Chapter 17

From Data to Knowledge Research Unit

Heikki Mannila, Jaakko Hollmén, Kai Puolamäki, Gemma Garriga, Jouni Seppänen, Robert Gwadera, Sami Hanhijärvi, Hannes Heikinheimo, Samuel Myllykangas, Antti Ukkonen, Nikolaj Tatti, Jarkko Tikka

17.1 Finding and using patterns

Finding trees

Hannes Heikinheimo, Jouni Seppänen, Nikolaj Tatti, Heikki Mannila

One of the most active topics in mining of binary data is finding interesting itemsets. A traditional approach is to search for frequent itemsets. In such sets the items co-occur frequently. While this definition of importance has many nice theoretical and practical properties it has one serious drawback: A frequent itemset, say AB, is less interesting if the individual attributes A and B are frequent. On the other hand, if AB is infrequent and A and B are frequent, then the itemset AB should be interesting.

One commonly used approach for defining the importance of the itemset is to compare the frequency against the independence assumption. The more the itemset deviates from the independence assumption, the more interesting it is. An alternative way of looking at this approach is to think that we are predicting the frequency an itemset from its individual attributes. In [1,2] we expand this idea by using the itemsets for the prediction. That is, we compare the itemset against a prediction based on a given family of itemsets. For prediction we use Maximum Entropy, a popular method for estimating distributions. For the comparison we use Kullback-Leibler divergence. We point out that this ranking method should be normalized and the normalization can be used as a statistical test.

We tested our methods with several real-world datasets. In our experiments we found out that a surprisingly large portion of itemsets that are important according to the independence model becomes unimportant when we are using larger itemsets for the prediction. However, in some cases using less itemsets produces a better ranking. This interesting phenomenon is a type of overlearning that occurs when too many itemsets mislead the prediction and model the noise in data (see Figure 17.1). To remedy this behavior we suggest in [1] a greedy algorithm that automatically picks the lowest rank for the itemset.

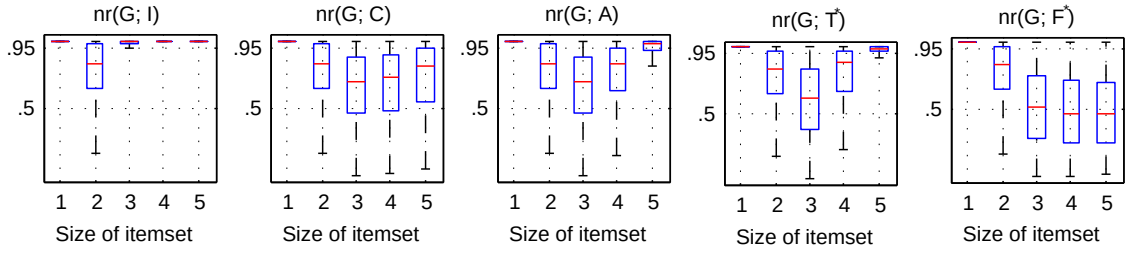


Figure 17.1: Box plots of the rank measures computed from *Paleo*. The y-axis is the importance of the itemsets. The prediction models from left to right are: the independence model, the Gaussian model, prediction based on all sub-itemsets, the best tree model, and the best model found by the greedy algorithm.

In [3] we propose a new class of co-occurrence patterns: trees. The idea is to search for hierarchies of general and more specific attributes. The novelty in our approach is that we start from unordered data, and by using frequent pattern mining techniques infer hierarchical orders from data using a specific pattern scoring function.

Tree pattern mining has interesting applications in domains, such as text mining, where such tree patterns may reflect interesting co-occurrences between usages of terms. Figure 17.2 shows examples of such patterns discovered form a real text data sets of scientific research abstracts.

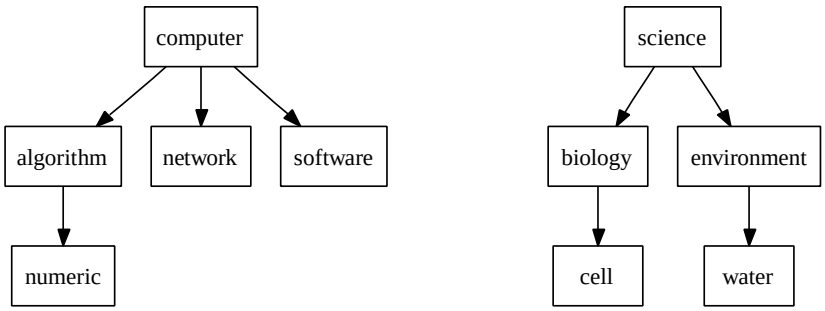


Figure 17.2: Two frequent trees patterns discovered from a text corpus of scientific research abstracts.

