Unsupervised learning: (Text) Clustering

Machine Learning for NLP

Magnus Rosell

Contents

Introduction

Clustering

To divide a set of objects into parts
To partition a set of objects so that . . .
The result of clustering: a clustering consisting of clusters

Clustering vs. Categorization

- **Categorization** (supervised machine learning)
  To group objects into predetermined categories.
  - Needs a representation of the objects, a similarity measure and a training set.
- **Clustering** (unsupervised machine learning)
  To divide a set of objects into clusters (parts of the set) so that objects in the same cluster are similar to each other, and/or objects in different clusters are dissimilar.
  - Needs a representation of the objects and a similarity measure.

Representation and Similarity

Representation:

	Features			
Objects	F1	F2	F3	...
obj1	.	.	.	...
obj2	.	.	.	...
⋮	⋮	⋮	⋮	⋱

Similarity: compare distribution of features between objects.
There are many different similarity measures.

What kinds of groups?

Depends on the algorithm, the similarity (and the representation), and the data set.

- Text: **Contents**, genre, readability...?

Why text clustering?

- Historically: preprocessing for searching
  the cluster hypothesis
- Postprocessing of search results, ex:
  - Clusty, http://clusty.com
  - iBoogie, http://www.iboogie.com
  - Carrot2, http://demo.carrot2.org/demo-stable/main
- Dimensionality reduction
- Multitext summarization
- Exploration tool:
  - Any unknown text set
  - Questionnaires
  - With different parameters and algorithms several different overviews of the same set of texts may be constructed.

Six Hypothetical Newspaper Headlines

1. Chelsea won the final.
2. Zimbabwe defeated China in the Olympic match.
3. Match-making in Olympic final.
4. Ericsson stock market winner, increased by 50 per cent.
5. Interest rate at 500 per cent in Zimbabwe.
6. Stock traders nervous as interest rate increases.

Term-document-matrix

	doc1	doc2	doc3	doc4	doc5	doc6
Chelsea	0.33	-	-	-	-	-
China	-	0.2	-	-	-	-
defeat	-	0.2	-	-	-	-
Ericsson	-	-	-	0.16	-	-
win	0.33	-	-	0.16	-	-
final	0.33	-	0.25	-	-	-
increase	-	-	-	0.16	-	0.16
interest	-	-	-	-	0.25	0.16
make	-	-	0.25	-	-	-
market	-	-	-	0.16	-	-
match	-	0.2	0.25	-	-	-
nervous	-	-	-	-	-	0.16
Olympic	-	0.2	0.25	-	-	-
per cent	-	-	-	0.16	0.25	-
rate	-	-	-	-	0.25	0.16
stock	-	-	-	0.16	-	0.16
trade	-	-	-	-	-	0.16
Zimbabwe	-	0.2	-	-	0.25	-

Text Categorization

Term-document-matrix

	Sports			Economy			Centroids	
	doc1	doc2	doc3	doc4	doc5	doc6	Sports	Economy
Chelsea	0.33	-	-	-	-	-	0.11	-
China	-	0.2	-	-	-	-	0.07	-
defeat	-	0.2	-	-	-	-	0.07	-
Ericsson	-	-	-	0.16	-	-	-	0.06
win	0.33	-	-	0.16	-	-	0.11	0.06
final	0.33	-	0.25	-	-	-	0.19	-
increase	-	-	-	0.16	-	0.16	-	0.11
interest	-	-	-	-	0.25	0.16	-	0.14
make	-	-	0.25	-	-	-	0.08	-
market	-	-	-	0.16	-	-	-	0.06
match	-	0.2	0.25	-	-	-	0.15	-
nervous	-	-	-	-	-	0.16	-	0.06
Olympic	-	0.2	0.25	-	-	-	0.15	-
per cent	-	-	-	0.16	0.25	-	-	0.14
rate	-	-	-	-	0.25	0.16	-	0.14
stock	-	-	-	0.16	-	0.16	-	0.11
trade	-	-	-	-	-	0.16	-	0.06
Zimbabwe	-	0.2	-	-	0.25	-	0.06	0.08

Centroid

Representation of a set of texts c. The **centroid**, the word-wise average of the document vectors:

$$\overline{c} = \frac{1}{|c|} \sum_{d_c \in c} d_c$$

Similarity between a text d and c: $\mathrm{sim}(d, c) = \mathrm{sim}(d, \overline{c})$.

