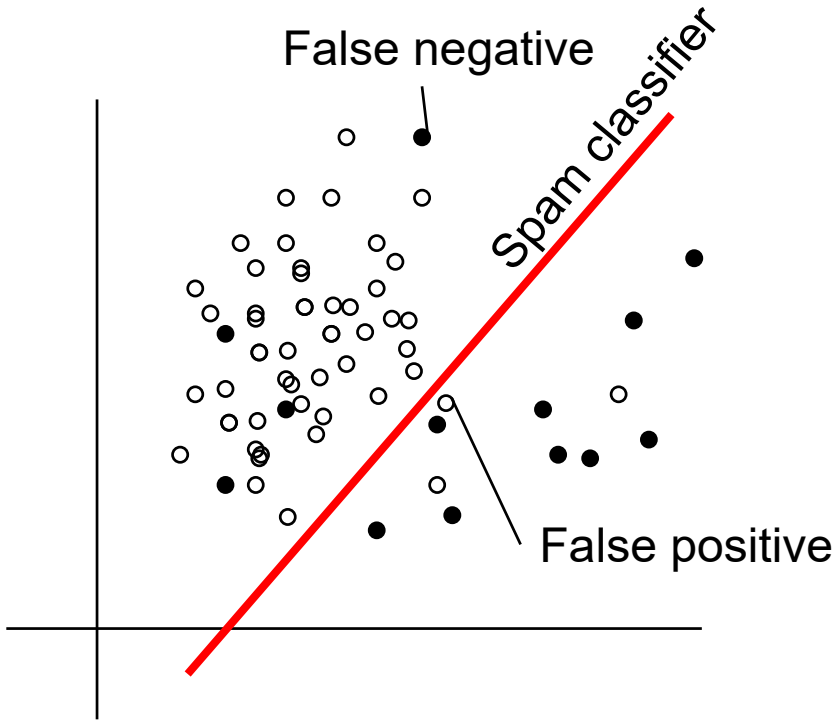
RUNNING THE CLASSIFIER

Expect False Results
eg: spam filtering



- Email data – non-spam
- Email data – spam

Training Set



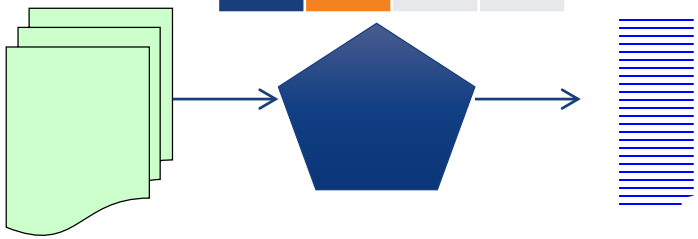
- Email non-spam
- Email spam

Real email stream

Overfitting the Training Set

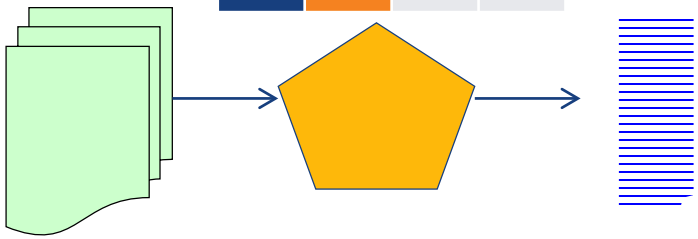
Training Set

Actual \ Predicted	Y	N
Y	700	0
N	0	300



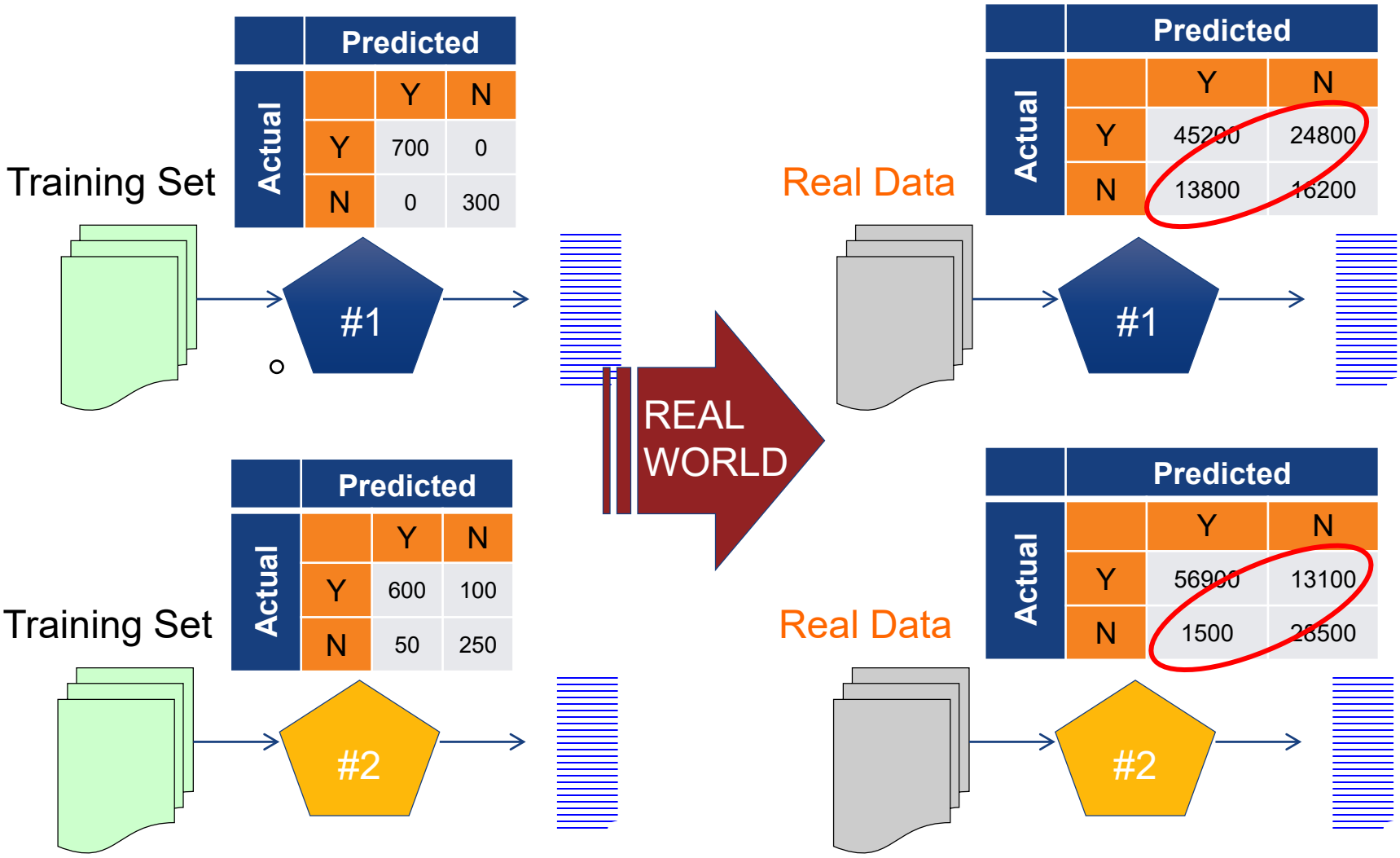
Training Set

Actual \ Predicted	Y	N
Y	600	100
N	50	250

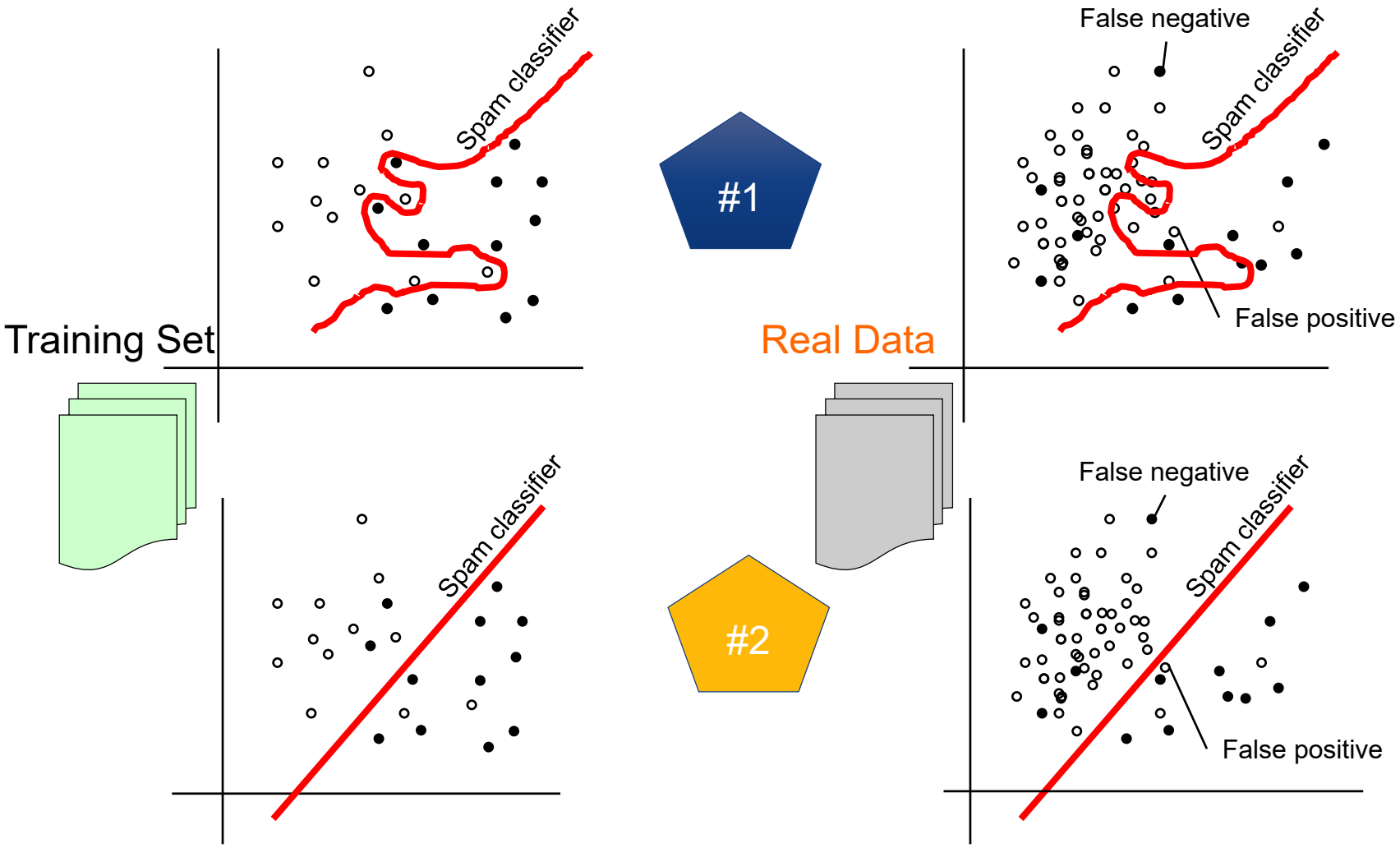


? Which classifier is better?

Overfitting the Training Set

