

Cross-mining Binary and Numerical Attributes

Gemma C. Garriga Hannes Heikinheimo Jouni K. Seppänen
 HIIT Basic Research Unit, Lab. Computer and Information Science
 FI-02015 Helsinki University of Technology, Finland
 firstname.lastname@tkk.fi

Abstract

We consider the problem of relating itemsets mined on binary attributes of a data set to numerical attributes of the same data. An example is biogeographical data, where the numerical attributes correspond to environmental variables and the binary attributes encode the presence or absence of species in different environments. From the viewpoint of itemset mining, the task is to select a small collection of interesting itemsets using the numerical attributes; from the viewpoint of the numerical attributes, the task is to constrain the search for local patterns (e.g. clusters) using the binary attributes. We give a formal definition of the problem, discuss it theoretically, give a simple constant-factor approximation algorithm, and show by experiments on biogeographical data that the algorithm can capture interesting patterns that would not have been found using either itemset mining or clustering alone.

1. Introduction

One of the most extensively studied problems in binary data is to find collections of itemsets that frequently occur together [1]. After several years of research many efficient algorithms have been developed to mine frequent itemsets, e.g. Apriori [2] or FP-growth [6] among others. Other variations of the problem are mining frequent closed sets [18] or mining maximal frequent sets [3]. In all cases, finding such frequent collections of items efficiently for the binary attributes of the data is not an issue anymore. In fact, it can be argued that when real-world data sets are concerned, the original Apriori algorithm can find in a short time a much larger collection of itemsets than is needed for applications such as classification [21]. However, defining good interestingness measures for the final patterns seems to be a hard problem that typically depends on the targeted application.

Many real-world data sets have not only binary attributes but also numerical ones. Therefore it would be sensible to connect the two kinds of attributes when mining the data.

One possibility is to use the numerical attributes to measure the interestingness of the itemsets consisting of binary attributes, or, from another viewpoint, to use the itemsets to assess the interestingness of clusters or other local models found by mining the numerical attributes. For example, consider spatial data where species occur over a set of locations. A relevant goal in such spatial data is to uncover how environmental variates, such as rainfall or temperature, affect the coexistence of species across the locations.

As another example, consider movie recommendation by collaborative filtering [8]. The binary data corresponds to movies rated by users, and the numerical data to demographic statistics of those same users. A key problem is to segment the market into groups of customers who like similar movies, but it is also in the interest of the service provider to produce descriptions of the groups for marketing purposes. To enable this kind of descriptions one must combine the binary movie rating data with the numeric demographics, which is exactly our cross-mining task.

Motivated by such applications calling for segmentations of the data characterized by relevant features, we develop on the idea that itemsets found within the binary attributes of the data reflect structure embodied in the real-valued attributes in the same data. A natural kind of such structure is clustering, but other examples include approximating the real-valued points with a low-dimensional subspace, or having low fractal dimension [4]. Given any such class of local models for the numerical data, each corresponding to an itemset in the binary data, we wish to cover the observations with itemsets, each supported by a set of rows whose numerical attributes are well explained by a local model.

To make the discussion more concrete, we now focus on a simple biogeographical case. Imagine a rectangular grid drawn on a map, each of whose cells corresponds to a row in a database, recording what bird species occur in the cell, and the geographical coordinates of the cell as well as the average yearly rainfall within the cell. The bird species data is binary in nature, while the coordinates and rainfall attributes are numerical. An itemset X (of birds) defines a *segment* of cells, consisting of those cells where all the birds

in X occur. For each segment we can fit a local model. If our models are defined by means, the model corresponding to the segment defined by X tells us the centroid of the cells where all birds in X co-occur, and the average rainfall in these cells. If the birds occur close together and in areas with similar rainfall, this model is a good fit to the segment.

