Boosting LVM-based Document Clustering and Visualization with Genetic Chromodynamics

Eleonóra Víg and Tekla Sinkó
{vig_nora, s_tekla}@yahoo.com

Babeş-Bolyai University
Cluj-Napoca, 2006

Abstract

We present probabilistical models for text document analysis and visualization by means of clustering and topographic organization. These models are capable of revealing the underlying semantic structure in text-based documents, and facilitate human interpretation of the high dimensional data by two-dimensional visualization.

However, results of applying *latent variable models* are under the influence of initialization values, and setting the number of latent variables beforehand is necessary. To solve these problems, we have resorted to *genetical chromodynamics*, a new and potentially appealing strategy for multimodal optimization and search problems.

To illustrate these models and examine their efficiency, we developed an application that solves two text mining problems: unsupervised classification and 2D visualization of text documents.

Keywords: document clustering and visualization, latent variable models, EM algorithm, genetical chromodynamics, automatic discovery of topics and their number

1 Preprocessing of documents

The analysis of text documents is only efficient if they are properly preprocessed first, as the quality of data has enormous impact on text mining results. Noise and erronous entries make it at best difficult, but they can also lead to false interpretation.

Preprocessing starts with tokenizing the documents, thus decomposing them to lexical items (called *tokens, terms*), converting all letters to lowercase. At the same time, the various form of words are replaced with their dictionary form (e.g. *is, are, were* to *be*). In English, as in some other languages, parts of speach, tense and number are conveyed by word inflections. Yet these forms don't denote new words, they rather hinder analysis, hence we only keep stems: words which derive from a common root are reduced to one symbol (e.g. "computation", "computing" and "computer" to the stem "comput"). This process is called *stemming*. Common stemming methods use a combination of morphological analysis (e.g. Porter's algorithm) and dictionary lookup (e.g. WordNet [3]). However, it must be used mindfully, as it can lead to disregarding important information. For example, Porter's algorithm stems both *university* and *universal* to *universe* [3]. Thus, although stemming reduces the dimensionality of the observation space, it has been found to reduce performance in some experimental evaluations [8]. When in doubt, it is better not to stem.

Most natural languages have so-called function words and connectives such as articles and prepositions that appear in a large number in documents and are typically of little use in pinpointing documents that satisfy our need (they are used for providing continuity and grammatical structure in language rather than content). Such words are called *stopwords*. As they don't play an important part in text mining, we leave them out. Yet, polysemy might give us a surprise: a word having multiple senses depends on context or part of speech. For example, *can* as a verb is not very useful keyword, but *can* as a noun could play central role in data analysis, so it should not be included in the stopword list.

These processes presume upon complex and time consuming morphological analysis, and efficient algorithms do exist for this purpose. As document preprocessing is not the primary goal of our paper, we use Andrew McCallum's *Bow, A Toolkit for Statistical Language Modeling, Text Retrieval* to achieve data of high quality [9].

2 Representation of documents

Our goal is to represent data in such a form that facilitates actual processing (an important criterion is the size of the dataset), this is why we choose the **vector space model**. Thus documents are represented as vectors in a multidimensional Euclidean space. Each axis in this space corresponds to a term (token).

For the representation of the documents, we should consider all different words in all documents. The usage of synonyms and different writing styles also increases the dimensionality of word space. This is obviously an inconvenience as implies manipulation of an extremly high dimensional data matrix, what due to current technical possibilities would be extremely inefficient. Therefore, we use a mindfully selected set of words. The size of this dictionary is previously fixed. As the selection of the words is a lengthy and complicated process, we use the program of McCallum, *Bow* [9] to extract the T most often occurring terms.

Let us consider a corpus consisting of N documents and a dictionary of T words. One document can be represented by a T dimensional vector, whose indices indicate the words of our dictionary. For the representation of the whole set (N independent observations), we use a $T \times N$ matrix denoted by D:

$$D = (d_{tn})_{t=1..T,n=1..N}$$

The i^{th} index of the j^{th} document's vector signifies the numerically expressed binding of the i^{th} term to the respected document. This value is usually 0 if the word doesn't appear in the document, and a positive number otherwise, which can be defined to suite the purpose of the analysis. The measurements of our choice were the following:

1. in case of **multivariate Bernoulli** or **binomial data model**, d_{ij} is set to 1;
2. when using **multinomial data model**, d_{ij} expresses term frequency.

Other term 'weightings' are also possible, such as (let us denote by w the number of occurrences of word i in document j):

1. $d_{ij} = \log(w + 1)$
2. $d_{ij} = \sqrt{w}$
3. $d_{ij} = w \log \frac{N}{n_w}$, where n_w denotes the number of occurrences of word i in the entire corpus (this formula is called the *Term Frequency – Inverse Document Frequency* weighting (TFIDF))

All these models are based on the intuition that although the importance of a word seems to be useful information, in many situations this isn't reflected in the frequence of its usage [8].