In [4] we study the use of entropy as a scoring function for frequent patterns. Using entropy we define low-entropy sets, a more general and expressive pattern class that of frequent sets. We show that entropy has many desired properties, such as the basic monotonicity, that allows to use the levelwise approach to efficiently discover all low-entropy patterns given some entropy threshold. Furthermore, we use entropy to defining a new tree pattern class, low-entropy trees, which can be seen as a probabilistic variant of the frequent tree pattern class defined in [3].

References

[1] Nikolaj Tatti. Maximum entropy based significance of itemsets. In *Proceedings of Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 312–321, 2007.

[2] Nikolaj Tatti. Maximum entropy based significance of itemsets. *Knowledge and Information Systems*. In press.

[3] H. Heikinheimo, H. Mannila, and J. K. Seppänen. Finding trees from unordered 0-1 data. In *Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 175–186, 2006.

[4] H. Heikinheimo, E. Hinkkanen, H. Mannila, T. Mielikäinen, and J. K. Seppänen. Finding low-entropy sets and trees from binary data. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 350–359, 2007.

Using patterns

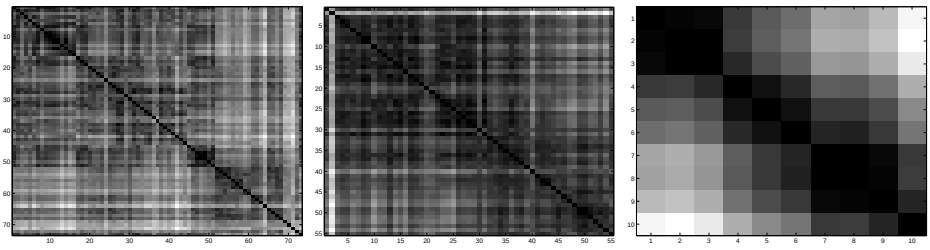
Nikolaj Tatti

A single itemset is a local pattern. It only represents a part of transactions and the information contained within the itemset is limited. However, a group of itemset can be very descriptive. In fact, if the group is large enough it can identify the whole dataset. While this is rarely the case in practice, we can think that a smaller group of itemsets captures the essential information of the original data. One of our research topic is how we can use itemsets in various tasks as a surrogate for data.

The theoretical part of these studies is in itself interesting and important, but there is also a practical point of view when we consider these scenarios in the context of privacy-preserving data mining. A group of itemsets can be viewed as a sanitized version of the actual data and the studies describe how we can work with itemsets without having the actual data.

In [1,2] we study how we can use these sets for predicting unknown itemsets. We show that this is an NP-hard problem but with certain assumptions we can ease the computational burden. The problem reduces to a linear program with an exponential number of variables. By applying the ideas from the theory of Markov Random Fields we are able to reduce the number of attributes. We also point out that our reduction is optimal, that is, if we reduce any additional attributes, then there is a dataset for which the additional reduction will alter the prediction.

In [3] we study how we can use itemsets for defining the distance between two binary datasets. Nowadays, there is a particular need for this type of work, since the amount of information keeps growing and getting more complex. Hence, we need algorithms and theorems in which a whole a database is considered as one data point. Once we have defined a distance between databases, we can expose these to traditional data mining tools, such as, clustering and visualization. We base the definition of our distance on a geometrical intuition. We show that we can derive the same distance from a different sets of axioms, hence giving the distance a strong theoretical support. We also described an efficient method for computing the distance: The distance is an Euclidean distance between the frequencies of certain parity formulae. We apply the distance for several datasets and demonstrate that the distance produces meaningful and interpretable results (see Figure 17.3).



(a) Books of Bible (b) Inaugural addresses of U.S. Presidents (c) Abstracts of NSF awards

Figure 17.3: Distance matrices for various datasets. We see a temporal behavior in Figures 17.3(b)–17.3(c) and a division between the Old Testament and the New Testament in Figure 17.3(a).

References

[1] Nikolaj Tatti. Computational complexity of queries based on itemsets. *Information Processing Letters*, pages 183–187, June 2006.

[2] Nikolaj Tatti. Safe projections of binary data sets. *Acta Informatica*, 42(8–9):617–638, April 2006.

[3] Nikolaj Tatti. Distances between data sets based on summary statistics. *Journal of Machine Learning Research*, 8:131–154, Jan 2007.

From models to patterns

Jaakko Hollmén, Jarkko Tikka, Samuel Myllykangas

In the context of bioinformatics, we are interested in describing DNA amplification patterns recorded as 0–1 data with compact and understandable descriptions. To understand the coarse structure of the amplifications (mutation patterns in the physical chromosome), we applied probabilistic clustering of 0-1 data with a finite mixture of multivariate Bernoulli distributions [3]. Model selection was performed with cross-validation based methods.

Clustering the data with the finite mixture model achieves a good clustering with six component distributions, depicted in the Figure 17.4. In [1], we have developed a method to describe the clusters with compact and understandable descriptions by extracting maximal frequent itemsets and transforming them to the nomenclature used to describe chromosomal areas.