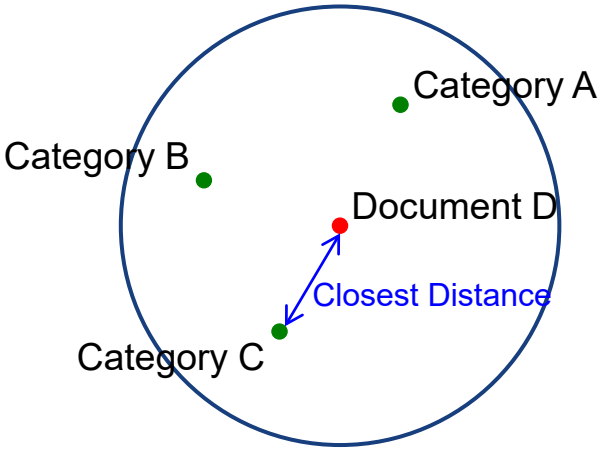


Overfitting the Training Set – what happened?



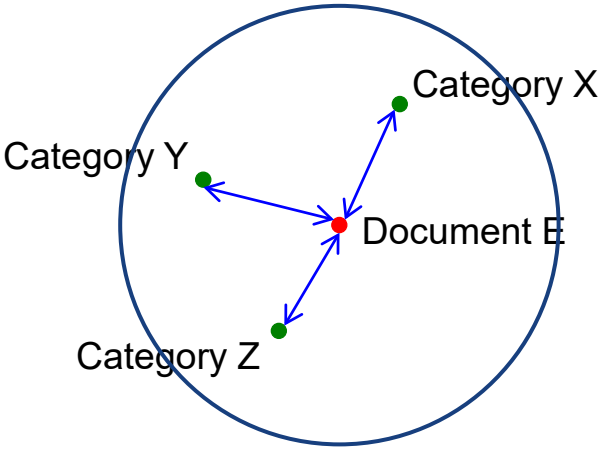
Hard and Soft Categorization

- **Fully automated classifiers make "hard" binary decisions**
  - In example to right, the document, D, is assigned to category C only.



- **Semi-automated (interactive) classifiers instead are created by allowing "soft" real-value decisions**
  - Rank the categories according to their measure of appropriateness for the document
  - In example to right, the document, E, is assigned 3 possible categories:

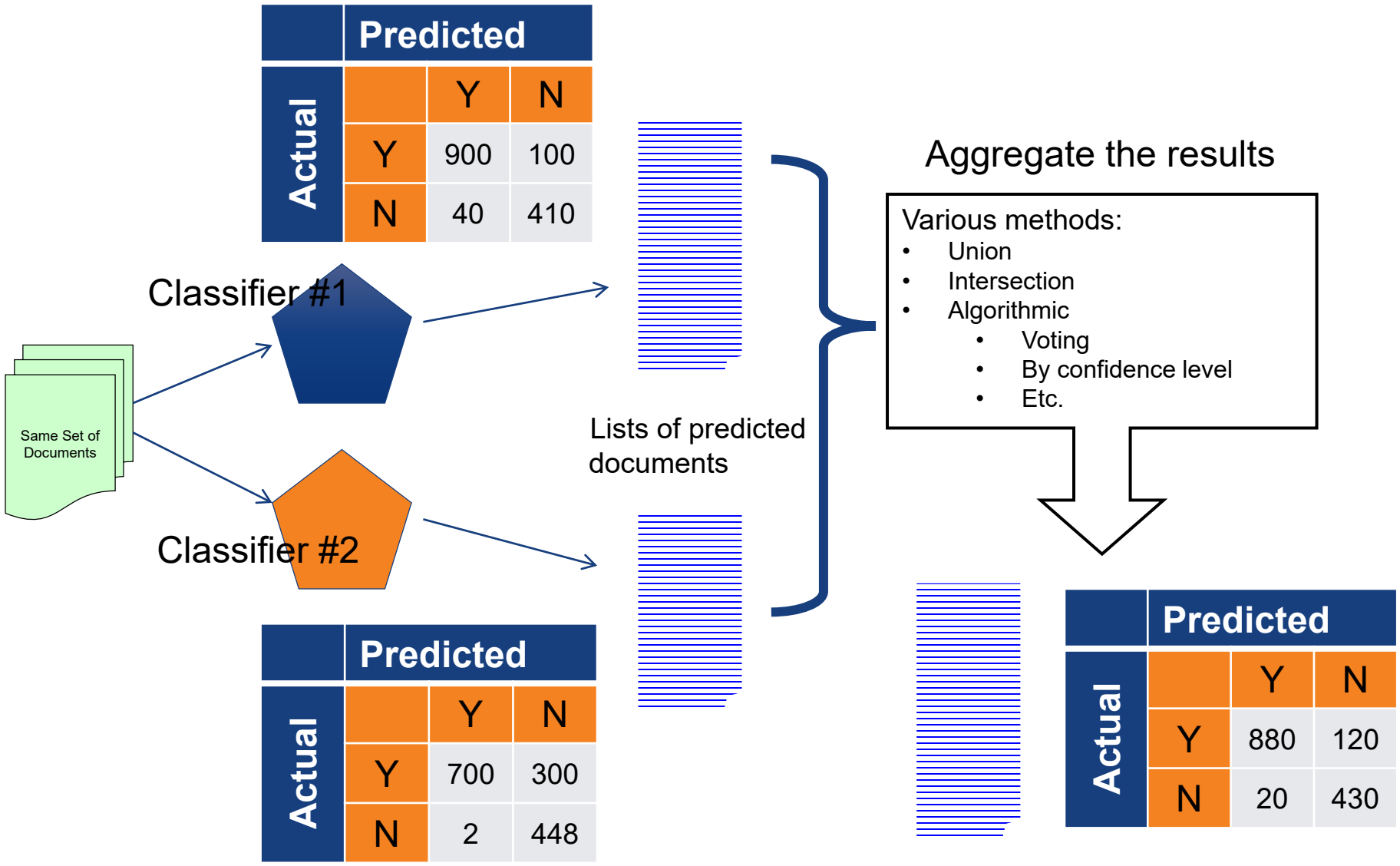
Rank	Category	Probability
1	Z	0.76
2	X	0.72
3	Y	0.54



  - Used for computer assisted human decision making
    - For example, in critical applications such as medical diagnosis