If we use normalized text vectors, the dot product as a similarity measure, and do not normalize the centroids:

$$\mathrm{sim}(d, c) = \frac{1}{|c|} \cdot \sum_{d_c \in c} \mathrm{sim}(d, d_c),$$

the average similarity between d and all texts in c.

Text Categorization (cont.)

Text Categorization Algorithms

- Centroid Categorization
- Naïve Bayes
- ...

Text Categorization Applications

- Categorization into groups of content, genre, ...
  - Automatic webdirectories, etc.
- Text Filtering, Spam filtering
- ...

Text Clustering

Representation

- The vector space model of Information Retrieval is common.
- Clusters are often represented by their centroids.
- Similarity: cosine similarity

The goal: clusters of texts similar in content.

Two types of clustering algorithms

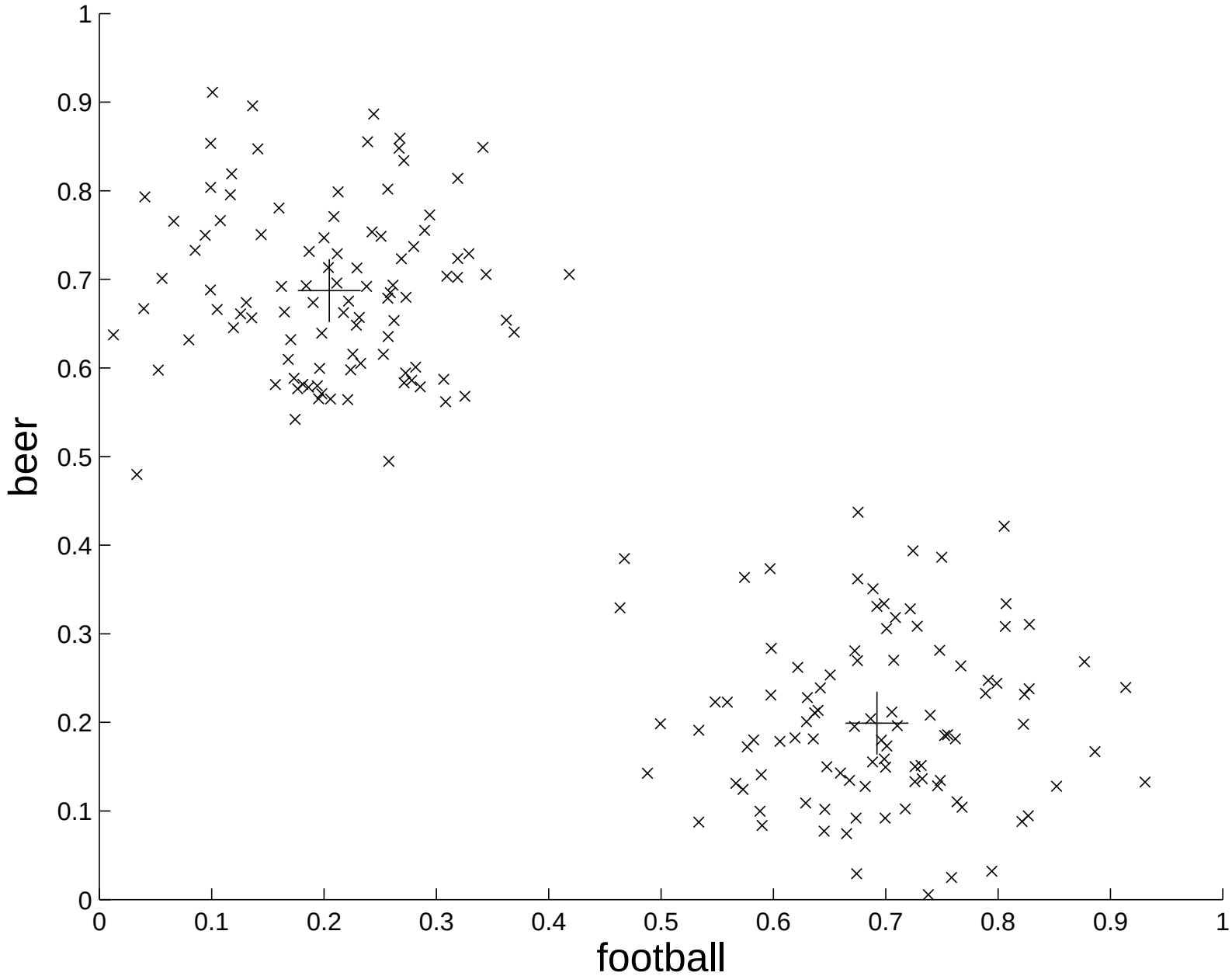
There are many clustering algorithms. Some of the most common can be divided into two categories:

- **Partitioning algorithms**, flat partitions
- **Hierarchical algorithms**, hierarchy of clusters

2D Clustering Examples

Will use some examples in 2D.
We only consider two words: *football*, *beer*.
In some way several texts are represented using only these two.
Use the inverse of cartesian distance as similarity: closer in the plane equals more similar.