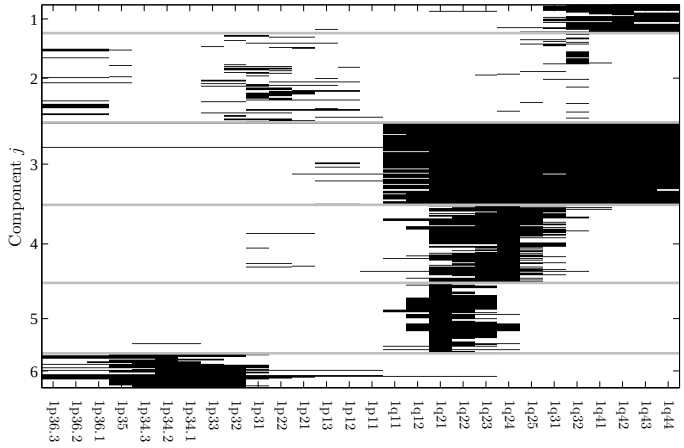


Figure 17.4: Clustered data from the human chromosome 1. Chromosomal locations are on the x-axis (marked under the figure) and the rows represent the clustered amplification patterns. Black areas are the DNA copy number amplifications. Compact and understandable descriptions are extracted from the cluster-specific data.

In a subsequent paper, the preceding results are reported in the problem of cancer classification [2]. We have clustered about 4500 cancer patients, one chromosome at the time and identified in total 111 amplification patterns in general. We investigated the associations of the amplification patterns with background factors of cancer types in order to underline the importance of a specific mutation in a particular cancer type.

References

[1] Jaakko Hollmén and Jarkko Tikka. Compact and understandable descriptions of mixture of Bernoulli distributions. In M.R. Berthold, J. Shawe-Taylor, and N. Lavrač, editors, *Proceedings of the 7th International Symposium on Intelligent Data Analysis (IDA 2007)*, volume 4723 of *Lecture Notes in Computer Science*, pages 1–12, Ljubljana, Slovenia, 2007. Springer-Verlag.

[2] Samuel Myllykangas, Jarkko Tikka, Tom Böhling, Sakari Knuutila, and Jaakko Hollmén. Classification of human cancers based on DNA copy number amplification patterns. Manuscript.

[3] Jarkko Tikka, Jaakko Hollmén, and Samuel Myllykangas. Mixture modeling of DNA copy number amplification patterns in cancer. In Francisco Sandoval, Alberto Prieto, Joan Cabestany, and Manuel Graña, editors, *Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN 2007)*, volume 4507 of *Lecture Notes in Computer Science*, pages 972–979, San Sebastián, Spain, 2007. Springer-Verlag.

17.2 Data mining theory

Cross-mining

Gemma C. Garriga, Hannes Heikinheimo, Jouni K. Seppänen

Most frequent pattern mining studies consider data sets with exclusively binary attributes. However, many real-world data sets have not only binary attributes but also numerical ones. As an example, consider movie recommendation systems: the binary data corresponds to movies rated by users, and the numerical data to their demographic statistics. A key problem is to segment users into similar groups according to their movie liking, but it is also in the interest of the service provider to produce descriptions of the groups for marketing purposes. Other such domains are for instance ecological data mining applications, where numerical environmental variates, such as rainfall or temperature, affect the binary occurrence (coexistence) of species across spatial locations.

In [1] we suggest a method for relating itemsets consisting of binary attributes to numerical attributes in the data. Our approach can be seen either as using the numerical attributes to measure the interestingness of the itemsets consisting of binary attributes, or, from another viewpoint, as using the itemsets to assess the interestingness of clusters or other local models found by mining the numerical attributes. Computing such models turns out to be computationally challenging, however, approximable within a constant factor using a simple greedy algorithm. Experiments show using biogeographical data that the algorithm can capture interesting patterns that would not have been found using either itemset mining or clustering alone.

References

[1] G. C. Garriga, H. Heikinheimo, and J. K. Seppänen. Cross-mining binary and numerical attributes. In *International Conference on Data Mining*, pages 481–486, 2007.

An approximation ratio for biclustering

Kai Puolamäki, Sami Hanhijärvi, Gemma C. Garriga

The problem of biclustering consists of the simultaneous clustering of rows and columns of a matrix such that each of the submatrices induced by a pair of row and column clusters is as uniform as possible. We have approximate the optimal biclustering by applying one-way clustering algorithms independently on the rows and on the columns of the input matrix. We have shown that such a solution yields a worst-case approximation ratio of $1 + \sqrt{2}$ under L_1-norm for 0–1 valued matrices, and of 2 under L_2-norm for real valued matrices.