Nevertheless, the multinomial and Bernoulli data models have wide utility, as they have been found to provide a good enough description for text classifiers or clustering algorithms and also enable a suggestive interpretation by 2D visualization.

The multinomial representation has been found superior to the Bernoulli in both clustering and visualization, as it provides more precise information about the words occurring in the documents. Normalizing the document length has the benefit of preventing longer text documents from capturing more importance than the shorter ones [8]. In other words, an occurrence of a word in a short document should obviously be treated as more important than an occurrence of a word in a long text.

From a linguist's perspective though, such models are insufferably crude: there is not a shade of grammar in these characterisations or short-range dependence that is commonly seen between terms [3] (e.g. the word *spite* is quite likely to follow the word *in* and precede the word *of*), as it assumes that the words are independent events. But note that these models are approximations to physical reality, and this representation of data offers a straightforward way for efficient statistical tools to be applied, producing acceptable experimental results for machine learning tasks in the text domain. This is feasible as the vector space model still provides a considerable amount of information about associations between words and documents which seem to be sufficient for topical clustering and visualization: results are greatly influenced by word context in the sense that a word by itself doesn't determine the document's appertainance to one cluster or another, the other words from the document also have an influence on this. Surprisingly, a number of studies have shown that little is to be gained by employing sophisticated computational linguistics models over this simple representation and developing richer models that are also computationally feasible remains a challenging problem [8].

3 Generative models for document clustering and visualization

Given a corpus with various topics, documents are likely to include terms highly indicative of one or relatively few topics, together with noise terms. Statistical pattern recognition and information retrieval algorithms are built on the premise that the patterns (documents) that we observe are *generated by random processes* that follow specific distributions. The observations let us estimate various parameters pertaining to those distributions which in turn let us design strategies for analysing the patterns by way of clustering, indexing or 2D visualization. Specifically we make the assumption that a distribution of the words in the dictionary is assigned to each cluster, respectively grid point. Proposing credible distributions that can generate natural language is very difficult, and the computation involved is usually heavy-duty [3]. Therefore, we must be satisfied to model only a few aspects of the observed data. For example, we won't consider dependencies and ordering between terms, and we assume that term occurrencies are independent events.

We propose generative models of documents that don't define similarity or distance measures: we assume K ($K << N$) random process generating the documents, our goal being the discovery of these processes, called *latent variables* and the associated parameters that are most likely to have generated a given collection of documents.

3.1 General formulation of the generative model

The general form of the latent variable model is:

$$d_n = g(Ac_k) + n,$$

where d_n denotes the T-dimensional observable data, generated by $g(.)$, a non-linear function of an L-dimensional latent variable $(c_k)_{k=1..K}$ and A, a matrix containing the model parameters. Presume that the observed data are the *noisy* extensions of a latent variable, as n shows [2, 10].

In the following, we will describe the actual process of generating documents. Component c_k is chosen at random, with probability $P(c_k)$, the so-called *prior probability*. This latent variable generates document d_n with probability $p(d_n|c_k)$. Hence the probability density function of document d_n is expressed by the linear combination:

$$p(d_n) = \sum_{k=1}^{K} P(c_k)p(d_n|c_k).$$

The posterior probabilities signify the responsabilities of components to generate a particular document. They can be written using Bayes' theorem in the form:

$$P(c_k|d_n) = \frac{p(d_n|c_k)P(c_k)}{p(d_n)}.$$

These probabilities satisfy the constraints:

$$\sum_{k=1}^{K} P(c_k|d_n) = 1$$

$$0 \leq P(c_k|d_n) \leq 1.$$

3.2 The Latent Class and Trait Model

The model described above is suitable for text mining, being able to discover meaningful semantical structure in the corpus. The *latent class model* provides probabilistical labels, useful for the interpretation of results. These are highlighted by two-dimensional visualization, achieved by applying the *latent trait model*.

In the case of text document clustering, the latent variable is played by *topic categories*. The latent dimension is the number of clusters (K), previously given.

On the other hand, the starting point for the latent trait variables will be a two-dimensional $2 \times K$ uniform grid of points (X), mapped by a set of L non-linear and linear basis functions to the L-dimensional latent space. Thus, the non-linear image of the k^{th} 2D gridpoint is the L-dimensional $c_k = \Phi(x_k)$ vector, where x_k is the k^{th} column of matrix X. The basis functions can be arbitrarily chosen among smooth functions. Throughout this paper, we used radial basis functions with constant unit variance. Our task is to estimate the probabilities of the latent variables generating a given document.