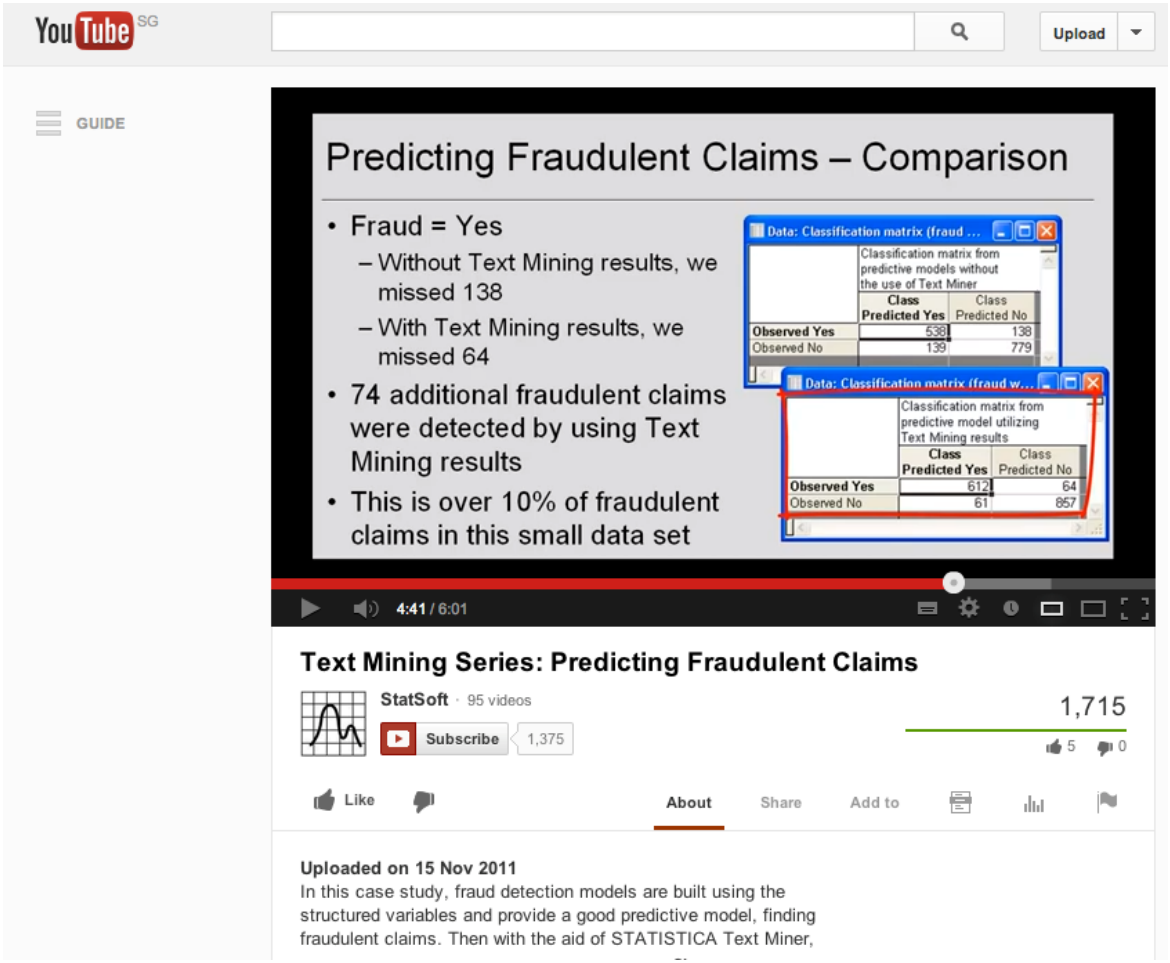
Aggregating multiple classifiers

Running with more than one classifier



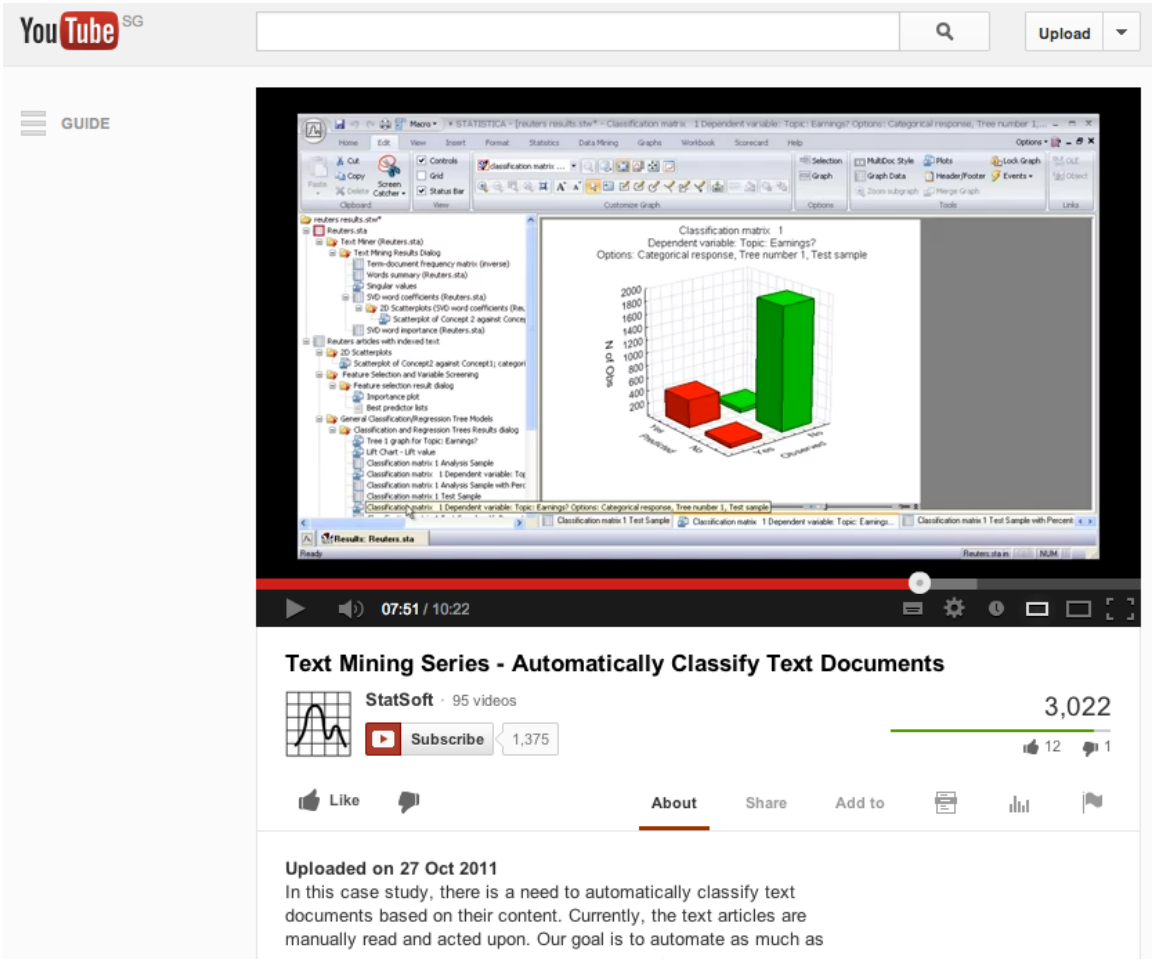
TEXT CATEGORIZATION APPLICATION EXAMPLES

Boosting Identification of Fraudulent Claims



From: http://www.youtube.com/watch?v=OlQpm8qTog4

Automatic Categorization of Documents



From: http://www.youtube.com/watch?v=Q5K3gyQJkC0

UNSUPERVISED TEXT CATEGORIZATION

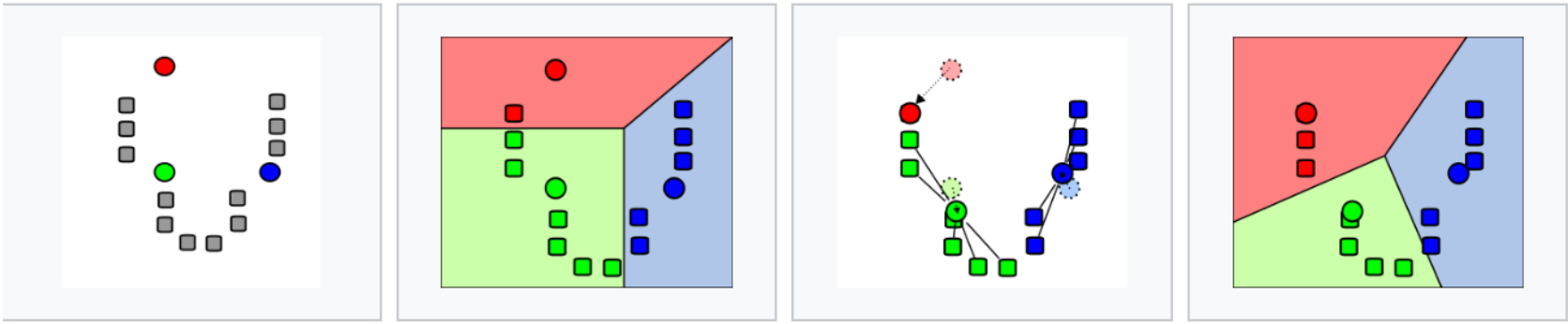
DOCUMENT CLUSTERING

What is text clustering?

- **Clustering is the task of grouping a set of documents in such a way that the documents in each group are more "similar" to each other than to documents in other groups.**
- **Clustering lets you explore your data**
  - Many tools are interactive
- **You can understand your data better, e.g.:**
  - What groupings exist in your data?
  - How many are there? How big is each group?
  - What are the common terms?
  - Are there anomalies?