Given a data matrix X, an optimal biclustering is a partition of rows and columns $\mathcal{R}$ and $\mathcal{C}$ into K_r and K_c partitions such that the cost

$$L = \sum_{R \in \mathcal{R}} \sum_{C \in \mathcal{C}} \mathcal{V}(X(R, C)),$$

is minimized, where we have used $\mathcal{V}(X(R, C))$ to denote the dissimilarity of the submatrix of X defined by the set of rows R and columns C.

Finding highly homogeneous biclusters has important applications for example in biological data analysis, where a bicluster may, for example, correspond to an activation pattern common to a group of genes only under specific experimental conditions.

We show that a straightforward algorithm, where a normal one way clustering algorithm is applied both to the rows and columns of the matrix. The scheme for approximating the optimal biclustering is defined as follows.

Input: matrix X, number of row clusters K_r, number of column clusters K_c

$\mathcal{R} = \text{kcluster}(X, K_r)$
$\mathcal{C} = \text{kcluster}(X^T, K_c)$

Output: a set of biclusters $X(R, C)$, for each $R \in \mathcal{R}$, $C \in \mathcal{C}$

---

kcluster(X, K) is a normal one way clustering algorithm, with a proven approximation ratio, which partitions the rows of matrix X into K clusters. This simple scheme gives an approximation ratio of $1 + \sqrt{2}$ for L_1 norm and 2 for L_2 norm, multiplied by the approximation ratio of the one-way clustering algorithm kcluster.

Our contribution shows that in many practical applications of biclustering, it may be sufficient to use a more straightforward standard clustering of rows and columns instead of applying heuristic algorithms without performance guarantees.

References

[1] Kai Puolamäki, Sami Hanhijärvi, Gemma C. Garriga. An Approximation Ratio for Biclustering. Publications in Computer and Information Science E13, arXiv:0712.2682v1. Accepted for publication in Information Processing Letters.

The cost of learning directed cuts

Gemma C. Garriga

Classifying vertices in digraphs is an important machine learning setting with many applications. We consider learning problems on digraphs with three characteristic properties: (i) The target concept corresponds to a directed cut; (ii) the total cost of finding the cut has to be bounded a priori; and (iii) the target concept may change due to a hidden context.

Recent learning theoretical work has concentrated on performance guarantees that depend on the complexity of the target function or at least on strong assumptions about the target function. In [1] we propose performance guarantees to learn a cut in a directed graph that only make natural assumptions about the target concept and that otherwise depend only on properties of the unlabelled training and test data. This is in contrast to related work on learning small cuts in undirected graphs where usually the size of the concept cut is taken as a (concept-dependent) learning parameter The first observation in [1] is that for learning directed cuts we can achieve tight, concept-independent guarantees based on a fixed size of the minimum path cover. We establish logarithmic performance guarantees for online learning, active learning, and PAC learning. We furthermore show which algorithms and results carry over to learning intersections of monotone with antimonotone concepts. An important contribution concerns learning algorithms able to cope with concept drift due to hidden changes in the context, i.e., the concept depends on an unobservable variable that can change over time. Worst case guarantees in this setting are related to adverserial learning.

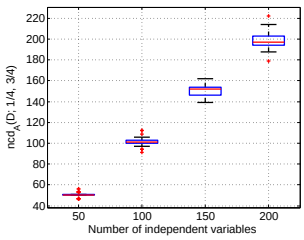
References

[1] T. Gärtner and G. C.Garriga. The Cost of Learning Directed Cuts. In *European Conference on Machine Learning (ECML)*, pages 152–163, 2007.

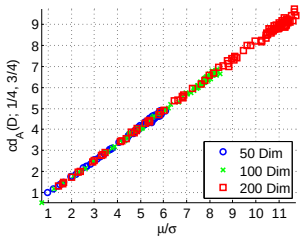
Dimensionality of data

Nikolaj Tatti and Heikki Mannila

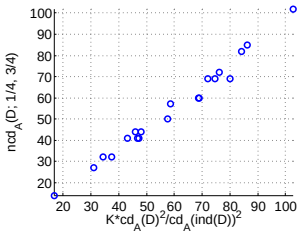
In [1] we study the intrinsic dimensionality of binary data. The idea of intrinsic dimension is rather important, since even though we may have very high-dimensional dataset, it may also possess a great amount of structure, hence its actual dimension is much smaller than the the number of attributes it is represented with. The concept of intrinsic dimension for binary data is a relatively new idea and it is an active topic in the data mining community. We apply fractal dimension — A concept that has strong theory base and has many applications with real-number datasets. We show that the fractal dimension has many interesting properties, however, the fractal dimension has some problems that are typical only to binary data. The value of the fractal dimension tend to be small for sparse binary data. Hence we tailor a new concept called the normalized fractal dimension. This dimension does not depend on the sparsity of data. In our experiments we study various properties of the dimension and compare it against several base measures.