2D Clustering Example 1: two well separated



A Partitioning Algorithm: K-Means

1. Initial partition, for example: pick k documents at random as first cluster centroids.
2. Put each document in the most similar cluster.
3. Calculate new cluster centroids.
4. Repeat 2 and 3 until some condition is fulfilled.

K-Means (cont.)

Initial partition

- Pick k center points at random
- Pick k objects at random
- Construct a random partition

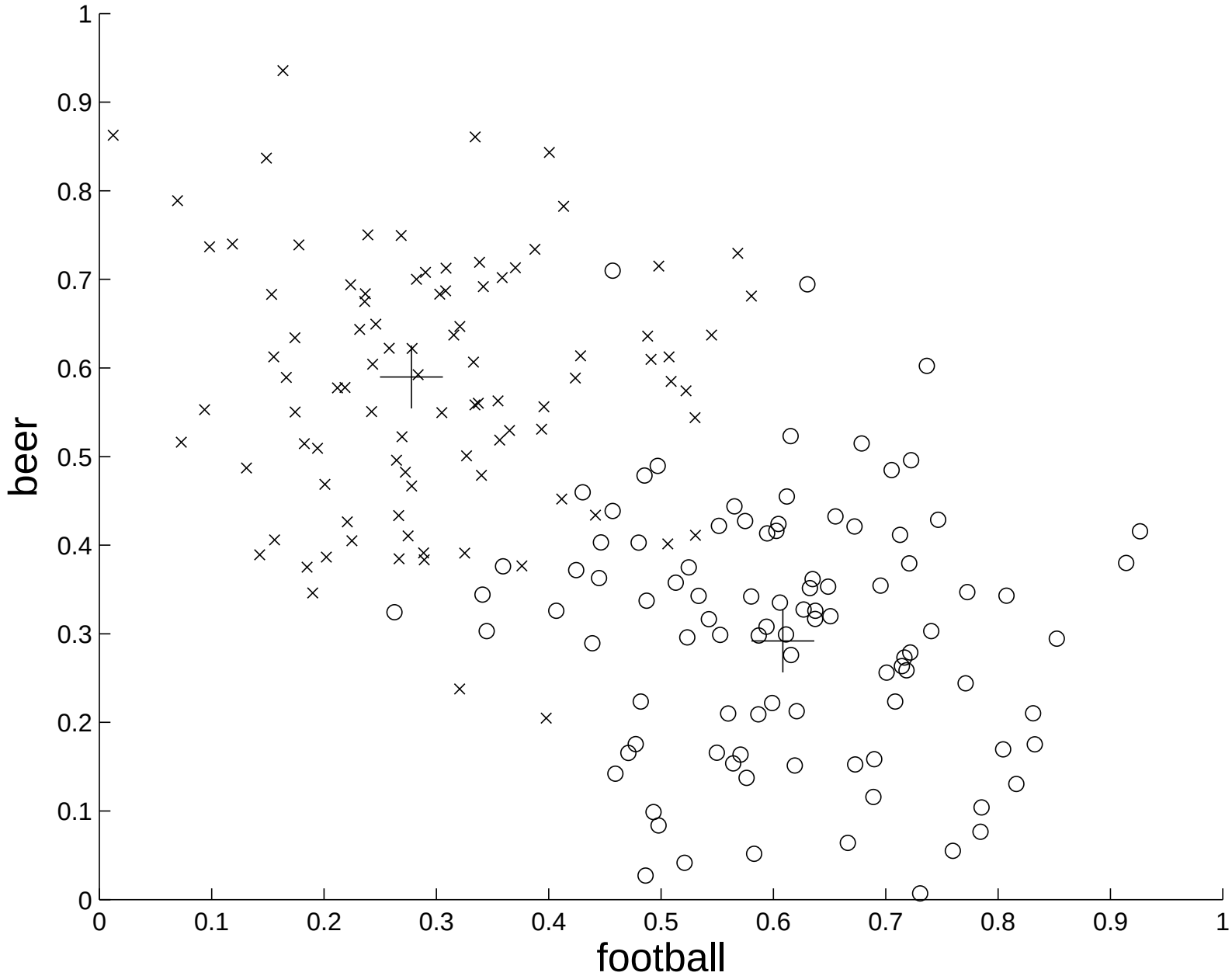
Conditions

- Repeat until (almost) no object changes cluster
- Repeat a predetermined number of times
- Repeat until a predetermined quality is reached

