HAL open science

A Study of Boolean Matrix Factorization Under Supervised Settings

Tatiana Makhalova, Martin Trnecka

▶ To cite this version:

Tatiana Makhalova, Martin Trnecka. A Study of Boolean Matrix Factorization Under Supervised Settings. ICFCA 2019 - 15th International Conference on Formal Concept Analysis, Jun 2019, Frankfurt, Germany. pp.341-348, 10.1007/978-3-030-21462-3_24 . hal-02162929v2

HAL Id: hal-02162929

https://hal.science/hal-02162929v2

Submitted on 16 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Study of Boolean Matrix Factorization Under Supervised Settings

Tatiana Makhalova[1,2] and Martin Trnecka[3]

[1] National Research University Higher School of Economics, Moscow, Russia
[2] LORIA, (CNRS – Inria – University of Lorraine), Vanduvre-lès-Nancy, France
[3] Dept. Computer Science, Palacký University Olomouc, Olomouc, Czech Republic
tpmakhalova@hse.ru, martin.trnecka@gmail.com

Abstract. Boolean matrix factorization is a generally accepted approach used in data analysis to explain data or for data preprocessing in the supervised settings. In this paper we study factors in the supervised settings. We provide an experimental proof that factors are able to explain not only data as a whole but also classes in the data.

Keywords: Boolean matrix factorization · Supervised settings · Classification· Quality of factors.

1 Introduction

Boolean matrix factorization (BMF) is a powerful tool that is widely used in data mining to describe data. It allows for data explanation by means of factors, i.e. hidden variables that rely on a solid algebraic foundation.

In general, BMF is used in the unsupervised settings, where the input data are not labeled, classified or categorized. However, evaluation of quality of generating factors did not received appropriate attention in the scientific literature on BMF. An exception is a pioneer work [4] that provides basic ideas of how the quality of BMF algorithms can be assessed in the unsupervised settings. In this paper we evaluate BMF algorithms in the supervised settings. To the best of our knowledge, the quality of factors in this settings has not been studied yet.

It was shown that BMF algorithms used as a preprocessing stage [2, 3, 18] or as neurons in a simple (one layer) artificial neural network [13] can improve classification quality. Other relevant works come from the Formal Concept Analysis [11] (FCA), since factors are often formal concepts [6]. In [1, 12, 14] closed sets of attributes, i.e. intents of formal concepts, were studied as basic classifiers (hypothesis) in different voting and inference schemes. In the mentioned studies the whole set of (frequent) factors was used to build classifiers. One may consider factors as a result of the selection of only relevant concepts (hypotheses) w.r.t. to coverage or MDL principle, e.g. in [17] MDL principle is used to select concepts that then were evaluated under supervised settings. From the FCA perspective, our study can be considered as evaluation of BMF-optimal concepts (intents or their generators) in the supervised settings. Under BMF-optimal concepts we mean those that are generated by a BMF algorithm.

Our contribution is twofold. First, we evaluate the ability of factors to explain classes of objects rather than the data as a whole. Second, we propose different models of factor-based classifiers and study their quality.

The paper is organized as follows. Section 2 introduces the used notation and the basic notions of BMF. In Section 3 we discuss how factors can be used and evaluated in supervised settings. Section 4 provides the results of a comparative study of factor sets generated by different BMF algorithms as well as evaluation of different models of factor-based ensembles of classifiers. In Section 5 we conclude and discuss direction of future work.

2 Preliminaries

In this section we recall the main notions used in this paper. Matrices are denoted by upper-case bold letters. $\mathbf{I}_{ij}$ denotes the entry of matrix $\mathbf{I}$ corresponding to the row i and column j. $\mathbf{I}_{i_}$ and $\mathbf{I}_{_j}$ denotes the ith row and jth column of matrix $\mathbf{I}$, respectively. The set of all $m \times n$ Boolean matrices is denoted by $\{0,1\}^{m\times n}$. The number of 1s in Boolean matrix $\mathbf{I}$ is denoted by $\|\mathbf{I}\|$, i.e $\|\mathbf{I}\| = \sum_{i,j} \mathbf{I}_{ij}$.