3.3 Data models

3.3.1 Independent Bernoulli model

In case of binary represented data (each document is a binary vector where the occurrence of a term is denoted by 1, while the absence by 0), the independent and identically distributed Bernoulli noise model is the usual assumption. The binomial Bernoulli distribution is one of the discrete members of the exponential family, specifying the cumulant function as:

$$G(Ac_k) = \log(1 + \exp(Ac_k))$$

Thus, the nonlinear projection function (link function) of the generative model is the gradient of G with respect to the natural parameter of the distribution:

$$m_k = g(Ac_k) = \frac{\exp(Ac_k)}{1 + \exp(Ac_k)}.$$

The well-known form of the Bernoulli distribution can be written in the form:

$$p(d_n|c_k) = \prod_{t=1}^{T} m_{tk}^{d_{tn}} (1 - m_{tk})^{1-d_{tn}}.$$

3.3.2 Multinomial model

Similarily, the multinomial distribution is another member of the exponential family, which is specified by

$$G(Ac_k) = \log\{\sum_{t=1}^{T} \exp(A_t c_k)\},$$

where A_t is the t-th row of matrix of parameters A. Thus, the link function in this case is:

$$m_k = g(Ac_k) = \frac{\exp(Ac_k)}{\sum_{t=1}^{T} \exp(A_t c_k)}.$$

Consequently, the form of the distribution is:

$$p(d_n|c_k) = \exp\{\sum_{t=1}^{T}(d_{tn}\log(m_{tk}))\} = \prod_{t=1}^{T} m_{tk}^{d_{tn}}.$$

4 EM algorithm for discrete data

The log-likelihood function of the observables is given as

$$l = \sum_{n=1}^{N}\log\{p(d_n)\} = \sum_{n=1}^{N}\log\{\sum_{k=1}^{K} p(d_n|c_k)P(c_k)\}$$

where $(c_k)_{k=1..K}$ denote the hidden variables (the clusters in case of the latent class model and the nonlinear images of the hidden grid point variables for the latent trait model).

In order to determine the parameters of the models, we need to maximize the data likelihood or its log-likelihood equivalent. This maximization can be performed using the **Expectation Maximization (EM) algorithm**. This algorithm *iteratively* modifies the parameters to increase the likelihood.

The EM algorithm performs the following two steps, in each iteration:

- In the **E-step**, sets the values of the parameters and maximizes the likelihood by the $p(c|d)$ posteriors.
- In the **M-step**, estimates the new values of the parameters using the previously optimized posteriors. These parameter values will be used in the E-step of the next iteration.

This is preceded by randomly setting the parameter values. It's important to mention that initial values of the parameters greatly influence which local minimum is found [4].

Instead of maximizing the log-likelihood, it is more convenient to maximize the relative likelihood, which does not contain the log of a sum. The form of the relative likelihood between old and new parameters is:

$$Q = \sum_{n=1}^{N}\sum_{k=1}^{K} p^{old}(c_k|d_n)\log\{p^{new}(d_n|c_k)P^{new}(c_k)\}$$

where $r_{kn} := p^{old}(c_k|d_n)$ is estimated in the **E**-step, being constant in the maximization step, and represents the *"responsibility"* of class k in generating the observation d_n.

In the latent trait model, the **M**-step of the algorithm consists of maximizing Q in the latent trait parameter A. For clustering, we are mainly interested in determining m_k $(k = 1..K)$ [7].

4.1 The EM algorithm

Denote by R_{init} the initial believes of the class-memberships over the dataset, and by D_{init} the subset of data for which we have some initial labels. If no prior information exist, they will be set randomly.

The steps of the algorithm are the following:

- **Initialization:**
  - for the latent class model:

$$M = D_{init} R_{init}^T E_{init}^{-1}; \qquad P(c_k) := \frac{1}{K}$$

  - for the latent trait model:

$$A = random; \qquad P(x_k) := \frac{1}{K}$$

- **Iterate until convergence:**
  - **E-step**: estimate the responsibilities of the latent variables

$$r_{kn}^{class} := p^{old}(c_k|d_n) = \frac{p(d_n|c_k)P(c_k)}{\sum_{k'=1}^{K} p(d_n|c_{k'})P(c_{k'})}$$

$$r_{kn}^{trait} := p^{old}(x_k|d_n) = \frac{p(d_n|x_k)P(x_k)}{\sum_{k'=1}^{K} p(d_n|x_{k'})P(x_{k'})}$$