(a) Normalized dimension, generated data



(b) Fractal dimension, generated data



(c) Normalized dimension, *20 Newsgroups*

Figure 17.5: Dimensions for various datasets. In Figure 17.5(a) we see that the normalized dimension is essentially the number of attributes if the attributes are independent. In Figures 17.5(b)–17.5(c) dimensions are plotted as functions of their estimates.

References

[1] Nikolaj Tatti, Taneli Mielikäinen, Aristides Gionis, and Heikki Mannila. What is the dimension of your binary data. In *Proceedings of Sixth IEEE International Conference on Data Mining (ICDM 2006)*, pages 603–612, 2006.

Miscellaneous problems

In [3] we introduced the discrete basis problem, a discrete version of matrix decomposition problems, and analyzed its complexity.

One of the problems in clustering is that different parameter settings can give different results. A possible solution is to use clustering aggregation, i.e., given a set of clusterings, combine them to a single clustering that agrees as well as possible with the given clusterings [2] While this problem is NP-hard, simple algorithms have a constant approximation ratio, and perform very well in practice.

Sampling in different forms is a strong technique in data analysis. The paper [1] formulates the problem of sampling hidden databases, i.e., getting unbiased samples from data sources that can be accessed only via queries. We show that the problem can, in several cases, be solved, and give simple yet powerful algorithms for the task.

References

[1] A. Dasgupta, G. Das, and H. Mannila. A Random Walk Approach to Sampling Hidden Databases. Proceedings of the 2007 ACM SIGMOD international conference on Management of Data (SIGMOD 2007), p. 629–640.

[2] A. Gionis, H. Mannila, P. Tsaparas. Clustering Aggregation (long version). ACM Transactions on Knowledge Discovery from Data, 1, 1 (2007),

[3] P. Miettinen, T. Mielikäinen, A. Gionis, G. Das, H. Mannila. The Discrete Basis Problem. 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD) 2006, p. 335–346. PKDD Best Paper.

17.3 Analyzing ordered data

Heikki Mannila, Kai Puolamäki, Antti Ukkonen

We have continued our work on developing data analysis algorithms both for finding and analyzing orders. Two results are discussed in this section, while a third, related topic is considered in the section related to randomization techniques.

Finding bucket orders

Given a set of *pairwise preferences* over a finite set of items M, we have considered the problem of ordering the items so that the resulting order agrees as much as possible with the given preferences. This problem is motivated for example by biostratigraphy, where the task is to determine the age of sediments using the fossils they contain. We use the proposed method for finding a temporal order for a number of sites (geographical locations) where fossils have been found. In this case the pairwise preferences tell us for each pair of sites which one of them is more likely to be older, and hence to precede the other one in a temporal ordering of the sites. Methods for estimating these probabilities from data based on the fossil record are considered in [1,2].

Usual approaches to combining pairwise preferences to a global order over the items try to produce a total order. This means that given the result on the fossil discovery sites, we can say for every two sites which one of them is older. There are some problems related to this, however. First, a total order may be an incorrect model class to begin with. In the fossil application it is reasonable to assume that the sites belong to certain paleontological eras. If two sites belong to the same era, it may be very difficult even for a skilled expert in stratigraphy to determine which one of the sites is in fact older. Hence, a model where the sites are not totally ordered, but placed in classes that correspond to different temporal periods is a justified approach.

Second, the preferences are likely to contain noise in some form. It is possible that two items appear to be ordered in a certain way due to random chance. Distinguishing such items from those for which the ordering is more certain is not possible from a total order. However, if we use a model that can leave some pairs of sites unordered, we can avoid errors that have been introduced to the result due to noise in the input.

Our result is an algorithm that finds a *bucket order* given the pairwise preferences. The preferences are expressed as probabilities $P(u \prec v)$ for every $u, v \in M$. This is the probability of the item u to precede item v in a global order on the items. A bucket order is a total order with ties, i.e., a disjoint partition of M to k buckets together with a total order on the buckets. In the paleontology example this means that the very oldest sites are put in the first bucket, the second oldest in the second, and so on, with the youngest sites in the last bucket.

More precisely, if the pairwise probabilities are stored in a $|M| \times |M|$ matrix C, the algorithm tries to find another matrix B that has a direct interpretation as a bucket order, and that is a good approximation of the original matrix C. In practice the algorithm minimizes a cost function of the form $|C - B|_1$, i.e., the L_1 norm between C and B. This problem turns out to be NP-hard, but the randomized algorithm we propose has an expected constant factor approximation guarantee.