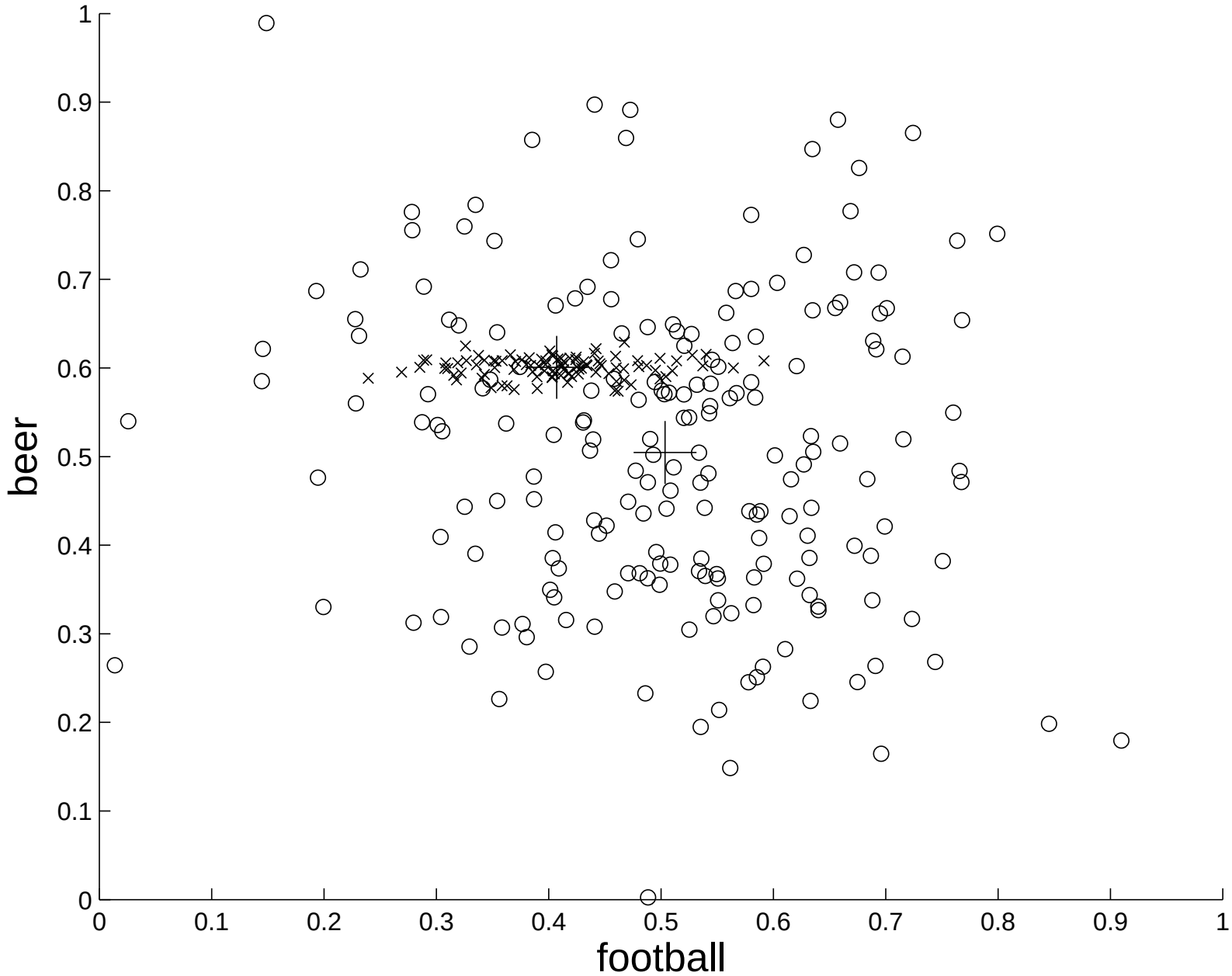
K-Means: example

Show K-Means example

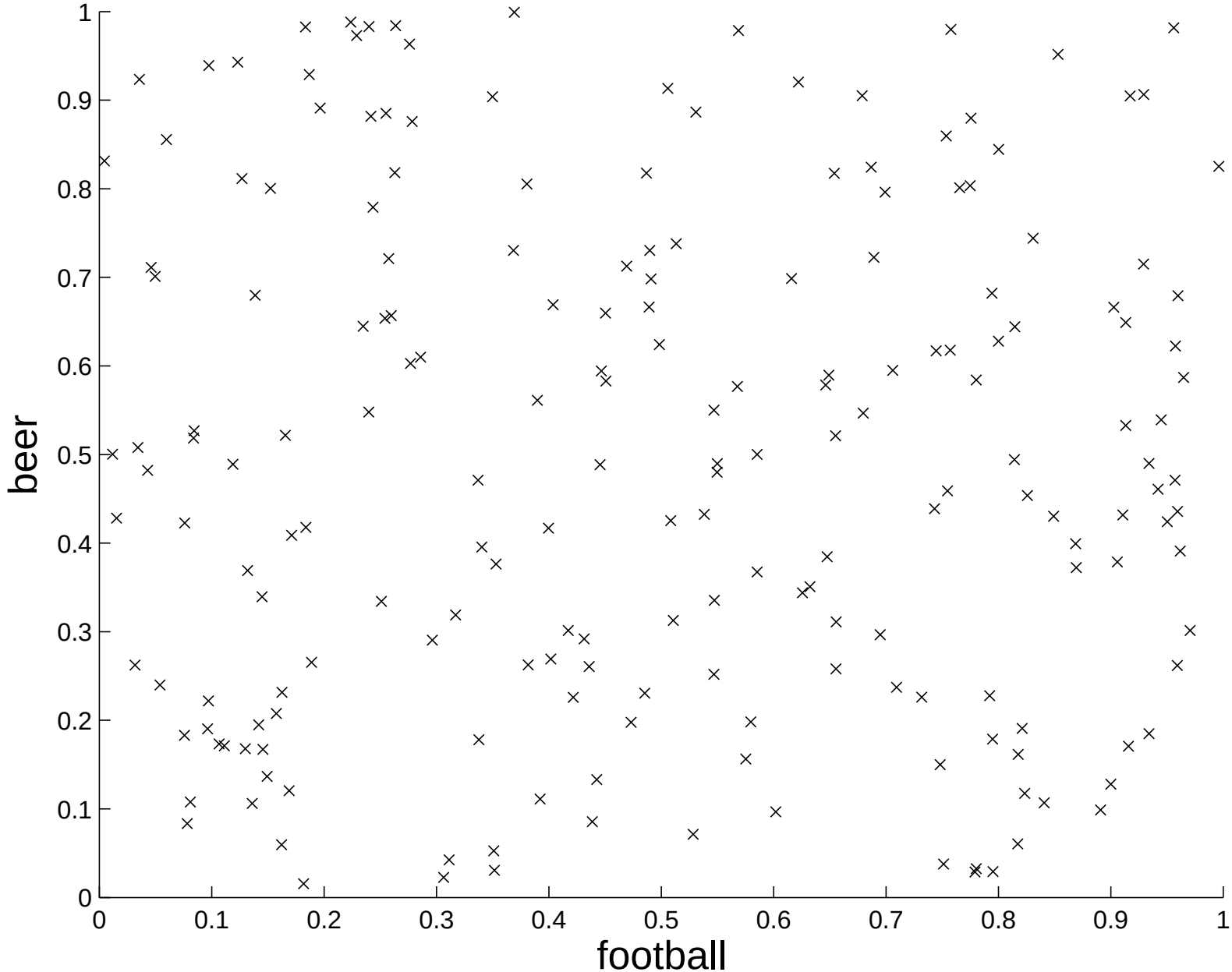
2D Clustering Example 2: two overlapping



2D Clustering Example 3: one tight, one wide



2D Clustering Example 4: random



K-Means (cont.)

K-Means

- Time complexity: $O(knI)$ similarity calculations where k number of clusters, n number of objects, and I number of iterations.
- Decide number of clusters in advance
  - Try several clusterings!
- Different results depending on initial partition
  - Try several initial partitions!
  - What is a good partition?
- "Global" – revisits all texts in every iteration: the centroids are kept up-to-date. Takes word co-occurrence into account through centroids.