  - **M-step**:
    - parameter update:
      - for the latent class model

$$M = DR^T E^{-1}$$

      - for the latent trait model (iterate until convergence)

$$\Delta A \propto (DR^T - g(AC)E)C^T$$

    - not obligatory, but the update of the latent priors is recommended

$$P^{new}(c_k) = \frac{1}{N} \sum_{n=1}^{N} r_{kn}^{class}$$

$$P^{new}(x_k) = \frac{1}{N} \sum_{n=1}^{N} r_{kn}^{trait}$$

5 Estimation of cluster numbers with Genetic Chromodynamics

Clustering requires the number of clusters to be given beforehand, i.e. how many latent variables do we suppose to have generated the text documents. However, we can set this value arbitrarily, the process resulting in more or less refined clusters. On the other hand, clustering results greatly depend on initialization values i.e. clustering the same dataset over and over, we may obtain slightly different classifications although the number of clusters remains the same.

Originally, we used *k-means clustering* to obtain initial cluster centers. To tackle both problems (find the optimal number of clusters and make clustering more consistent), we experimented with *genetic chromodynamics* [13] to see if it is a viable solution. Standard genetic algorithms fail to detect multiple optimum points, on the other hand premature local convergence does present difficulty. The strategy we have considered is a flexible method intended to solve multimodal optimization and search problems in distributed artificial intelligence applications, like cooperative multi-agents.

5.1 Genetic Chromodynamics principles

The main idea of this approach is to force the formation and maintenance of subpopulations of solutions. These subpopulations co-evolve and eventually converge towards several local and global optimal solutions. The population size decreases with time as similar solutions are merged into a single one, that can be considered either the fittest or the mean of the individuals to be merged. The final population contains the detected optimum points.

It uses a local interaction principle according to which the formation and evolution of solution subpopulations are favored. The recombination mate of a given individual is selected within a given *mating region*. For selecting a mate, we used roulette wheel selection, according to which the probability of selection is proportional to an individual's fitness. If no mate can be found in the neighborhood, it will suffer mutation. Thus mutation and recombination are mutually exclusive operators. To prevent migration between subpopulations and the extinction of some optimum points, we may expect that mutated offsprings have to belong to the mating region of their parents.

After the formation of the new generation one last operation is performed on its individuals: if distance between solutions is less than a given threshold, called *merging radius*, then the individuals will be merged. The solution with the best fitness value will be kept, and all the others will be deleted from the population.

5.2 Initial population

Thus each row of the data matrix D will represent an individual of the initial population:

$$d_i = (d_{i1}, \cdots, d_{iT}),$$

where $d_{ij} \in \Re, i = 1..N, j = 1..T$, N representing the number of chromosomes (*i.e.* documents) and T the number of genes (*i.e.* words).

5.3 Fitness assignment and stop condition

The starting-point of several clustering algorithms is the definition of distance between the objects to be clustered. Thus the precise definition of similarity or distance is crucial. The similarity of two documents can be quantified using the simple Euclidean distance, the Pearson correlation, Spearman rank correlation or tailored measures [12]. In our application, we used the Manhattan distance. Thus the fitness function has the following form:

$$eval(d_i) = e^{-\sum_{j=1}^{N} dist(d_i,d_j)},$$

where $dist(d_i, d_j) = \sum_{j=1}^{N} |d_i - d_j|$ the Manhattan distance.

The algorithm stops when, after a predefined number of iterations (usually proportional to the number of documents in the data set), there is no significant change in the population. The number of individuals in the final population gives the optimal cluster number, while its members correspond to the centers (representative documents) of the resulting clusters (topics).

6 Results and Conclusions

In order to illustrate the models described in this paper and examine their efficiency, we developed an application that solves two text mining problems: clustering and visualization of text documents, and boosted the performance of LVM-based clustering by automatically extracting the topic numbers from the data set itself. For the latter, we turned to a novel clustering evolutive algorithm, based on genetic chromodynamics.

As corpus, we selected different documents provided by **20-Newsgroups** [1], whose text base consists of 20000 text files resulted from communication over the internet (e-mail), covering various topics grouped in 20 classes. Some topics of interest: "alt.atheism", "talk.politics.gun", "talk.politics.misc", "comp.windows.x", "comp.sys.ibm.pc.hardware", "comp.graphics", "comp.sys.mac.hardware", "rec.sport.hockey", "sci.crypt", "sci.space". The diversity of subjects and the number of documents available makes it possible to test the algorithms for clustering and 2D visualization.