Once we have mined all frequent itemsets (or, e.g., closed itemsets), we seek to select k itemsets whose segments cover the numerical data with as well-fitting models as possible. Formally this corresponds to minimizing the error when each tuple is modeled by the best itemset model from the solution set. Other variations of this optimization problem, such as partitioning or when k does not need to be specified a priori are also natural. We will refer to these optimization problems as *itemset segmentation problems*.

This itemset segmentation problem has an apparent similarity to the segmentation problems defined for catalog segmentation [14, 13], to facility location problems [11, 19] and also to clustering [5, 10]. Our contribution lies in the border of all these problems and moreover provides a characterization of the final segments with the itemsets found on the same data. This allows for a very intuitive interpretation of our results. Another perspective is that we give a solution to the problem of ranking frequent itemsets: the measure is an itemset's relevance in describing the structure of the numerical features. Our contribution can also be seen as a new kind of constrained clustering [20], where instead of *must-link* and *cannot-link*, the constraints arise from the frequent itemsets of the binary part of the data. A thorough discussion of the related work is presented in Section 5.

By using an analysis framework based on submodular functions, we show in this paper that a simple greedy strategy obtains a solution which is within $1 - \frac{1}{e}$ for the itemset segmentation problems defined here. Unfortunately, minimizing the error function, commonly defined with L2-norm, does not satisfy the submodularity requirements to achieve such approximation bound. Thus, we transform the score function by measuring the error gap of each candidate segment (itemset) from a default worst-score model that always covers all the data. With this transformation the problem becomes that of maximizing the deviation of the solution set from the default model.

A difficulty arising when using itemsets is that a single data point may be covered by multiple itemsets. We address this problem by letting each data point select the itemset that fits it best. However, each model is selected using the whole subset of data that supports the corresponding itemset, instead of only the data whose error is computed from the model. This is because of two reasons: first, otherwise the optimal model for each data point could keep changing when other data points are assigned to models, leading to a possibly difficult optimization problem already in computing the score of the itemset; second, this solution ensures

that adding overlapping itemsets to the collection has diminishing returns, thus encouraging the algorithm to seek itemsets covering different parts of the data.

2. Description of the framework

Consider a set T of tuples where each tuple contains two types of attributes, binary-valued attributes and real-valued attributes. We refer to this vertical split on T by denoting each $t \in T$ as $t = [t_B, t_R]$, where t_B is a binary vector and t_R is a real-valued vector. Also let \mathcal{I} denote the universe of items defining the binary attributes of T . It is sometimes more convenient to refer to the binary part of each tuple as a subset of the universe of items, i.e., $t_B \subseteq \mathcal{I}$ containing all those binary attributes with value 1 in the tuple. For an itemset $X \subseteq \mathcal{I}$ and a tuple $t = [t_B, t_R]$, we will use the indicator function $t(X)$ that returns 1 if $X \subseteq t_B$, and 0 otherwise. We say that an itemset $X \subseteq \mathcal{I}$ covers all those tuples $t \in T$ such that $t(X) = 1$.

We denote a class of models by \mathcal{M} . Given an itemset $X \subseteq \mathcal{I}$, we define the segment characterized by X as $s = (X, m)$, where m is the best model from the class \mathcal{M} that can be assigned to the real-valued attributes of the tuples covered by X , that is $m = \mathcal{M}(\{t_R \mid t(X), t = [t_B, t_R] \in T\})$. To simplify the discussion we assume that the local models concern clustering around a mean, so that $m = \text{mean}(\{t_R \mid t(X), t = [t_B, t_R] \in T\})$. However, more complex models are also possible, like the local dimension of the data, or an entropy measure.

It is convenient to assume the existence of a default segment that always covers all the data in T . This will be the segment characterized by the empty set, default = (\emptyset, m_0) with m_0 being the mean of the real-valued attributes of all tuples. In this way all the tuples of the data are initially assigned to at least the default segment.