Clustering Example

Demonstration of the standard algorithm



1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

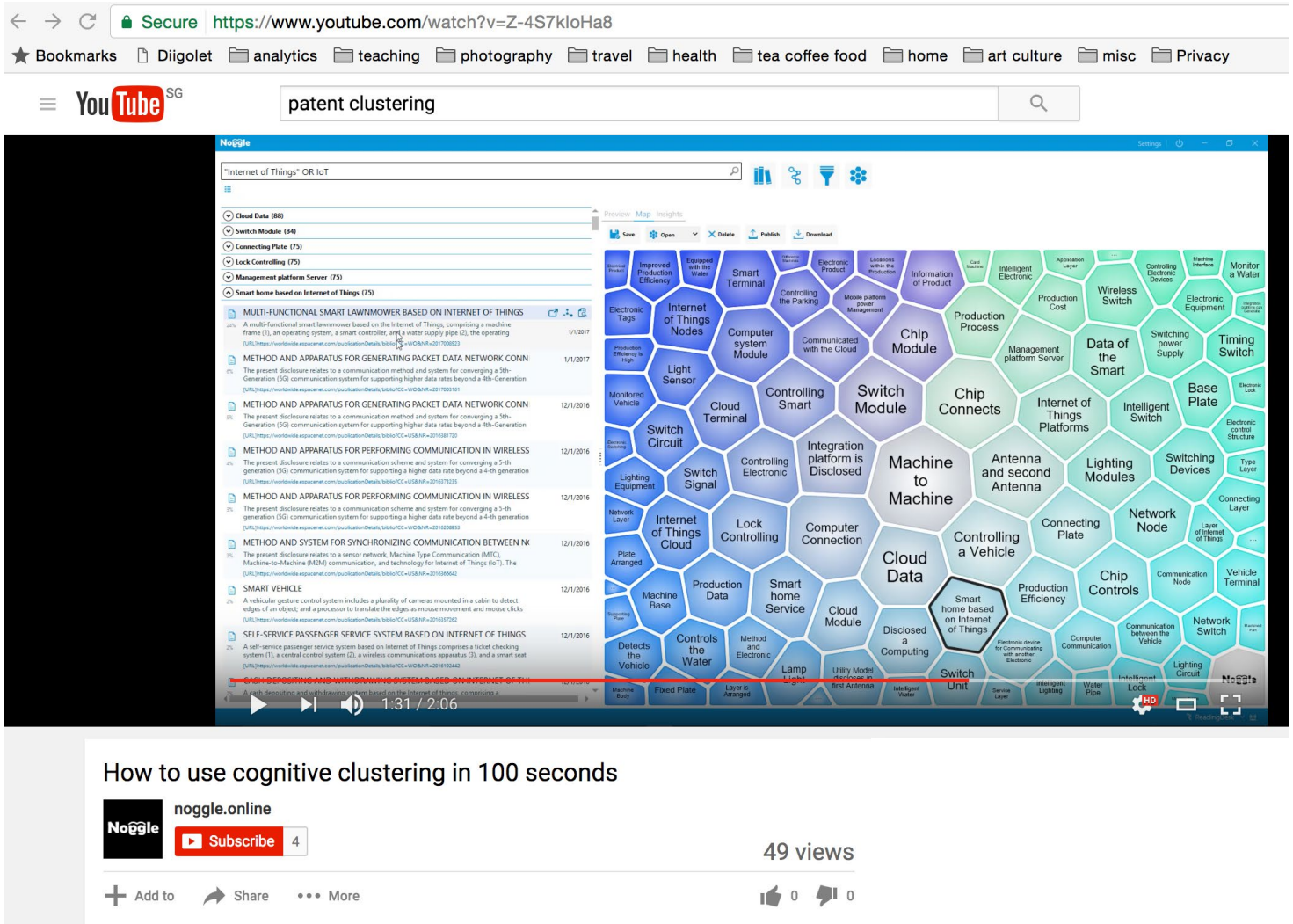
3. The centroid of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

In mathematics, a Voronoi diagram is a partition of a plane into regions close to each of a given set of objects.

From: https://www.youtube.com/watch?v=CHlrx4gsoJI

Patent Clustering



From: https://www.youtube.com/watch?v=Z-4S7kloHa8

Notes about Clustering

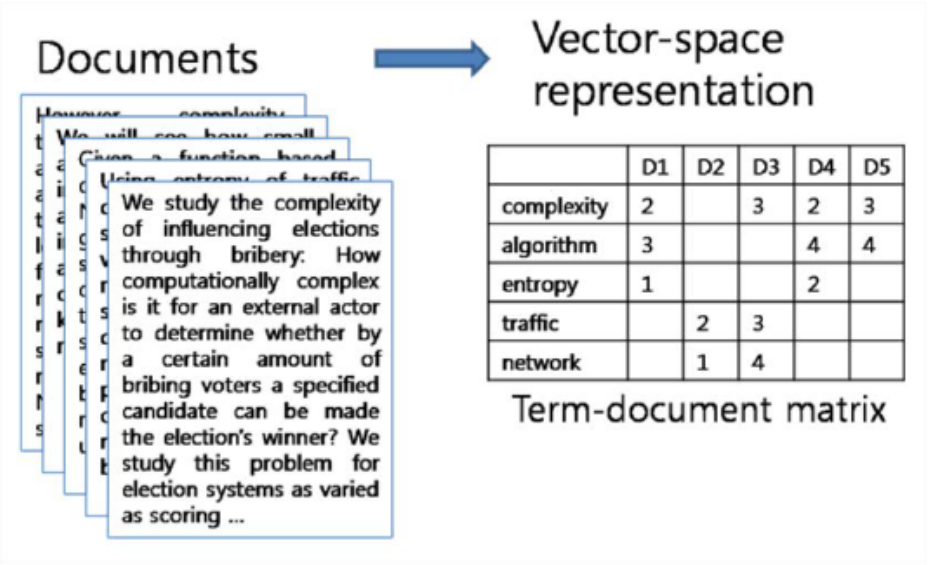
- **Your clusters may surprise you**
  - Documents tend to fall into natural classes (clusters)
  - There will be some surprising ones (worth drilling down!)
- **You can control the number of clusters (depends on the algorithm)**
  - You don’t want too many clusters (overfit!)
  - You don’t want too few clusters (meaningless)
  - **Clusters should lead to fulfilling business outcomes**
- **You don’t need training phase to create clusters**
  - Clustering can be language independent (but monolingual)

DIMENSIONAL REDUCTION

Dimensional Reduction

- Sparsity
- High dimension
- SVD single Valude Decomposition
- Low dimension

Document Clustering

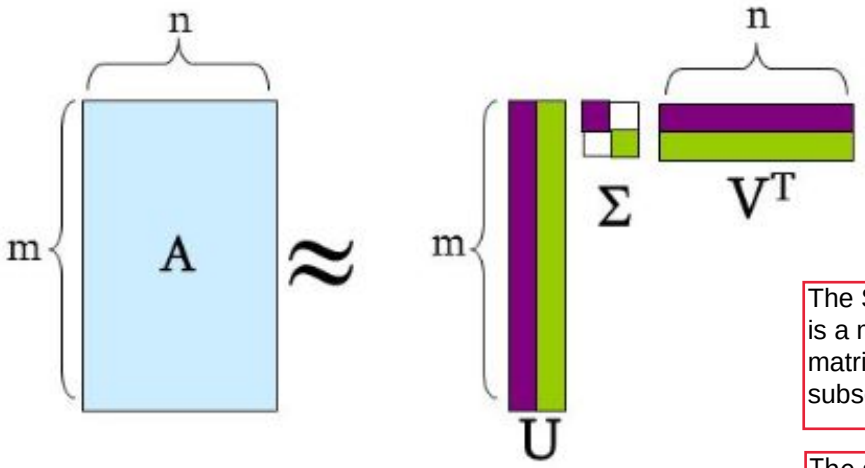


	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

	T1	T2	T3	T4	T5	T6	T7	T8
Doc1	2	0	4	3	0	1	0	2
Doc2	0	2	4	0	2	3	0	0
Doc3	4	0	1	3	0	1	0	1
Doc4	0	1	0	2	0	0	1	0
Doc5	0	0	2	0	0	4	0	0
Doc6	1	1	0	2	0	1	1	3
Doc7	2	1	3	4	0	2	0	2

Singular Value Decomposition

$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^T$$



https://www.youtube.com/watch?v=P5mlg91as1c&list=RDQM6QrFQnObRk8&start_radio=1&ab_channel=ArtificialIntelligence-AllinOne

The Singular-Value Decomposition, or SVD for short, is a matrix decomposition method for reducing a matrix to its constituent parts in order to make certain subsequent matrix calculations simpler.

The singular values are non-negative real numbers, usually listed in decreasing order (s1(T), s2(T), ...).