All documents consist of two parts: a header containing information about how the e-mail came into existence and the topic it was assigned to – information we disregard in the application – separated of the actual text holding valuable data, by a blank row. For example, the content of the document `53326` categorized as belonging to the `talk.politics.guns` topic, is:

```
Path: cantaloupe.srv.cs.cmu.edu crabapple.srv.cs.cmu.edu
europa.eng.gtefsd.com elroy.jpl.nasa.gov
From: franceschi@pasadena-dc.bofa.com
Newsgroups: talk.politics.guns
Subject: Re: Gov't break-ins (Re: 60 minutes)
Message-ID: <1993Apr5.155733.114@pasadena-dc.bofa.com>
Date: 5 Apr 93 15:57:33 PDT
Organization: Bank America Systems Engineering, Pasadena, CA
Lines: 20

On a Los Angeles radio station last weekend, the lawyers for
the family of the MURDERED rancher said that the Los Angeles
```

```
Sheriff's Department had an assessment done of the rancher's
property before the raid.

This strongly implies that the sheriff's department wanted the
property; any drugs (which were not found) were only an excuse.

In Viet Nam, Lt Calley was tried and convicted of murder
because his troops, in a warsetting, deliberately killed
innocent people. It is time that the domestic law enforcement
agencies in this country adhere to standards at least as moral
as the military's.

Greed killed the rancher, possibly greed killed the Davidian
children. Government greed.

It is time to prosecute the leaders who perform these
invasions.

Fred Franceschi   (These are my own opinions!)
```

The goal of our experiments is to investigate the potential power of the latent trait and class model in discovering meaningful semantical organizations, word groupings in a corpus of text-based documents, on one hand; on the other hand, to check which data model is more suitable to our needs – whether or not the multinomial model indeed outperforms the Bernoulli model, exploiting the plus of information granted by representation.

In our application, we implemented that when clicking on a symbol, certain information regarding the represented text document is displayed: its name, topic (original label), the words it contains – from the dictionary – and their occurrences, and in case of clicking on a figure visualizing results of clustering, what cluster is it associated to.

6.1 Experiment: Clustering

To investigate the efficience of clustering with multinomial model, we compared the classification based on the original labels and the ones provided by the clustering algorithm. Four document classes have been chosen, such that two of them were strongly overlapping: "talk.politics.guns", "talk.politics.mideast", "rec.sports.baseball" and "sci.crypt". 100 documents were selected arbitrarily from each of these classes (preprocessing of documents was achieved with the help McCallum's program, *Bow* [9]) and a dictionary of 200 words was built up based on word frequency.

Conspicuously, documents with "similar" content tend to conglomerate, while those pertaining to different topic categories are mapped further away in the plane. Taking a closer look at the list of most probable words, one can discover meaningful word groupings.

Clustering requires the number of clusters to be given, i.e. how many latent variables do we suppose to have generated the text documents. However, we can set this value arbitrarily. For example, we can classify the documents of the four topics into six classes, as it is shown in the next figure. Notice that such a grouping results in the refinement of