For a segment $s = (X, m)$, the score (error) of a tuple $t = [t_B, t_R] \in T$ with respect to s is defined as follows.

$$e(t \mid s) = \begin{cases} \infty & \text{if } t(X) \neq 1, \\ \|t_R - m\|^2 & \text{otherwise.} \end{cases} \quad (1)$$

The score of a tuple $t \in T$ for a set of segments $S = \{s_1, \dots, s_n\}$ is defined as the error obtained by choosing the best segment in S . As will become clear later, we define it as the reduction in error from the default model.

$$e(t \mid S) = \max_i \{e(t \mid \text{default}) - e(t \mid s_i), 0\} \quad (2)$$

So, given a fixed set of models in S , we usually say that a tuple $t \in T$ is assigned (or modeled) by the best segment (model) in S , corresponding to the argument of the maximum of Equation 2.

Note that the score $e(t | S)$ of Equation 2 could be also defined as a minimization function of the form $\min_i \{e(t | \text{default}), \{e(t | s_i)\}\}$. In such case, it would resemble the minimization function of a facility location problem [19] in which the goal is to open some number of facilities to serve customers. However, the algorithmic issues of our problem turn out to be technically quite different from those in facility location problems, but still closer to the segmentation problems presented in [14]. As in the general segmentation problems, the space of our possible decisions is exponential – arising from the exponential combinations of \mathcal{I} items that can define the segments – and thus, an analysis based on submodularity properties is clearly more appropriate.

The problem we study can be defined as follows.

Problem 1 (ITEMSET SEGMENTATION PROBLEM). *Given a set of tuples T and a set of all candidate segments C defined by a collection of itemsets, find a subset $S \subseteq C$ of size k such that the sum $\sum_{t \in T} e(t | S)$ is maximized.*

As in [14], we can also consider the equivalent partitioning version of this problem which corresponds to commuting the max operators in the maximization function. It can be defined as follows: Given a set of tuples T , a set of segments S and an integer k , find a partition of T into k sets T_1, \dots, T_k such that it maximizes the sum $\sum_{j=1 \dots k} \max_{s \in S} \sum_{t \in T_j} e(t | s)$. It is easy to see that this partitioning variant is equivalent to Problem 1.

Still more interesting however, is to consider the problem where k is not fixed a priori.

Problem 2 (ITEMSET SEGMENTATION PROBLEM, VARIABLE k VERSION). *Given a set of tuples T , a set of all candidate segments C defined by some collection of itemsets, and an integer γ , find k segments $S \subseteq C$ such that the sum $(\sum_{t \in T} e(t | S)) - \gamma k$ is maximized.*

Notice that these problems are defined for a general collection of itemsets C . In practice, this collection C may correspond to frequent sets, closed sets or maximal frequent sets mined on the binary tuples of the data. Because closed sets are a non-redundant representation of frequent sets – i.e., an itemset X is not closed if there exists a superset $X \subset Y$ with exactly the same cover [18] – it will be better in practice to directly consider frequent closed sets over the regular frequent itemsets.

To conclude this section, we discuss the complexity of the problems. Proofs of these and other theorems are omitted for space constraints.

Theorem 1. *The itemset segmentation problem is NP-hard.*

If the number of itemsets defining the possible segments S is constant, the number of possible k solutions will also be bounded constant. In such cases the problems could be

solved in time proportional to the $|T|c$, where c is the possible number of all solutions. Of course, the number of possible itemsets may be exponentially large if a very low frequency threshold has been used.

After Theorem 1 it follows now that the maximization of the variable segmentation of Problem 2 is also NP-hard. Indeed, as we will later see, the function there is a profit function which is known to be a basic NP problem.

3. Algorithms

To show approximation guarantees on the ITEMSET SEGMENTATION PROBLEM, we will follow an analysis based on submodular functions. A function is said to be submodular if it satisfies the following property: the marginal gain from adding an element s to the solution set S decreases if the element s was added on a superset of S : $f(S \cup \{s\}) - f(S) \geq f(S' \cup \{s\}) - f(S')$ for $S \subseteq S'$.