C_1	C_2
8 9 10 5 12 6	5 10 4 12 1 11
9 10 5 3 11 6	5 2 3 4 12 11
8 9 3 5 11 6	6 3 8 4 11 7
9 8 5 11 12 2	5 2 4 8 1 9
7 9 10 5 3 6	2 12 8 7 9 1
7 1 5 2 6 4	2 5 4 3 7 1
1 9 3 5 2 12	5 2 6 4 12 1
9 5 2 11 12 4	2 6 10 4 9 1
7 8 3 12 2 4	5 10 3 9 7 11
9 8 3 5 2 4	5 6 10 12 9 7
9 10 3 12 2 11	5 3 8 9 11 1
7 10 2 12 11 6	5 10 3 4 1 11
7 6 12 2 11 4	2 6 3 8 9 1
1 8 11 12 6 4	2 6 8 12 7 11
7 9 8 5 6 4	5 10 12 7 1 11

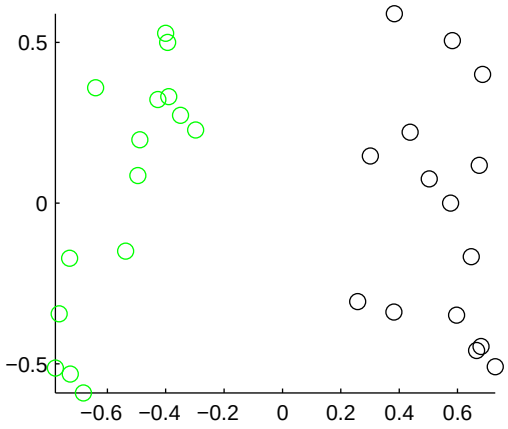


Figure 17.6: Visualizing sets of chains: On the left is a set of chains that can be divided to two groups, C_1 and C_2, based on the contents of the chains. On the right is a scatterplot where chains belonging to C_1 are indicated in green and chains belonging to C_2 appear in black. (Image from [2].)

Visualizing sets of chains

A *chain* is a totally ordered subset of a finite set of items M. For example, if M contains the title of every movie that came out in 2007, and we ask a person to rank those movies of M she has seen according to her preferences, the resulting order is a chain on M. If we ask a number of different people to rank the movies they have seen, we obtain a set of chains on the movie titles. This type of data can be used for instance to divide the respondents to a number of groups so that respondents in the same group have similar preferences. Market analysis and collaborative filtering are examples of practical applications of this approach.

We have considered different techniques for representing a set of chains as a two dimensional scatterplot where each chain appears as a single point. Our aim is to construct the visualization so that points associated to similar chains appear close to each other in the figure. Such visualizations can be useful for example for identifying structure in data or for manual classification of unseen data points.

The basic approach we take is to first map each chain of the input to a single point in a high dimensional euclidean space. Subsequently some dimension reduction method is applied to this set of points to obtain the final scatterplot. Our main contribution in this work is the development of two techniques for mapping chains to vector spaces. A central problem related to this is that comparing two chains that have no items in common is in general hard. It is possible that two chains should be mapped to points that are close to each other in the high dimensional space despite them having no common items. This happens for instance when the chains are generated by two different components; chains emitted by the same component should be placed closer to each other and away from those emitted by the other component.

Our first mapping is based on comparing each chain in the input with the other chains, and the second one maps the chains directly to points on the surface of a high dimensional hypersphere. While the first approach will place chains that have been generated by the same component close to each other, it can in practice be slow. The latter method is very

fast in comparison, but it does not attempt to recognize if two chains are emitted by the same component. In practice the two results seem to give similar results, however.

References

[1] Kai Puolamäki, Mikael Fortelius and Heikki Mannila. Seriation in Paleontological Data Using Markov Chain Monte Carlo Methods. In PLoS Computational Biology 2(2): e6, 2006.

[2] Antti Ukkonen. Visualizing Sets of Partial Rankings. In Michael R. Berthold, John Shawe-Taylor and Nada Lavrac, editors, *Advances in Intelligent Data Analysis: IDA 2007*, volume 4723 of *Lecture Notes in Computer Science*, pages 240–251. Springer, 2007.

[3] Antti Ukkonen. Algorithms for Finding Orders and Analyzing Sets of Chains. Ph.D Thesis, Department of Information and Computer Science, Helsinki University of Technology, 2008.

17.4 Randomization methods in data analysis

Heikki Mannila and Antti Ukkonen

Swap randomization of 0–1 data

Determining whether data analysis results could be the result of chance is obviously an important task. Traditional statistical significance testing methods are not very suitable for assessing the results of complex data mining operations, as the distributional assumptions behind the traditional methods are typically always false. We have been considering randomization-based methods in different contexts.

In [2] we consider a simple randomization technique for producing random datasets that have the same row and column margins with the given dataset. Then one can test the significance of a data mining result by computing the results of interest on the randomized instances and comparing them against the results on the actual data. This randomization technique can be used to assess the results of many different types of data mining algorithms, such as frequent sets, clustering, and rankings. To generate random datasets with given mar- gins, we use variations of a Markov chain approach, which is based on a simple swap operation. We give theoretical results on the efficiency of different randomization methods, and apply the swap randomization method to several well- known datasets. Our results indicate that for some datasets the structure discovered by the data mining algorithms is a random artifact, while for other datasets the discovered structure conveys meaningful information.

