(N)GSLT Clustering Course

Day 1

Magnus Rosell and Viggo Kann

2008-05-28

Contents

The informal HLT group at CSC

The informal HLT group at CSC

- Viggo Kann, professor CS
- Magnus Rosell, lic CS
- Ola Knutsson, Ph.D. HCI
- Jonas Sjöbergh, Ph.D. CS
- Martin Hassel, Ph.D. CS

Research focus

Research focus

- developing efficient, resource lean and evaluable NLP methods, especially for Swedish
- developing useful and freely available NLP resources and tools using these methods

Recent projects

Recent projects

- Spelling and grammar checking
- Construction of dictionaries
- Giving the computer a sense of humour
- Information retrieval using text clustering

Introduction

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 – Endless Possibilities

Number of possible partitions of a set with n objects:

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$$

$B_0 = 1, B_1 = 1, B_2 = 2, B_3 = 5, B_4 = 14, B_5 = 52, B_6 = 203, \ldots$

Test all to see which one is best?
Restrictions $\Rightarrow$ suboptimal result.
What is a good partition?

- Depends on the purpose.
- There might be several good partitions.

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.

What kinds of groups?

Depends on the represenation and the similarity measure.

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

Terms

Partition – Grouping
Categorization – Classification
Clustering

Terms (cont.)

- Information/Knowledge Extraction/Retrieval
- Data Mining, Text Mining – to automatically find (new) information in (large) data sets, text sets
- Text Categorization
- Automatic Text Categorization
- Text Clustering

All terms can used with "document" instead of "text".

Terms (cont.)

- Language Technology
  - Information Retrieval
    - Search Engines
    - Statistical lexical semantics (LSA, Random Indexing ...)
    - Document Clustering
    - Document Categorization
- Supervised Machine Learing
  - Categorization
    - Document Categorization
- Unsupervised Machine Learing
  - Clustering
    - Document Clustering

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.

Show Clustering Search Engine

Show Clustering Search Engine

Course administration

- Course Plan and Content
- Requierments
  - Literature
  - Laboration reports
  - Individual Project
- Literature
- Two Laborations
  - Reports
- Individual Project
  - Definitions due
  - Presentation on third day
  - Report due
- Date for third day

Information Retrieval

Foundational assumptions:

- The texts can be represented by the words that appear in them (bag of words).
  (The content or meaning of texts . . . ?)
- Texts that contain the query terms are relevant to the query.

To find relevant information.

- Are search engines good?
- Are the returned texts relevant?
- What do we mean by relevance?
- Too subjective – quantize!
  - Test set, with queries and relevant documents.

- Precision
$$p = \frac{D(ret, rel)}{D(ret)}$$
- Recall
$$r = \frac{D(ret, rel)}{D(rel)}$$

where

$$\begin{aligned} D(rel) &= \text{number of relevant documents (in total)}, \\ D(ret) &= \text{number of returned documents}, \\ D(ret, rel) &= \text{number of returned and relevant documents}. \end{aligned}$$

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.

	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	-

IR – Query

Query

The query is processed like the texts.

- For instance "Olympic final"
  - gives a vector
    with 0.5 for *Olympic*, 0.5 for *final*, and zero otherwise.
  - dot product with the document vectors
    gives a result bigger than zero for documents 1, 2, and 3.
- Rank the retrieved documents by their dot product value.

IR – Weighting

The values of the matrix, the weight: tf·idf

- Local Weight – tf: term frequency
  - How common the word is in a document.
  - $\mathrm{tf}_{i,j} = n_{i,j}/n_j$, where $n_i^{(j)}$ is the number of times the word i appears in document j, and n_j is the number of words in document j.
- Global Weight – idf: inverse doucment frequency
  - How specific the word is (in the set of texts).
  - For instance $\mathrm{idf}_i = \log(N/n)$, where N is the total number of documents, and n is the number of documents word i appears in.

Many other weighting schemes. Local and global factors.

IR – Weighting (cont.)

- Stoplist
  - Remove non-content words.
  - Linguistically motivated – grammatical words.
  - Statistically – very common words.
- Term normalizing
  - Truncation – predetermined maximal length.
  - Lemmatizing – morphological analysis.
  - Stemming – middle ground: strip affixes.
- Document normalization
  - Scale (normalize) the document vectors to unit length.
  - The direction of the vector represents the text.
  - Two texts with the same relative frequency of words, but different in length, will be considered identical.

IR – Similarity Measures

Similarity measures, or distance measure

- Cartesian distance, number of common terms, etc.
- *Cosine Measure*: the cosine of the angel between the text vectors – the dot procuct between the text vectors normalized to unit length.