Submodular functions have been extensively studied for optimization problems. Particularly interesting for our problem is the following result shown by Nemhauser, Wolsey, and Fisher [17].

Theorem 2 ([17]). *For a non-negative, monotone submodular function f , let S be a solution set of size k obtained by selecting an element $s \notin S$ one at a time, each time choosing that element that provides the largest marginal gain $f(S \cup \{s\}) - f(S)$. Let S^* be a set that maximizes the value of f , then $f(S) \geq (1 - 1/e) \cdot f(S^*)$.*

The use of this submodularity result for Problem 1 has similar flavor as maximizing the spread influence through a social network of [12]. By showing that the function we wish to maximize here, namely $E(T, S) = \sum_{t \in T} e(t | S)$, satisfies the assumptions of Theorem 2, we can show that the following greedy algorithm has approximation guarantee $1 - 1/e$ for Problem 1.

Algorithm 1. GREEDY ALGORITHM FOR PROBLEM 1. *Given a set of tuples T and the set of all segments C defined by all possible itemsets. At all times maintain a candidate solution set S . Choose the next element $s \in C$ such that $s \notin S$ having the best marginal gain $E(T, S \cup \{s\}) - E(T, S)$ until $|S| = k$.*

We now consider the itemset segmentation problem in the variable case that an integer k is not fixed a priori to limit the solution set (Problem 2). In this case we wish to maximize a monotone submodular function minus a fixed-cost function that depends on the number of elements in the solution set penalized by a fixed constant γ . These kind of functions are typically called profit functions. The analysis we follow for this problem is exactly the same proposed in [14]. The following result is useful for our application.

Theorem 3 ([14]). For a profit function f , let S be the a solution set of size k obtained by selecting elements $s \notin S$ for which the marginal gain $f(S \cup \{s\}) - f(S)$ is at least γ . Let S^* be the minimum-size optimal solution set, then this greedy algorithm achieves a performance guarantee of at least $(\sqrt{1 + \mu} - 1)^2 / (1 + \mu)$, where $\mu = f(S^*) / |S^*|$

Given that our function $E(T, S)$ is already proven to be submodular and then, $E(T, S) - k\gamma$ is a profit function by definition, we can directly approximate our Problem 2 by means of the greedy strategy defined by Theorem 3. Moreover, by re-scaling $E(T, S)$ we can assume without loss of generality that $\gamma = 1$.

Algorithm 2. GREEDY ALGORITHM FOR PROBLEM 2. Given a set of tuples T , a fixed cost γ and the set of all segments C defined by all possible itemsets. At all times maintain a candidate solution set S . Choose the next element $s \in C$ such that $s \notin S$ a marginal gain $E(T, S \cup \{s\}) - E(T, S)$ of at least γ .

4. Experimental evaluation

To verify the applicability of itemset segmentation we implemented the proposed algorithms and tested their behaviour on various data sets. In this version of the paper we report on an experiment using presence/absence data on European mammals [16]. The geographical region under study is divided into 2183 grid cells, in which the presence of 124 species is recorded¹. As numerical attributes we used mean temperature, mean annual rain fall, mean annual temperature range and average elevation.²

Figure 1(a) shows the itemset segmentation of Europe. The top-6 segments were selected by the greedy algorithm from the 55,130 closed sets with minimum support over 25%. Each segment, represented by a closed set, uses as a model the mean of the real-valued transactions supported by the itemset. Figure 1(b) shows each one of the chosen segments separately. The final coverage is not necessarily equal to the support of the itemset characterizing the model. E.g., the default model corresponding to the empty set initially covers all data points, but most points are eventually assigned to other models.

Table 1 describes the top 6 segments of Figure 1 in terms of itemsets. The empty set corresponds to the default model that initially supports all the data points and has a final cover of 392 points. Graphically, the final coverage of the default model lies around the center part of Europe (Figure 1).