A similar approach is used in [1] for comparing segmentations.

Randomization of chains

A *chain* is a total order on some subset of a finite set of items M. See Section 17.3 for a more thorough description and example of a set of chains. Here we consider an algorithm for creating random sets of chains that share a number of statistics with a given set of chains.

In case of 0–1 data preserving the row and column sums is of interest as argued above. With chains we want to maintain a number of different statistics. When running a data analysis algorithm on a set of chains, we are essentially investigating the rankings, i.e., the results we obtain should be a consequence of the *ordering information* present in D. This is only one property of the input. Others are *the number of chains in the set*, *distribution of the lengths of the chains*, *frequencies of all itemsets* (when each chain is viewed as a set of items), and *the number of times the item u precedes the item v for all $u, v \in M$*.

The first property simply states that the random data sets should be of the same size as the original one. This is a very intuitive requirement. Maintaining the second property rules out the possibility that the found results are somehow caused only by the length distribution of the chains in the input. Likewise, maintaining the itemsets should rule out the possibility that the result is not a consequence of the rankings, but simply the co-occurrences of the items. Finally, maintaining the pairwise frequencies is analoguous to maintaining the mean of a set of real valued vectors.

The method we propose generates a random data set that is equivalent to the given data set if only the above properties are considered. It is a Markov Chain Monte Carlo algorithm that starts from the given set of chains and makes small local modifications to the chains that are guaranteed to preserve all of the properties above. These modifications affect two chains at a time, transposing two adjacent items in both. In general the algorithm will be run until it reaches a state that is no longer correlated with the initial state. This final state is selected as the random data set. This process is repeated until enough random data sets have been sampled.

References

[1] Niina Haiminen, Heikki Mannila, Evimaria Terzi. Comparing segmentations by applying randomization techniques. BMC Bioinformatics 2007, 8:171 (23 May 2007).

[2] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing Data Mining Results via Swap Randomization. ACM Transactions on Knowledge Discovery from Data (TKDD), Volume 1 , Issue 3 (December 2007) Article No. 14.

17.5 Applications

Ecological applications

Hannes Heikinheimo and Heikki Mannila

Many central ecological questions are related to understanding how different species communities form and what are the kind of effects climate has them. Related to this, we have applied data mining methods to study the spatial distributions of European land mammal fauna and their relationship to the environment [1]. Using clustering techniques we found that the mammalian species divide naturally into clusters, which are highly connected spatially, and that the clusters reflect major physiographic and environmental features and differ significantly in the values of basic climate variables (see Figure 17.7).

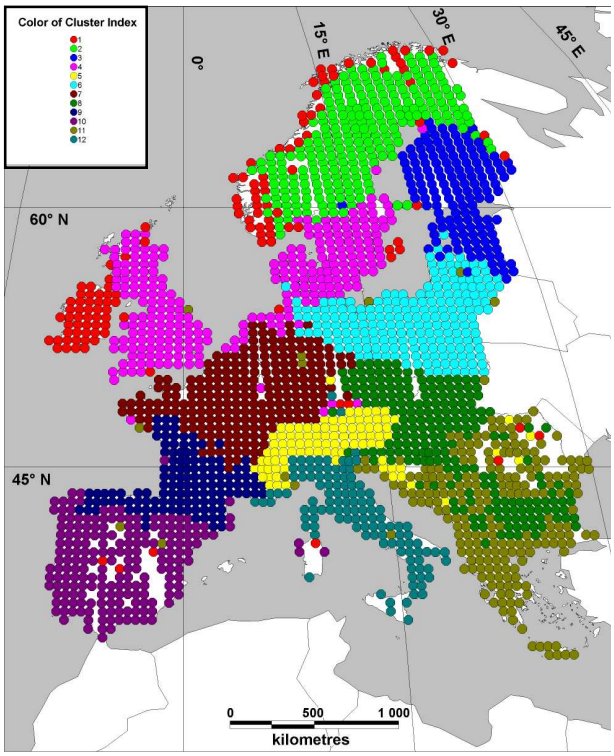


Figure 17.7: Spatial clustering of Europe using species occurrence data of 124 European land mammal species.

Our palette of applications includes a wide variety of topics. For example, in [5] we consider the problems in analyzing datasets arising in the study of linguistic change: the key issue is the small sample size for any particular period of time. The article [4] looks at haplotyping (a genetic data analysis problem) by using string models of variable length.

The ecological theme is strongly present in [2], where we study the computational properties of nestedness, a concept arising from ecology, and its generalization, segmented nestedness. These concepts turn out to be quite useful also for other applications, and their combinatorial and algorithmic properties are challenging.