Hierarchical Algorithms: Agglomerative

Agglomerative

1. Make one cluster for each document.
2. Join the most similar pair into one cluster.
3. Repeat 2 until some condition is fulfilled.

The result is a cluster hierarchy. Often depicted as a dendogram.

Hierarchical Algorithms: Agglomerative (cont.)

Conditions

- Repeat until one cluster.
- Repeat until a predetermined number of clusters
- Repeat until a predetermined quality is reached (the quality is usually increasing with number of clusters)

Hierarchical Algorithms: Agglomerative (cont.)

Similarity in Agglomerative:

1. Single Link: similarity between the most similar texts in each cluster. "elongated" clusters
2. Complete Link: similarity between the least similar texts in each cluster. "compact" clusters.
3. Group Average Link: similarity between cluster centroids. "middle ground".
4. Ward's Method: "combination" method.

Hierarchical Algorithms: Agglomerative (cont.)

The similarity matrix:

	doc1	doc2	doc3	...
doc1	s(1,1)	s(1,2)	s(1,3)	...
doc2	s(2,1)	s(2,2)	s(2,3)	...
doc3	s(3,1)	s(3,2)	s(3,3)	...
⋮	⋮	⋮	⋮	⋱

Calculation time: $O(n^2)$

Hierarchical Algorithms: Agglomerative (cont.)

Agglomerative

- Time complexity: $O(n^2)$ similarity calculations
- Results in a hierarchy that could be browsed.
- Deterministic: same result everytime.
- "Local" – in each iteration only the similarities at that level is considered. Bad early descisions can't be undone.

Hierarchical Algorithms: Divisive

Divisive algorithm

1. Put all documents in one cluster.
2. Split the worst cluster.
3. Repeat 2 until some condition is fulfilled.

Example: **Bisecting K-Means**

- Splits the biggest cluster into two using K-Means.
- Different results depending on all initial partitions.
- Considers *many* objects in every iteration.
- Robust – dominating features may be dealt with in the beginning.
- $O(n \log(k) I)$

Comparison of Algorithms

K-Means

- flat partition
- decide number of clusters in advance
- different results depending on initial partition
- "global", considers all objects in every iteration
- fast: $O(knI)$

Agglomerative

- hierarchy
- may stop at "optimal" number of clusters
- same result every time (deterministic)
- "local", bad early descision can not be changed
- slow: $O(n^2)$

Algorithm Discussion

- Will always find clusters: 2D Example 4: random.
- Outliers.
- Hard and soft (fuzzy) clustering
- Texts on the border between clusters
- Many other clustering algorithms
  There is no clear evidence that any clustering algorithm performs better in general. Some algorithms perform better for particular data and applications.

Clustering Result

Number of texts

	Word	Economy	Culture	Sports	Sweden	World	Total
Cluster 1	per cent, index, stock, increase, rate	167	4	1	37	23	232
Cluster 2	film, aftonbl, write, tv, swede	18	421	22	176	40	677
Cluster 3	game, match, swedish, win, club	0	19	452	10	14	495
Cluster 4	reut, press, company, stockholm, stock	312	8	6	36	10	372
Cluster 5	police, death, wound, person, tt	3	48	19	241	413	724
Total	tt, swedish, per cent, write, stockholm, reut, game, time, day, stock	500	500	500	500	500	2500

Evaluation

It is very hard to evaluate a clustering. What is a good clustering? Relevance?

- External Evaluation
  - Indirect evaluation: how clustering influences an application that uses it as an intermediate step.
  - **External quality measures**: compare the result to a known partition or hierarchy (categorization)
- Internal Evaluation
  - **Internal quality measures**: use a criterion inherit for the model and/or clustering algorithm. For instance the average similarity of objects in the same cluster.

Evaluation: Definitions

Let

$C = \{c_i\}$	a clustering with γ clusters c_i
$K = \{k^{(j)}\}$	a categorization with κ kategories $k^{(j)}$
n	number of documents
n_i	number of documents in cluster c_i
$n^{(j)}$	number of documents in category $k^{(j)}$
$n_i^{(j)}$	number of documents in cluster c_i and category $k^{(j)}$
$M = \{n_i^{(j)}\}$	The confusion matrix

Evaluation: Internal Quality Measures

$sim(c_i, c_i)$ is the cluster *intra similarity*. It is a measure of how "cohesive" the cluster is.