- **U,V**
  - Columns are orthogonal and unit vectors
- **Σ**
  - Entries (singular values) are positive and sorted in decreasing order of importance

Singular Value Decomposition

Original Matrix

	document	error	invalid	message	file	format	unable	to	open	using	path	variable
1	d1	1	1	1	1	1	0	0	0	0	0	0
2	d2	1	0	2	1	0	1	1	1	1	1	0
3	d3	1	0	0	0	1	1	1	0	0	0	1

(3,11)

≈

	document	SVD1	SVD2
1	d1	1.63	.49
2	d2	3.14	-.96
3	d3	1.35	1.64

(3,N)

×

Decreasing values

Sorted Singular Values

12.29		
	6.2	
		...

(N,N)

×

T

Weights

	U_2	
error	43	.30
invalid	.11	.13
message	.55	-.37
file	.33	-.12
format	.21	.55
unable	.31	.18
to	.31	.18
open	.22	-.25
using	.22	-.25
path	.22	-.25
variable	.09	.42

(11,N)^T

- **Top N=2 dimensions**

Singular Value Decomposition

Original Matrix

	document	error	invalid	message	file	format	unable	to	open	using	path	variable
1	d1	1	1	1	1	1	0	0	0	0	0	0
2	d2	1	0	2	1	0	1	1	1	1	1	0
3	d3	1	0	0	0	1	1	1	0	0	0	1

- **New Matrix**

	document	SVD1	SVD2
1	d1	1.63	.49
2	d2	3.14	-.96
3	d3	1.35	1.64

[3,N]

×

Sorted Singular Values

12.29		
	6.2	
		...

[N,N]

[3,N]

- **Dimensions reduced from 11 to N=2**

Dimension Reduction

Original Matrix

	document	error	invalid	message	file	format	unable	to	open	using	path	variable
1	d1	1	1	1	1	1	0	0	0	0	0	0
2	d2	1	0	2	1	0	1	1	1	1	1	0
3	d3	1	0	0	0	1	1	1	0	0	0	1

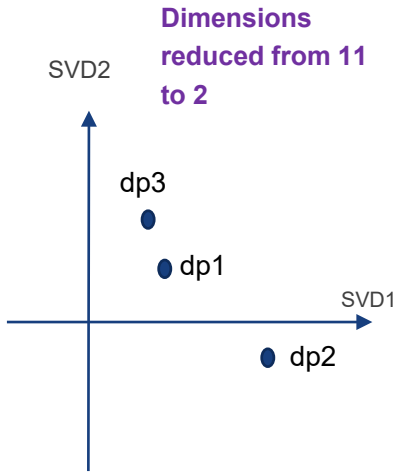
Step1. Apply SVD/PCA

You get SVDs/Concepts.

	document	SVD1	SVD2
1	d1	1.63	.49
2	d2	3.14	-.96
3	d3	1.35	1.64

Sorted Singular Values

12.29	
	6.2



DataPoint1 = [1,1,1,1,1,0,0,0,0,0,0]

DataPoint2 = [1,1,2,1,0,1,1,1,1,1,0]

DataPoint3 = [1,0,0,0,1,1,1,0,0,0,1]

≈

Datapoint1 = [20.1, 3.0]

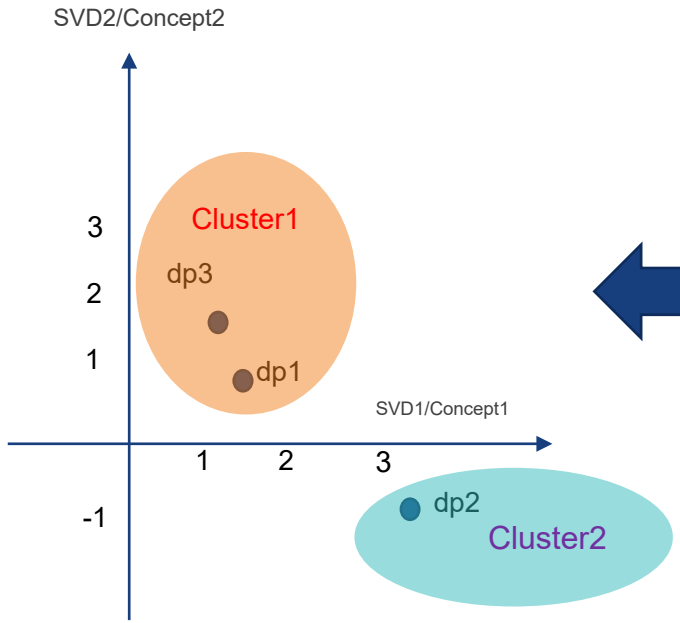
Datapoint2 = [38.6,-5.95]

Datapoint3 = [16.6, 10.2]

Concept#

/SVD#

≠ **Cluster#**

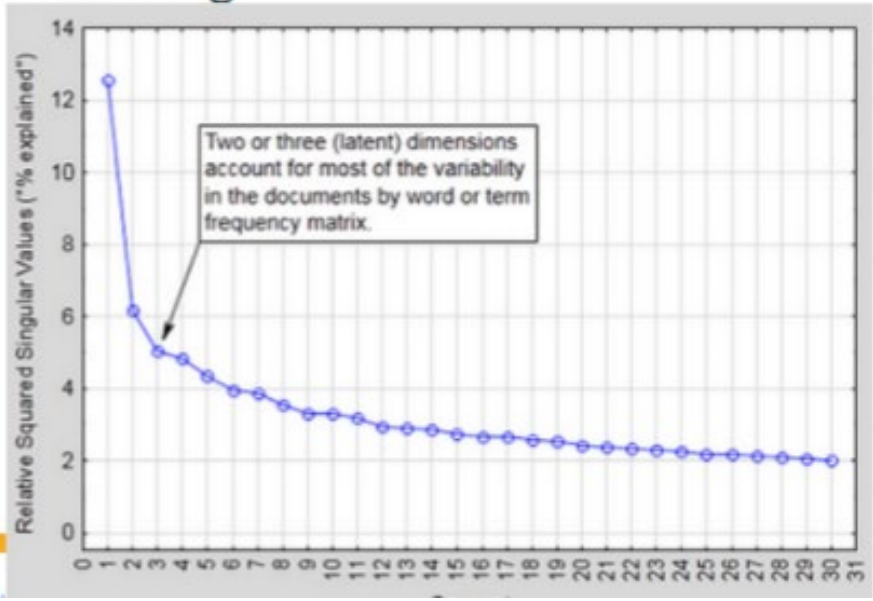


Step 2. Apply KM Or other classifiers

Singular Value Decomposition

SVD – How Many Dimensions?