References

[1] H. Heikinheimo, M. Fortelius, J. Eronen, and H. Mannila. Biogeography of European land mammals shows environmentally distinct and spatially coherent clusters. *Journal of Biogeography*, 34(6):1053–1064, 2007.

[2] H. Mannila, E. Terzi. Nestedness and segmented nestedness. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2007), p. 480–489.

[3] N. Haiminen, H. Mannila. Discovering isochores by least-squares optimal segmentation. Gene 394 (Issues 1–2), 2007, pp. 53–60 (1 June 2007).

[4] N. Landwehr, T. Mielikäinen, L. Eronen, H. Toivonen, H. Mannila. Constrained hidden Markov models for population-based haplotyping. BMC Bioinformatics 2007, 8(Suppl 2):S9.

[5] A. Hinneburg, H. Mannila, S. Kaislaniemi, T. Nevalainen and H. Raumolin-Brunberg. How to Handle Small Samples: Bootstrap and Bayesian Methods in the Analysis of Linguistic Change. Literary and Linguistic Computing 22, 2 (June 2007) 137–150; doi: 10.1093/llc/fqm006

Seriation of paleontological data

Kai Puolamäki and Heikki Mannila

Seriation, the task of temporal ordering of fossil occurrences by numerical methods, and correlation, the task of determining temporal equivalence, are fundamental problems in paleontology. With the increasing use of large databases of fossil occurrences in paleontological research, the need is increasing for seriation methods that can be used on data with limited or disparate age information.

We have developed a simple probabilistic model of site ordering and taxon occurrences. As there can be several parameter settings that have about equally good fit with the data, we have developed Bayesian Markov chain Monte Carlo methods to obtain a sample of parameter values describing the data. As an example, the method is applied to a dataset on Cenozoic mammals.

The orderings produced by the method agree well with the orderings of the sites with known geochronologic ages.

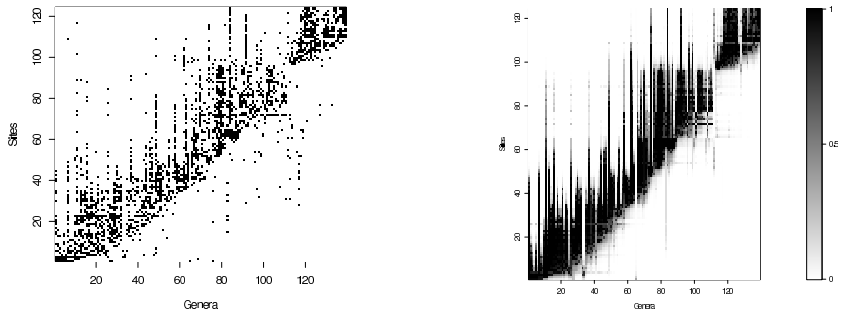


Figure 17.8: The original paleontological data matrix, where the rows correspond to the find sites and columns to the genera, and black that a genus has been found from the find site (left), as well as the probability that the genus existed during the time period of a given site (right).

References

[1] Kai Puolamäki, Mikael Fortelius, Heikki Mannila. Seriation in Paleontological Data Using Markov Chain Monte Carlo Methods. PLoS Computational Biology 2(2): e6, 2006.

17.6 Segmentation

Heikki Mannila and Robert Gwadera

For sequential data, segmentation is the counterpart of clustering. We have in recent years studied different aspects of segmentation, both theory and practice. The basic segmentation problem can be solved in polynomial time by using dynamic programming, and there are several interesting variants for study.

Segmental prediction is applicable in situations where the phenomenon of interest is governed by different models at different points of time. Such phenomena occur naturally in, e.g., atmospheric data, where for example winter and summer conditions for aerosol formation differ qualitatively and quantitatively. In [1] we studied the combination of dynamic programming and facility location approaches to obtain a small set of recurring models to be used in prediction.

A biological application of dynamic programming for the discovery of isochore structure is given in [3], while [4] looks at the randomization models needed for comparing segmentations. In [2] we study the segmentation of models of different depth for segmenting strings, especially DNA.

References

[1] S. Hyvönen, A. Gionis, H. Mannila. Recurrent predictive models for sequence segmentation. Advances in Intelligent Data Analysis VII (IDA 2007), p. 195–206.

[2] R. Gwadera, A. Gionis, H. Mannila. Optimal Segmentation using Tree Models. 2006 IEEE International Conference on Data Mining, p. 244–253, 2006

[3] N. Haiminen, H. Mannila. Discovering isochores by least-squares optimal segmentation. Gene 394 (Issues 1–2), 2007, pp. 53–60 (1 June 2007).

[4] Niina Haiminen, Heikki Mannila, Evimaria Terzi. Comparing segmentations by applying randomization techniques. BMC Bioinformatics 2007, 8:171 (23 May 2007).