If the centroids are not normalized it is the average similarity of the texts in the cluster.

The *intra similarity* of a clustering:

$$\Phi_{intra}(C) = \frac{1}{n} \sum_{c_i \in C} n_i \cdot sim(c_i, c_i), \quad (1)$$

which is the average similarity of the texts in the set to all texts in their respective clusters.

Evaluation: Internal Quality Measures (cont.)

Similarily, the average similarity of all texts in each cluster to all the texts in the entire set may be calcuated:

$$\Phi_{inter}(C) = \frac{1}{n} \sum_{c_i \in C} n_i \cdot sim(c_i, C). \quad (2)$$

This measures how separated the clusters are.

Evaluation: Internal Quality Measures (cont.)

Internal quality measures are depending on the representation and/or similarity measure.

- Don't use to evaluate different representations and/or similarity measures.
- Don't use to evaluate different clustering algorithms, since they may utilize this differently.

Evaluation: External Quality Measures

Precision and Recall

For each cluster and category:

- **Precision**: $p(i,j) = n_i^{(j)}/n_i$
- **Recall**: $r(i,j) = n_i^{(j)}/n_j$

Precision, $p(i,j)$, is the probability that a text drawn at random from cluster c_i belongs to category $k^{(j)}$. In other words: $p(k^{(j)}|c_i)$, the probability that a text picked at random belongs to category $k^{(j)}$, given that it belongs to cluster c_i.

Evaluation: External Quality Measures (cont.)

Purity

- **Cluster Purity**:

$$\rho(c_i) = \max_j p(i,j) = \max_j \frac{n_i^{(j)}}{n_i}$$

- **Clustering Purity**:

$$\rho(C) = \sum_i \frac{n_i}{n} \cdot \rho(c_i)$$

Purity disregards all other than the majority class in each cluster. But a cluster that only contains texts from two categories is not that bad . . .

Evaluation: External Quality Measures (cont.)

Entropy

- **Entropy** for a cluster: $E(c_i) = -\sum_j p(i,j) \log(p(i,j))$
- Entropy for the Clustering: $E(C) = \sum_i \frac{n_i}{n} E(c_i)$

Entropy measures the disorganization. A lower value is better.

- $\min(E(c_i)) = 0$, when texts only from one category in the cluster.
- $\max(E(c_i)) = \log(\kappa)$, when equal number from all categories.

Normalized entropy: $NE(c_i) = E(c_i)/\log(\kappa)$.

Evaluation: External Quality Measures (cont.)

Entropy does not consider the number of clusters.
A clustering of n clusters would be considered perfect.

Sometimes you might accept to increase the number of clusters to get better results, but not always.

We can consider the whole confusion matrix at once . . .
. . . **Mutual Information**. And normalized mutual information.

Evaluation: External Quality Measures (cont.)

External quality measures are depending on the quality of the categorization.

- How do we know a categorization is of high quality?
- What is a good partition?
- There might be several good partitions.
- A certain corpus might be especially hard to cluster.
  None of the external measures takes into account how "difficult" the texts of the actual corpus are.

If we do not have the possibility to test in the context of another task or by asking humans, we have to stick to external quality measures.

Evaluation (cont.)

Try to use at least one internal and one external measure. Be careful when discussing the implications!

Use baseline methods, for instance random clustering.

(Word Clustering)

One possibility: cluster the columns of the term-document-matrix.

Clusters of related words: words that have a similar distribution over the texts.

Text Clustering Exploration

- Text Mining
- Scatter/Gather
  1. Cluster a set
  2. Choose clusters (words with high weight)
  3. Cluster again
  4. Repeat until satisfied.

Clustering Result Presentation

A human has to interpret the (end) result!

- Search Engine – index
- Clustering – dynamic table of contents

How should the result be presented?

- Textual
- Graphical – Visualization

Textual Presentation

The search engines, the browser pages, the confusion matrix.
Display: lists of clusters, texts, and words.

Cluster Labels/Descriptions: usually single words or phrases based on word distribution:

- in cluster (*descriptive*)
- between clusters (*descriminating*)

Other:

- Suffix Tree Clustering
- Frequent Term-based Clustering

Visualization

- Similarity
  - SOM - Self Organizing Maps: WEBSOM.
  - Projections
- Representation
  - Infomat