¹The information is collected by the Societas Europaea Mammalogica, www.european-mammals.org, and is used in this paper with their permission. The preprocessed binary data used here is available as part of the supplementary information accompanying the article [7].

²The climate and elevation observations were obtained from the dataset [9], available online at www.worldclim.org.

These are the points closer to the mean of the real-valued attributes of the whole data set. The segments are listed in order of addition by the greedy algorithm. As expected, the marginal gain decreases with the order.

Scandinavian countries are modeled by segments 1st and 6th: The first segment describes locations with low temperature and low altitude, while the sixth segment defines a specific subarea of the Nordic characterized by extremely low temperatures and high altitude. Elks are the prototypical mammals of the first segment, while a specific type of rabbits characterize the sixth segment. The second segment chosen by the greedy algorithm describes a flat area with high precipitation characterized by hedgehogs and mice. The third segment defines the mountain systems across Alps and Pyrenees; and the fourth segment defines southern European countries, characterized mainly by the high temperature where typically the wildcat mammals live. Notice that although coordinates were not included in the analysis of real-valued attributes, the final segments show a very good spatial coherence.

For this specific spatial data set, the segments chosen by the greedy algorithm are very stable, that is, decreasing the support (thus having more specific candidate segments) does not change the segmented areas very much. However, when using maximal frequent sets instead of closed frequent sets we obtain segments with less coverage with a better characterization of the species cohabiting in it. For example, with maximal itemsets of minimum support of 25%, the first 5 selected segments cover areas similar to the ones shown above; obviously the itemset descriptions are more specific and thus have smaller support in each one of these subareas. Only the 6th segment changes: with maximal sets, it spreads across the Dutch area of Europe characterized by different species of mice.

5. Related work

The problems defined here exhibit an apparent similarity to *facility location problems*, where one seeks to “open” some number of facilities to serve customers [11, 19] and typically, there is a fixed cost for each opened facility plus a penalty for each customer-facility distance. The nature of this problem has produced mainly results on the minimization of the proper score function. As mentioned in Section 2, the score $e(t | S)$ of Equation 2 could be also defined as a minimization function of the form $e(t | S) = \min_i \{e(t | \text{default}), \{e(t | s_i)\}\}$. Then the problem would be transformed into minimizing the sum $\hat{E}(S, T) = \sum_{t \in T} e(t | S)$. The nature of optimization problems that seek to minimize a score function is very different from the ones presented here. A good summary of the approximation bounds for facility location problems can be found in [11]. Indeed, the approximation factor obtained in this paper can