The term-document-matrix: $M = \{m_{w,t}\}$. The cosine measure:

$$\begin{aligned} sim(t_u, t_v) &= \frac{1}{\|t_u\| \cdot \|t_v\|} t_u \circ t_v = \{\text{normalized}\} = \\ &= t_u \circ t_v = \sum_{w \in W} m_{w,t_u} \cdot m_{w,t_v} \end{aligned}$$

IR – Vector Space Model

A Vector Space Model

- Each text is represented by a vector in a many-dimensional vector space. The term-document-matrix.
- The vectors are compared using a similarity measure: the cosine measure.

The objective: to measure similarity between texts!

IR – Objections to the Vector Space Model

Objections

- Different types of texts.
- Different languages.
- A text may have several topics.
- Words may mean different things, depending on the context.
- There is content in word order.
- etc.

IR – Extensions

- Language specific:
  - Different languages poses different problems when constructing the vector space. For instance: what is a word/term?
  - Stemming, lemmatization (increase precision and recall)
  - Solid compounds (Finnish, German, Swedish, etc.)
  - Part of speech
- Other possiblities when indexing:
  - Phrases
  - N-gram (letters?)
  - Structure: bold face, sections, meta data (tags, like bold face, paragraphs) etc.
  - Most notably: links, Googles PageRank.

IR – Extensions (cont.)

- Relevance feedback.
- Term expansion. To automatically augment texts/queries with related words.
  - Lexicon of synonyms (WordNet)
  - Statistical lexical semantics (LSA, Random Indexing)
  - Spelling errors

Presentation

- Encourage "good" queries.
- Snippets
- Text summarization
- Grouping of similar documents:
  - Categorization
  - Clustering

Text Categorization

Term-document-matrix

	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.

Centroid

Centroids: Meaning and Co-occurrence

Consider the polysemous word *salsa*. It may have a high weight in the centroids of

- a "music" group
- and a "food" group,

but it will be *defined* by the other words with high weight.

Also, pairs of synonyms will appear in the same centroid as they have a lot of words both co-occurr with.

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.

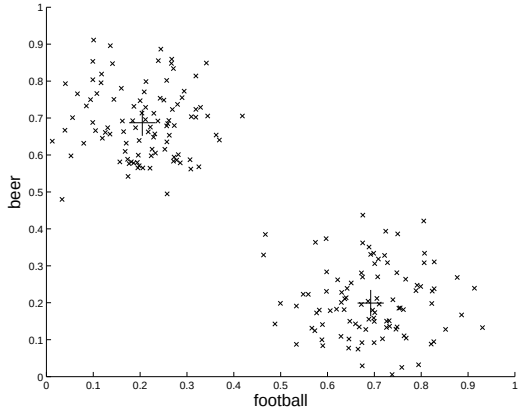
Two types of clustering algorithms

- **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



K-Means

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

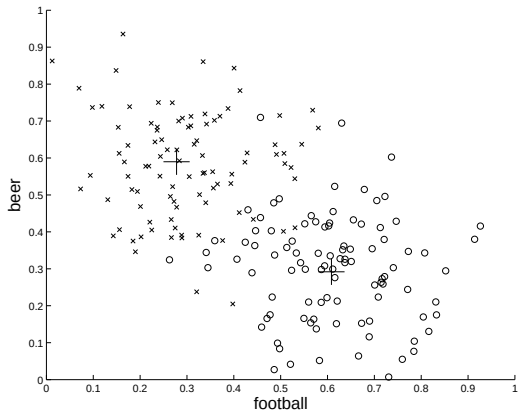
Conditions

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

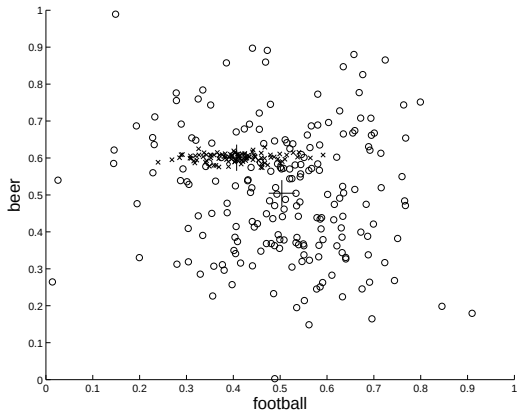
K-Means: example

Show K-Means example

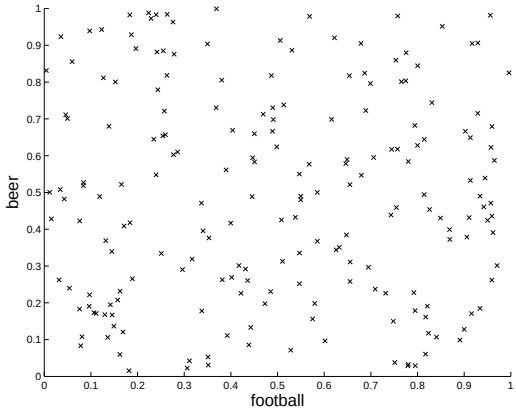
2D Clustering Example 2: two overlapping