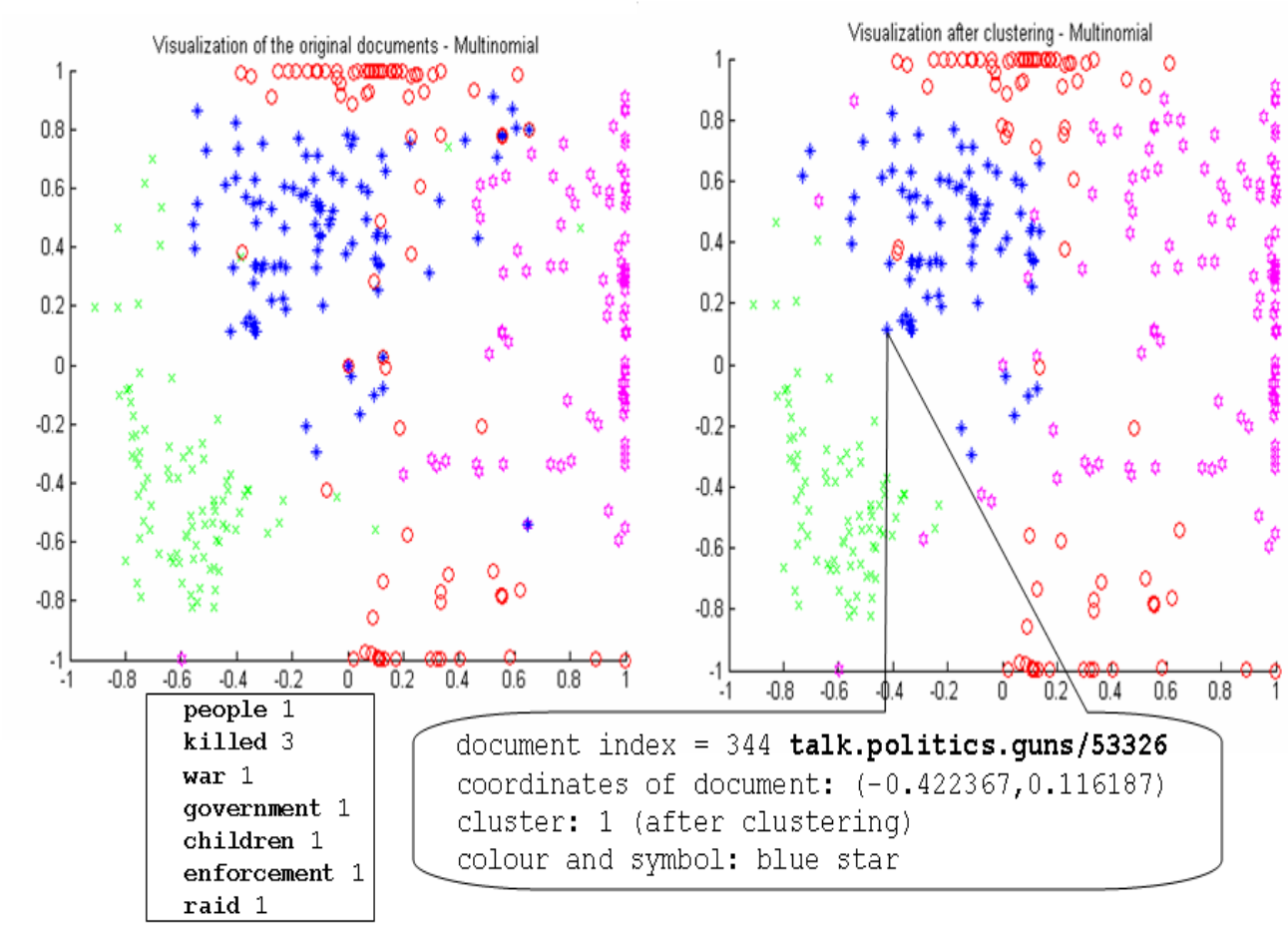

Figure 1: Document clustering and visualization: Posterior means in the case of multinomial data model using 4 document classes from the 20-Newsgroups corpus: "talk.politics.guns", "talk.politics.mideast", "rec.sports.baseball" and "sci.crypt".

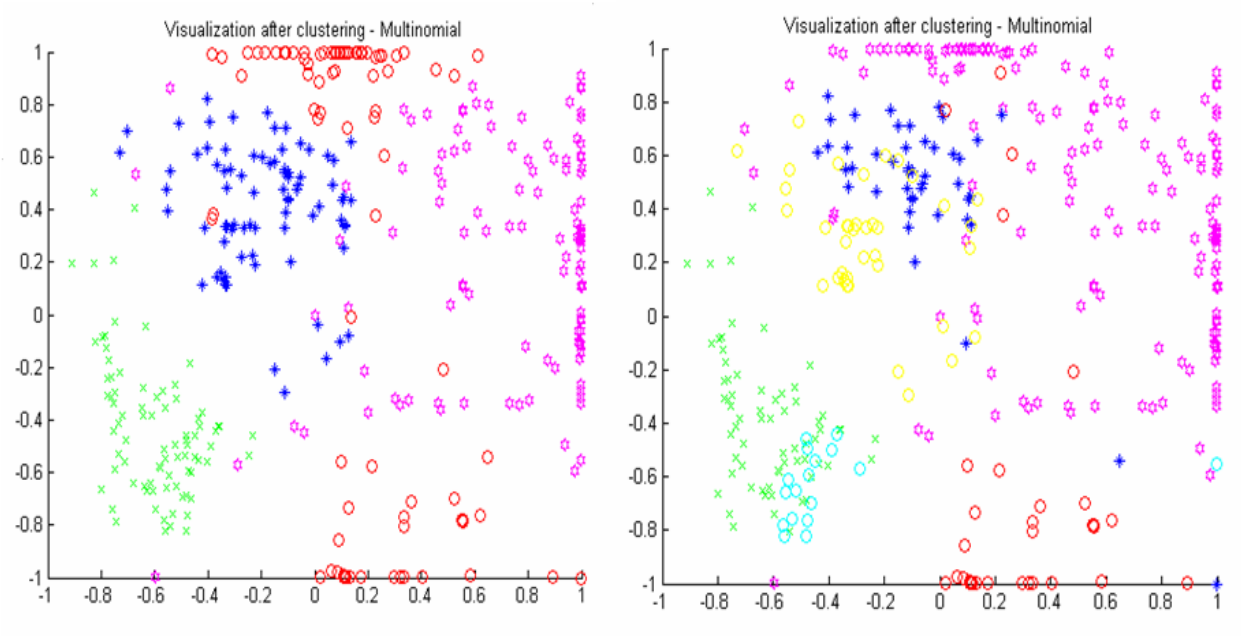

Figure 2: Document clustering: The number of clusters can be arbitrarily chosen (left picture: K=4, the right: K=6).

the classification: we can discover smaller mounds within the green and blue clusters. Nevertheless, the self-organizing clustering algorithm may also find different clusters, besides refinement of classes. For instance, the red and magenta classes are delimited differently in the case of four, respectively six presumed classes.