For matrices $\mathbf{A} \in \{0,1\}^{m\times n}$ and $\mathbf{B} \in \{0,1\}^{m\times n}$ we define the following element-wise operations: (i) *Boolean sum* $\mathbf{A} \oplus \mathbf{B}$, i.e. the normal matrix sum where $1+1 = 1$. (ii) *Boolean subtraction* $\mathbf{A} \ominus \mathbf{B}$, i.e. the normal matrix subtraction where $0 - 1 = 0$.

The objective of BMF is the following one: for a given Boolean matrix $\mathbf{I} \in \{0,1\}^{m\times n}$ to find matrices $\mathbf{A} \in \{0,1\}^{m\times k}$ and $\mathbf{B} \in \{0,1\}^{k\times n}$ such that

$$\mathbf{I} \approx \mathbf{A} \circ \mathbf{B}, \tag{1}$$

where $\circ$ is Boolean matrix multiplication, i.e. $(\mathbf{A} \circ \mathbf{B})_{ij} = \max_{l=1}^{k} \min(\mathbf{A}_{il}, \mathbf{B}_{lj})$, and $\approx$ represents an approximate equality assessed by $||\cdot||$, see [5] for details. The matrices $\mathbf{I}$, $\mathbf{A}$, and $\mathbf{B}$ describe the object-attribute, object-factor, and factor-attribute relations, correspondingly.

Under this model, the decomposition of $\mathbf{I}$ into $\mathbf{A} \circ \mathbf{B}$ may be interpreted as discovery of k factors that exactly or approximately explain the data, i.e. with $\mathbf{I}_{ij} = 1$ the object i has the attribute j, if and only if there exists factor l such that l applied to i and j is one of the particular manifestations of l.

3 Factors under Supervised Settings

Quality of factors is most often understood as their ability to explain data [4]. However, a lot of problems is needed to be solved in the supervised settings, where class labels of objects are available.

In supervised settings, Boolean matrix $\mathbf{I} \in \{0,1\}^{m\times n}$ corresponds to m objects described by n attributes. A special target attribute refers to an object class. More formally, we define a function *class* that maps row $\mathbf{I}_{i_}$ to its class label $c = class(\mathbf{I}_{i_}) \in \mathcal{Y}$, the size of set $\mathcal{Y}$ is equal to the number of classes.

3.1 Key Components of Classifiers

Representation and Labeling For the Boolean matrix factorization $\mathbf{I} = \mathbf{A} \circ \mathbf{B}$ we consider *factor-classifier* as a tuple (f_i, c, sim), where f_i is the ith Boolean factor (represented by the ith column and ith row of matrices $\mathbf{A}$ and $\mathbf{B}$, respectively), c is a class label given by *class* function, and *sim* is a classification strategy (see details below). In our study we assign to c a class label of the majority of objects from column $\mathbf{A}_{_i}$. If the majority is not unique, we do not consider the factor as a classifier.

Strategy of Classification We focus on two common classification strategies, namely *rule-based* and *similarity-based*.

According to the first strategy, object $g = \mathbf{I}_{j_}$ (given by n-dimensional vector) is classified by factor-classifier (f_i, c, sim) if $\mathbf{B}_{i_} \cdot g = \mathbf{B}_{i_}$, i.e. the object g has all attributes of factor f_i, "$\cdot$" denotes the element-wise multiplication.

With the second strategy, the object g is classified by factor-classifier (f_i, c, sim) if $similarity(\mathbf{B}_{i_}, g) > \varepsilon$, i.e. the attributes of factor f_i are quite similar to the attributes of object g. The similarity can be defined by means of either a distance measure or an asymmetrical operator.

It should be noted that the rule-based classification strategy is a particular case of the similarity-based one, where for $g = \mathbf{I}_{j_}$ $similarity(\mathbf{B}_{i_}, \mathbf{I}_{j_}) \equiv \sum_{l=1}^{n}(\mathbf{B}_{il} \rightarrow \mathbf{I}_{jl}) \equiv \sum_{l=1}^{n}(\overline{\mathbf{B}_{il}} \,|\, \mathbf{I}_{jl}) = n$. Operations $\rightarrow$ and $|$ represent logical implication and logical OR, respectively.