- Usually no more than 5 to 20 dimensions extract most of the information from the TDM.
- More dimensions (up to a few hundred) can be retained if the processed data is for subsequent predictive modeling or clustering



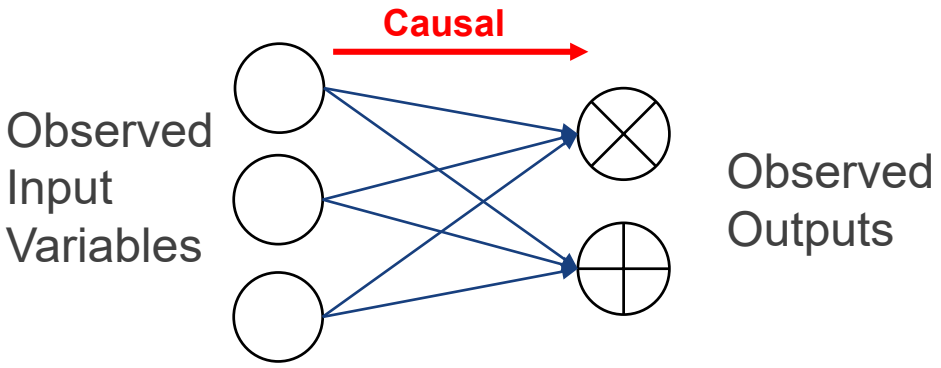
Latent Semantic Analysis (LSA)
The objective of LSA is reducing dimension for classification.

Figure 11.3 Plot of relative squared singular values by number of latent semantic dimensions
From *Practical Text Mining and Statistical Analysis for Non-structured Text data*

TOPIC MODELING

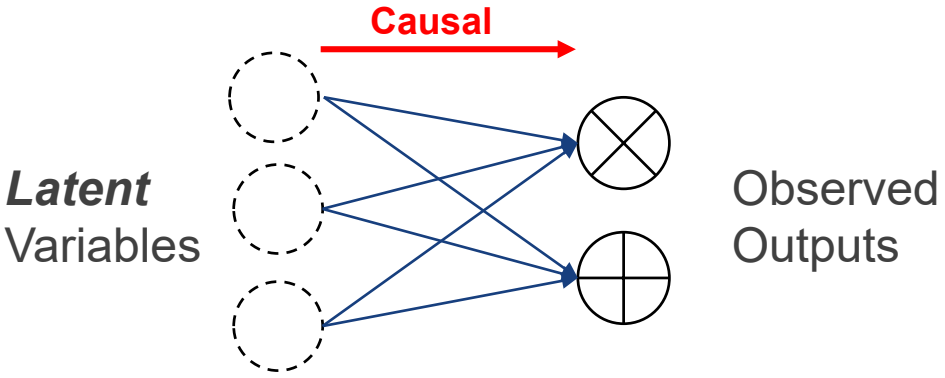
Supervised vs Unsupervised

- **Supervised categorization**



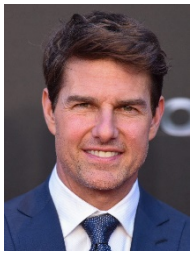
Essentially similar models
- **Inputs causally effects the outputs**
- **Input variables may not be observable**

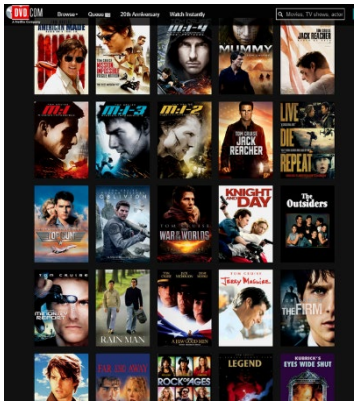
- **Unsupervised categorization**



Example

- Actors and their movies are observed inputs





- The potential tags of “Action”, “War” are latent variables

Text Example

- **What's unusual?**
  - Anomaly detection

Date	Amount	Location
2 Mar	$40	Penang
5 Mar	$20	KL
17 Mar	$30	KL
4 Apr	$80	Ipoh
9 Apr	$30	KL
14 Apr	$70	KL
20 May	$100	Johor
25 May	$20	KL
31 May	$3	Kiev
4 Jun	$40	KL
23 Jun	$50	KL
30 Jun	$30	KL
16 Jul	$70	Ipoh
16 Jul	$50	Ipoh

What is topic modeling?

A latent variable is a variable that cannot be observed. The presence of latent variables, however, can be detected by their effects on variables that are observable.

- **"Topic" modeling**
  - Can we figure out what discourses (==latent variables) would generate the collection of documents?
  - These discourses are just bunches of words
    - If done well, the bunches of words would seem naturally to be together, e.g.,
      - "wag", "bark", "bone", "bite", "dog"
      - "pilot", "plane", "wing", "flight"
  - These bunches of words constitute **topics**

Topic modeling is an unsupervised machine learning technique that's capable of scanning a set of documents, detecting word and phrase patterns within them, and automatically clustering word groups and similar expressions that best characterize a set of documents.

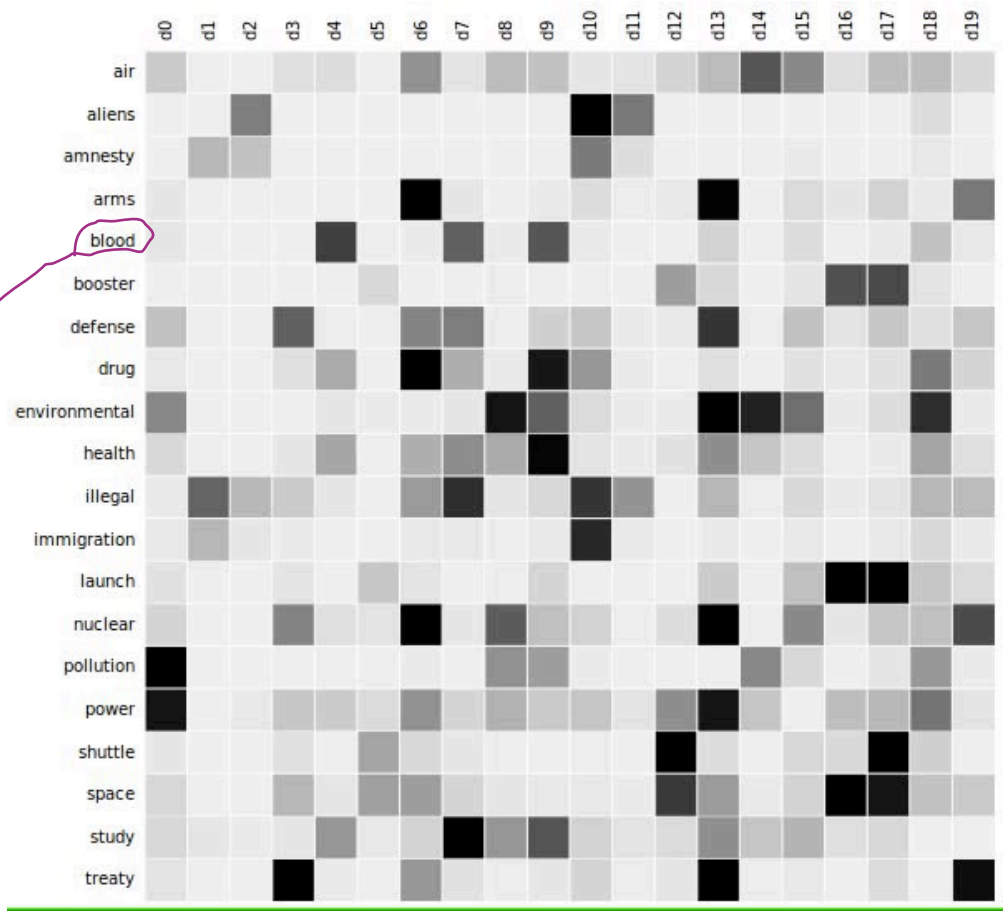
Animation of topic modeling

- Columns = documents
- Rows = words
- Squares = frequency
- Darker = higher frequency