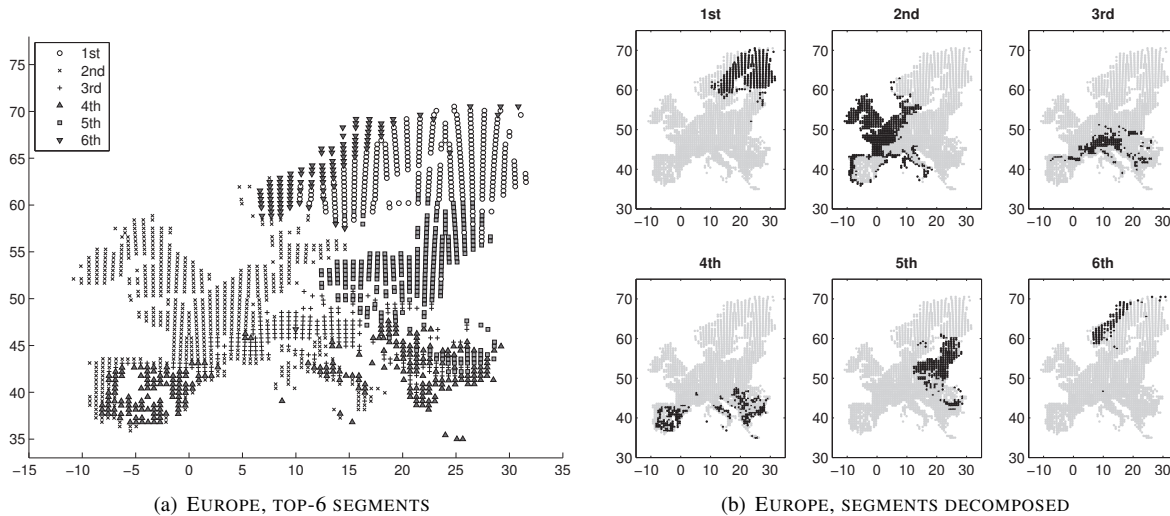


Figure 1. (a) Top 6 itemset segments selected by greedy algorithm. (b) Spatial coverage of each top segment, shown separately in order of addition by the greedy strategy. The default model, not shown, covers unassigned points.

be also expressed in terms of the minimization version of the same problem by conveniently re-scaling with the default model. It is easy to see that the approximation factor of the minimization problem is transformed then to $\hat{E}(T, S) \leq (1 - 1/e) \cdot \hat{E}(T, S^*) + 1/e \sum_{t \in T} e(t \mid \text{default})$.

Our contribution is also closely related to the *segmentation problems* from [14, 13], where one seeks to find the best segments according of an optimization function. The bounds obtained there correspond to the catalog segmentation problem, which can be seen as a generalization of the problem presented here. However, the idea of using collections of itemsets to bound the space of decisions to good models is new to the best of our knowledge. Moreover, the analysis of our greedy algorithm 1 by means of proving that our score is a submodular function is novel and still we provide with a characterization of the final segments with a set of relevant features corresponding to itemsets. Algorithm 2 stems directly from the results presented in [14].

A similar application of our contribution is clustering, an important algorithmic problem starting from [10]. As mentioned above, the partitioning version of Problem 1 can be seen as a specific clustering method. Our contribution here is then two-fold: First, from the numerous clustering methods it turns usually difficult to evaluate the quality of the partition; in our case, the partition is always properly represented by a unique itemset which defines a relevant set of features describing the partition. Second, if one wishes to cluster real attributes independently from the binary attributes, the solution might contain different partitions that

are identified by a collection of itemsets (not a single one as we propose here). Yet another drawback, different partitions would end up being characterized by the same itemset (or several similar itemsets).

The cross-mining problem can also be seen as a form of constrained clustering. In typical constrained clustering algorithms, the constraints are of the form *cannot-link* and *must-link* [20], but in cross-mining the constraints are defined by the itemset structure of the binary data. A *cannot-link* constraint that says that tuples t_1 and t_2 must belong to different clusters can be encoded by assigning disjoint binary parts to the tuples, and a *must-link* constraint can be approximated by assigning the tuples exactly the same binary attributes; however, this does not guarantee that the tuples end up in the same segment if they have very different numerical attributes. Intermediate constraints can be represented by partially overlapping binary parts. If *must-link* constraints are required in the application, they can of course be added to the cross-mining problem.

There are other works in the current literature using submodular functions to prove approximation bounds, e.g. [12]. To the best of our knowledge no contribution has been made similar to the line of this paper.

Finally, our results contribute to ranking itemsets [15], ranked in this paper by the ability to describe good partitions of the real-valued data. Hence, we showed that with a simple greedy strategy we can always provide a good approximate ranking of the best top- k itemsets in the data.