As we observed, the self-organizing clustering algorithm assigns a document to a cluster other than what one might expect, based on its original label, when the document doesn't contain words relevant to the topic or it also contains key words from another topic. At the same time, we must emphasize that the assignment of a document to a cluster is soft, not categorical: it belongs to every cluster with a certain probability.

6.2 Experiment: 2D Visualization

The ultimate scope of our visualization application is to create a 2D representation for a document corpus easily understandable and interpretable by humans, that captures as much as possible of the important structure of the data i.e. similar text document instances are mapped close to each other while unrelated instances are mapped further away in the latent space. This map can be used to generate an illustrative graphical display of the data set which facilitates information retrieval and exploring the topical content of the document collection.

In order to evaluate the latent variable model for count based text representation, the same documents and dictionary was used as in the clustering experiment. In all experiments, a 10×10 uniform grid ($K = 100$) mapped onto 25 Gaussian basis functions of unit variance has been chosen, thus we projected the high dimensional data – documents from the vector space representation – onto only 2 dimensions through latent dimension 25. The parameter matrix has been initialized randomly.

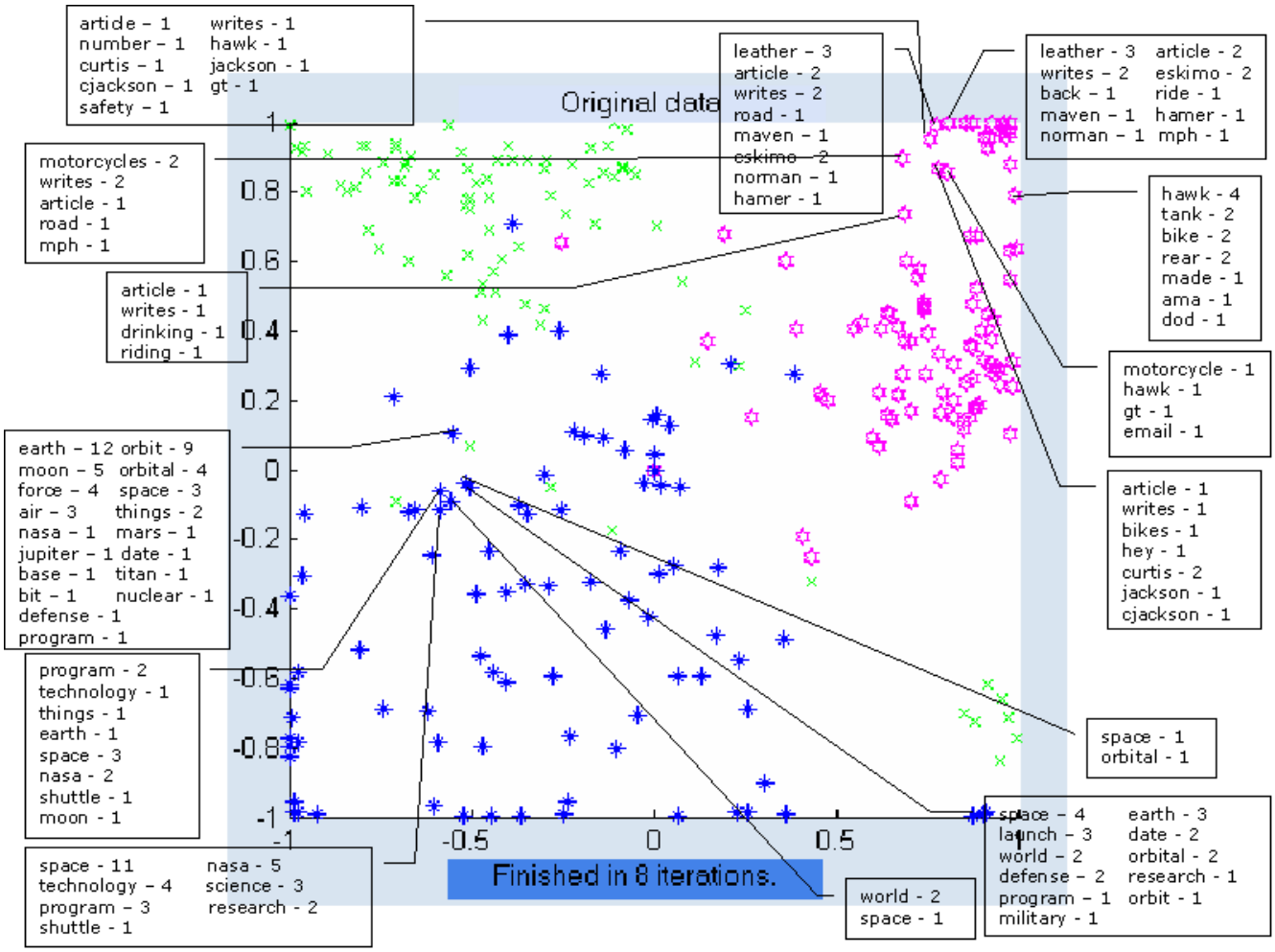

Figure 3: Document clustering and visualization: visualization in the case of multinomial data model on two overlapping ("talk.politics.guns", "talk.politics.mideast") and two distinct ("rec.sports.baseball" and "sci.crypt") classes from the 20-Newsgroups corpus, and inspecting some of the documents for the contained words and their occurrences.

Inspecting the lists of the most probable words, associated to the different grid points – i.e. the top of the reference vectors $g(Ac_k)$ – one can recognize meaningful word groupings: neighbouring grid points have many words in common and words grouped together arise from the same subject area. Thus , "similar" documents tend to form mounds, while documents with different topic are located further apart from each other.

One can also observe that grid points that lie in one of the classes' central regions store meaningful keywords in their lists of the most probable words, whereas those lying in an overlapping area store words that occur in documents concerning the junction of these discussion topics, thus demonstrating that the presented model can indeed consistently reveal the semantic structure of the corpus on a 2D map.

6.3 Experiment: Multinomial vs. Bernoulli