For the sake of simplicity, we will use (f_i, c) to denote a classifier, because in our experiments we use only the *similarity* function.

Responses of Classifiers We say that object g is classified by (f_i, c, sim) if $sim(\mathbf{B}_{i_}, g) > \varepsilon$. To assign a class label to g, the responses of classifiers (f_i, c, sim) can be accounted with weights $w^g_{(f_i,c,sim)}$, e.g. precision, accuracy of f_i, or similarity between $\mathbf{B}_{i_}$ and g. We assume that f_i does not contribute to the final decision on a class of g (the response is 0) if g is not classified by (f_i, c, sim). Again, for the sake of simplicity, we will use $w^g_{(f_i,c)}$ instead of $w^g_{(f_i,c,sim)}$.

To compute a class label of an object, the responses of classifiers (weights) are aggregated. We discuss aggregation strateges in Section 4.2.

4 Experimental Evaluation

To evaluate factors under supervised settings, we use 11 different real-world datasets from UCI repository [8] binarized with tools from [7]. The characteristics of the datasets are shown in Table 1. In our experiments we use 10-fold cross-validation.

We compare most common BMF algorithms, namely 8M [9], GreConD [6], GreEss [5], Hyper [19], MDLGreConD [16], NaiveCol [10] and PaNDa$^+$ [15].

Table 1. Datasets and their characteristics.

dataset	size	density $\mathbf{I}$	class distribution
`anneal`	898×66	0.20	0.76/0.04/0.11/0.07/0.01
`breast`	699×14	0.64	0.34/0.66
`hepatitis`	155×50	0.36	0.79/0.21
`horse colic`	368×81	0.21	0.63/0.37
`iris`	150×16	0.25	0.33/0.33/0.33
`led7`	3200×14	0.50	0.11/0.09/0.10 ($\times$8 classes)
`mushroom`	8124×88	0.25	0.52/0.48
`nursery`	1000×27	0.30	0.32/0.34/0.34
`page block`	5473×39	0.26	0.90/0.02/0.01/0.05/0.02
`pima`	768×36	0.22	0.650/0.35
`wine`	178×65	0.20	0.33/0.40/0.27

4.1 Factor as Classification Rule

In this section we examine factors as single classifiers. We study (i) the connection between factor ranks given by unsupervised and supervised quality measures, and which factors are best ones w.r.t. the supervised quality measures, (ii) how well the factors summarize classes.

Connection between Supervised and Unsupervised Quality Measures The mentioned BMF algorithms are based on a greedy strategy. The generated factors are ordered w.r.t. their importance. The importance of factors is estimated by a particular objective of an algorithm. Put it differently, the factors generated first might best explain data. Since some factor sets are very small, we cannot use correlation analysis to examine the dependence between the importance of factors (unsupervised quality measure) and their precision (supervised quality measure). To assess the connection between these measures we count how many factors we need to compute to get the best k factors w.r.t. precision. The less the number of factors we need to compute, the stronger connection between unsupervised and supervised quality measures.

The average number of factors is given on Figure 1. We note that the PaNDa$^+$ factor sets are small, but it does not mean that most important factors provide best precision. These small values are caused by the small sizes of the PaNDa$^+$-generated factor sets. The extremely small size of factor sets produced by PaNDa$^+$ affects also the factor quality in the unsupervised settings [4, 5].

Figure 1 shows that the lowest values correspond to the MDLGreConD factors. It means that we need to compute only few factors to get the most precise classifiers. The most important factors w.r.t. the MDLGreConD objective have relatively higher precision than the most important factors generated by other BMF algorithms.

In the next section we discuss the ability of factors to explain classes rather than data as a whole, i.e. their ability to distinguish a single class from others.