Order	Itemset	Mean model (alt, prec, temp, temp-range)	Marginal gain	Support	Final cover
–	empty set	(307.1, 716.8, 9.6, 27.3)	–	2183	392
1st	Elk	(221.6, 576.2, 1.6, 33.9)	1363.6	651	308
2nd	Hedgehog, Mouse	(150.6, 866.4, 10.6, 21.4)	898.67	1103	603
3rd	Dormouse, Vole, Roebuck	(969.2, 995.5, 7.2, 26.8)	393.76	557	224
4th	Wildcat	(514.6, 648.6, 12.5, 28.5)	237.81	624	269
5th	Raccoon dog, Brown hare	(126.5, 605.6, 7.6, 30.0)	160.36	552	289
6th	Blue hare	(693.8, 1019.1, 0.9, 25.8)	155.07	721	103

Table 1. Top 6 models for Mammals, selected from closed itemsets with support over 25%. Legend: (alt, prec, temp, temp-range) = altitude, precipitation, temperature, annual temperature range; support = # transactions where itemset occurs; final cover = # data points assigned to model.

6. Concluding remarks

We have presented the cross-mining problem of mining binary data along with corresponding numerical data. Our experiments show that the greedy algorithm finds interesting patterns that would not be discovered by itemset mining alone, which results in a huge number of patterns, nor by clustering alone (comparison experiment not shown due to lack of space). Future work includes exploration of other model classes such as the local dimension of the data, and new algorithms, such as a generalization of k -means to take the itemset constraints into account.

References

- [1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD*, pages 207–216, 1993.
- [2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo. Fast discovery of association rules. *Adv. Knowl. Disc. and Data Mining*, pages 307–328, 1996.
- [3] D. Burdick, M. Calimlim, J. Flannick, T. Yiu, and J. Gehrke. MAFIA: A maximal frequent itemset algorithm. *IEEE Trans. Knowl. Data Eng.*, 17(11):1490–1504, 2005.
- [4] C. Faloutsos and I. Kamel. Beyond uniformity and independence: Analysis of R-trees using the concept of fractal dimension. In *ACM PODS-1994*, pages 4–13, 1994.
- [5] T. Feder and D. Greene. Optimal algorithms for approximate clustering. In *ACM STOC-1988*, pages 434–444, 1988.
- [6] J. Han and J. Pei. Mining frequent patterns by pattern-growth: methodology and implications. *SIGKDD Explor. Newsl.*, 2(2):14–20, 2000.
- [7] H. Heikinheimo, M. Fortelius, J. Eronen, and H. Mannila. Biogeography of European land mammals shows environmentally distinct and spatially coherent clusters. *J. Biogeography*, 34(6):1053–1064, 2007.
- [8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.
- [9] R. J. Hijmans, S. E. Cameron, J. L. Parra, P. G. Jones, and A. Jarvis. Very high resolution interpolated climate surfaces for global land areas. *Intl. J. Climatology*, 25(15):1965–1978, 2005.
- [10] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
- [11] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k -median problems using the primal-dual schema and Lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.
- [12] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *ACM SIGKDD-2003*, pages 137–146, 2003.
- [13] J. Kleinberg, C. Papadimitriou, and P. Raghavan. A micro-economic view of data mining. *Data Min. Knowl. Discov.*, 2(4):311–324, 1998.
- [14] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Segmentation problems. *J. ACM*, 51(2):263–280, 2004.
- [15] T. Mielikäinen and H. Mannila. The pattern ordering problem. In *PKDD 2003*, pages 327–338, 2003.
- [16] A. J. Mitchell-Jones, G. Amori, W. Bogdanowicz, B. Krystufek, P. J. H. Reijnders, F. Spitzenberger, M. Stubbe, J. B. M. Thissen, V. Vohralik, and J. Zima. *The Atlas of European Mammals*. Poyser, 1999.
- [17] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Math. Prog.*, 14(1):265–294, 1978.
- [18] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *ICDT-1999*, pages 398–416, 1999.
- [19] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *ACM STOC-1997*, pages 265–274, 1997.
- [20] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k -means clustering with background knowledge. In *ICML 2001*, pages 577–584, 2001.
- [21] Z. Zheng, R. Kohavi, and L. Mason. Real world performance of association rule algorithms. In *ACM SIGKDD-2001*, pages 401–406, 2001.