- Group:
  - Documents using similar words
  - Words which occur in similar documents

Resulting set of words are "topics"

Number of Groups are pre-defined



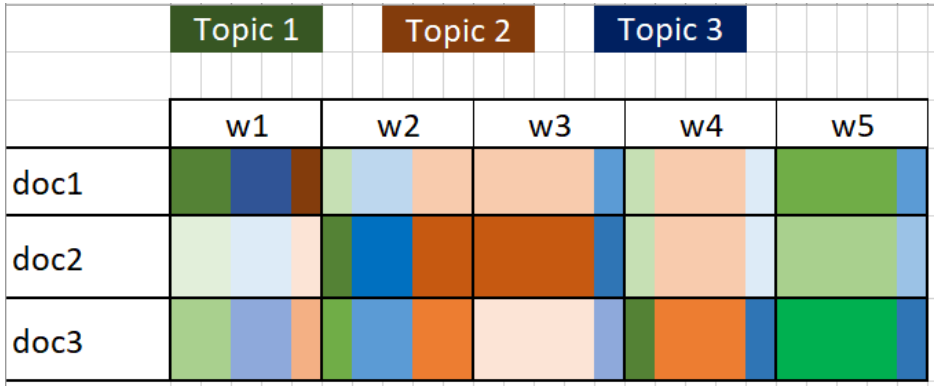
From: http://topicmodels.west.uni-koblenz.de/ckling/tmt/svd_ap.html

Animation of topic modeling

Input

output

	Topic 1	Topic 2	Topic 3		
	w1	w2	w3	w4	w5
doc1					
doc2					
doc3					



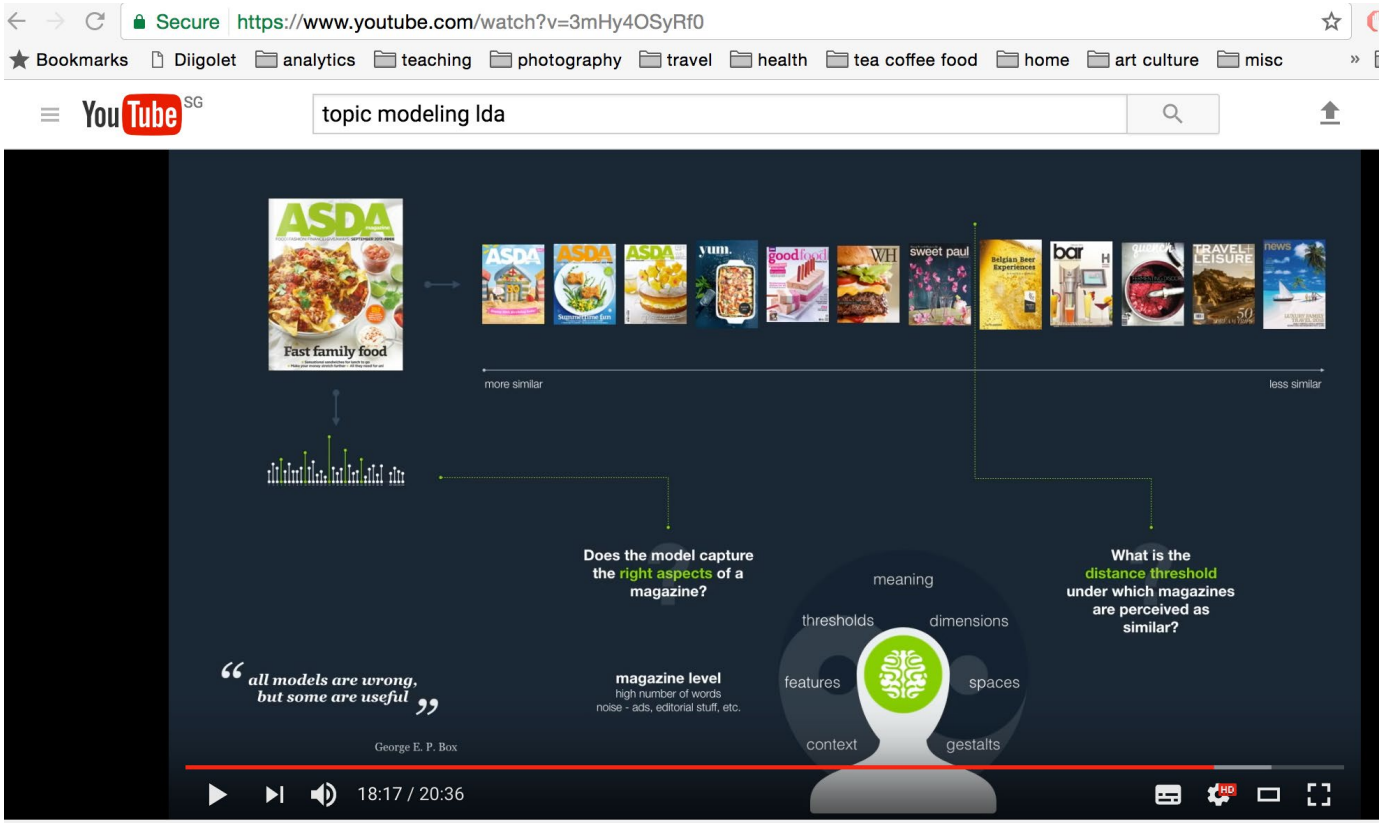
- Predefine number of topics
- TDM

- Word-Topic distribution
- Doc-Topic distribution
- Topics are indexed with numbers without tags
- Topics are represented by a list of (important) words

Is the 'colour size' and 'colour darkness' equally weighted? No
is there order of importance: area vs darkness? No

https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24

LDA Topic Model Explanation



From: https://www.youtube.com/watch?v=3mHy4OSyRf0

More examples of applications

- **Analysis of text, e.g.,**
  - Diachronic analysis: concerned with the way in which something, especially language, has developed and evolved through time.
    - Speeches during election campaign
      - Economy, abortion, build wall, reduce taxes,…
    - Speeches after taking office
      - Reduce taxes, create jobs, immigration, China,…
  - Contrast analysis:
    - Different candidates positions and issues
    - Characteristics of various media publications

https://towardsdatascience.com/lda-topic-modeling-an-explanation-e184c90aadcd

Reference & Resources

- Fabrizio Sebastiani, *A Tutorial on Automated Text Categorization*, web.iiit.ac.in/~jawahar/PRA-03/textCat.pdf
- F. Aiolli, *Text Categorization*, downloaded from http://www.math.unipd.it/~aiolli/corsi/SI-0607/Lez09.251006.pdf
- John Elder, Gary Miner, Bob Nisbet. *Practical Text Mining and Statistical Analysis for non-Structured Text Data Applications*, Academic Press, 2012
- Chris Manning & Hinrich Schutze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999
- Scott Weingart, *Topic Modeling for Humanists: A Guided Tour*, downloaded from http://www.scottbot.net/HIAL/index.html@p=19113.html
- Ted Underwood, *Topic Modeling made just simple enough*, downloaded from https://tedunderwood.com/2012/04/07/topic-modeling-made-just-simple-enough/
- NLP resources: http://nlp.stanford.edu/links/statnlp.html