Based on our experiments, we can draw the conlclusion that the latent class and trait models are suitable for topographical visualization and semantic structure discovery from a text collection, for both the binary and count based text representations. One can see that both data models have found a valid meaningful mapping, as related document instances are clumped together on the maps, while those concerning different topics are mapped further from each other. However, in the differentiation of the two overlapping classes, the multinomial model's result is clearly superior: the mapped corpus has organized in more separated regions or classes. This difference can also be seen from the clustering of documents. This result was expected as the frequency count information gives a more visual discrimination between possibly overlapping classes than which binary occurrence information.

7 Conclusions

The class and trait models have been described, which are suitable for analysis, clustering and visualization of corpora containing text documents. Application of the proposed models for text mining was also presented, and a readily interpretable representational structure was obtained on text based documents. In the models described in this paper, the assumption of data instances being independent and identically distrtributed was made. These methods are useful for any such data analysis applications where the visualization and clustering of possibly sparse multivariate discrete data set is sought.

Examining whether genetic chromodynamics automatically provides cluster centers that meet previously described requirements was also set out and confirmed by the experiments.

References

[1] *The 20 Newsgroups dataset*, http://www.cs.cmu.edu/~TextLearning/

[2] **Bishop, C. M.**, *Latent variable models, Learning in Graphical Models*, MIT Press, 1999, 371-403

[3] **Chakrabarti, S.**, *Mining the Web, Discovering Knowledge from Hypertext Data*, Morgan Kaufmann Publishers, 2003

[4] **Dellaert, F.**, *The Expectation Maximization Algorithm*, Technical Report number GIT-GVU-02-20, 2002

[5] **Fayyad, U., Grinstein, G. G., Wierse A.** (editors), *Information visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann Publishers, 2001

[6] **Han, J., Kamber, M.**, *Data Mining. Concepts and Techniques*, Elsevier Inc., 2001

[7] **Kabán Ata, Girolami Mark**, *A Combined Latent Class and Trait Model for the Analysis and Visualisation of Discrete Data*, IEEE Transactions on Pattern Analysis and Machine Intelligence 23(8), UK, 2001

[8] **Kabán Ata**, *Latent Variable Models with Application to Text Based Document Representation*, PhD Thesis (Computer Science), University of Paisley, UK, 2001

[9] **McCallum, A.**, *Bow, A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering*,
http://www-2.cs.cmu.edu/~mccallum/bow/, 1998

[10] **Nabney, I. T.**, *NETLAB, Algorithms for Pattern Recognition*, Springer, UK, 2002

[11] **Nabney, I. T., Bishop, C.** , *NETLAB neural network software - toolbox of Matlab functions and scripts*,
http://www.ncrg.aston.ac.uk/netlab/index.html, 2002

[12] **Coley, D. A.**, *An introduction to genetic algorithms for scientists and engineers*, World Scientific, 1999

[13] **Dumitrescu, D.**, *Genetic chromodynamics*, Studia Universitatis Babes-Bolyai, Ser. Informatica, 2000, 39-59