2D Clustering Example 3: one tight, one wide



2D Clustering Example 4: random



K-Means (cont.)

K-Means

- Time complexity: $O(knI)$ similarity calculations
- 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. Single and Complete Link do not use the word co-occurrence in the centroids. The centroid co-occurence information is partly lost when calculating similarities between centroids, as in Group Average Link.

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.
- $O(n \log(k) I)$

Hierarchical Algorithms: Bisecting K-Means Example

Show Bisecting K-Means example

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.
2D Example 2: two overlapping.
2D Example 3: one tight, one wide.
"local" vs. "global"

Algorithm Discussion (cont.)

Hard and soft (fuzzy) clustering
Texts on the border between clusters
Many other clustering algorithms

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	371
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
  - Task oriented.
  - **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.)

Similarly, 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 ...

Evaluation: External Quality Measures (cont.)

Mutual Information

Let $p_i^{(j)} = n_i^{(j)}/n$, $p_i = n_i/n$ and $p^{(j)} = n^{(j)}/n$.

- Mutual Information for a clustering:

$$MI(C,K) = \sum_i \sum_j p_i^{(j)} \log\left(\frac{p_i^{(j)}}{p_i p^{(j)}}\right)$$

Evaluation: External Quality Measures (cont.)

A theoretical tight upper bound for the mutual information is $MI_{max}(C,K) = \log(\kappa\gamma)/2$, the mean of the theoretical maximal entropy of the clustering and the categorization compared to each other.

Normalized Mutual Information:
$NMI(C,K) = MI(C,K)/MI_{max}(C,K)$

NMI can theoretically be compared over clusterings with different numbers of clusters and categorizations with different numbers of categories.

None of the external measures takes into account how "difficult" the texts of the actual corpus are.

Evaluation: External Quality Measures (cont.)

Pair Measures

We may count the number pairs of texts instead of single texts:

	Same category	Different categories
Same cluster	tp	fp
Different clusters	fn	tn

- Precision and Recall:

$$P(C,K) = \frac{tp}{tp+fp},$$
$$R(C,K) = \frac{tp}{tp+fn}.$$

The number of pairs in the same group is usually much smaller than the number of pairs in different groups.

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.

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)

(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.

Clustering Exploration

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.

Hypothesis Generation

Revealing Relations between Open and Closed Answers in Questionnaires through Text Clustering Evaluation

Magnus Rosell and Sumithra Velupillai

Text set:

- The Swedish Twin Registry
- A questionnaire
  - An open answer: occupation (41 549)
    - Vector space model
  - A closed answer: smokers (29%)

Hypothesis Generation (cont.)

1. Cluster the text set
2. Identify interesting clusters (high or low percentage of smokers)
3. Explore cluster contents (Infomat)
4. Formulate potential hypotheses

- Iterate
- Interactive exploration
- Pursue hypotheses further

Hypothesis Generation (cont.)

Results:

- Farmers smoke less than the average
  - A few hours of exploration
  - No prior knowledge on smoking habits in occupation groups
- Comparable surveys
- Hypotheses can be generated

Conclusions:

- No need to avoid free-text answers

Future work:

- Questionnaires from other domains
- Similar text sets, e.g. electronic medical records

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

We have seen several examples: the search engines, the browser pages, the confusion matrix.

- Cluster lists
- Text lists
- Word lists

Textual Presentation (cont.)

Cluster Labels/Descriptions

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

Infomat and Laboration 1

Infomat

A Vector Space Visualization and Processing Tool

Infomat GUI:

- View of the weight matrix
- Similarity visual as a distributional patterns
- Some of the tools:
  - Visual: zoom in and out, remove rows, columns, and matrix elements
  - Preprocessing: stoplist, filtering, weighting
  - Grouping of rows and columns: K-Means, Relative Clusterer, Based on file location
  - Textual: matrix element info, result exportation, grouping and group lists
  - Evaluation: visual and traditional

Infomat and Laboration 1

Infomat Demo and Laboration 1