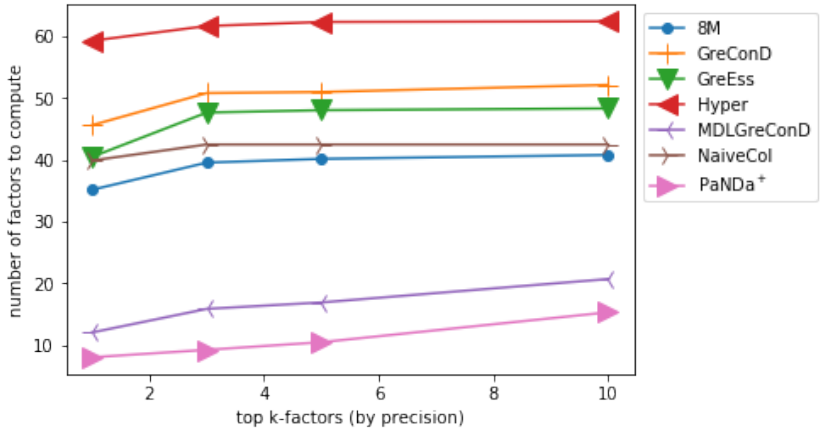

Fig. 1. The no. of factors required to be computed in order to get k best factors w.r.t. precision.

Summary of Classes For every factor-classifier (f_i, c) that corresponds to row $\mathbf{A}_{_i}$ and column $\mathbf{B}_{i_}$ we compute precision, recall and accuracy as follows:

$$prec(f_i, c) = \frac{tp}{tp + fp},\ recall(f_i, c) = \frac{tp}{tp + fn},\ accuracy(f_i, c) = \frac{tp + tn}{m},$$

where $tp = |\{\mathbf{A}_{ji} \mid \mathbf{A}_{ji} = 1, class(\mathbf{I}_{j_}) = c, j = 1, \ldots, m\}|$ is the true positive rate, $fp = |\{\mathbf{A}_{ji} \mid \mathbf{A}_{ji} = 1, class(\mathbf{I}_{j_}) \neq c, j = 1, \ldots, m\}|$ is the false positive rate, $fn = |\{\mathbf{A}_{ji} \mid \mathbf{A}_{ji} = 0, class(\mathbf{I}_{j_}) = c, j = 1, \ldots, m\}|$ is the false negative rate.

Precision and accuracy characterize how well f_i describes class c. The factors with high values of these measures summarize better the given class c. The only difference between accuracy and precision is the following one: precision is the "local" class specificity (it shows how well objects from c are distinguished among the classified objects), while accuracy is the "global" class specificity (it shows how well objects from c are distinguished among all objects). Precision and accuracy give preference to classifiers with low values of fp and $fp + fn$, respectively.

The results of the experiments given in Table 2 show that the highest average precision is achieved for factors computed by PaNDa$^+$ (0.78, on average), the MDLGreConD factors also have quite high values of precision (0.74, on average). The MDLGreConD factors have the most stable quality measures (precision on test sets is smaller by 0.07 than on training sets).

More than that, Table 2 provides precision of factor-classifiers on training and test data. Precision on training data for all algorithms is quite similar (the best algorithm is PaNDa), while MDLGreConD demonstrates the best precision on test sets. It should be noticed that MDLGreConD has the smallest difference in precision for training and test data. That might indicate its ability to generalize

Table 2. The average values of precision on training/test sets. Best values are highlighted in bold.

	SM	GreCond	GreEss	Hyper	MDLGreCond	NaiveCol	PaNDa$^+$
anneal	0.86/0.67	0.85/0.66	0.86/0.64	0.84/0.62	0.85/0.75	0.84/0.63	**0.87**/**0.87**
breast	0.88/0.73	0.88/0.85	0.84/0.84	**0.93**/0.64	0.87/0.87	0.80/0.80	0.85/0.81
hepatitis	0.80/0.64	0.81/0.61	0.81/0.60	0.81/0.68	**0.83**/0.75	0.79/0.59	**0.83**/0.55
horse colic	0.70/0.48	0.69/0.60	0.69/0.60	0.72/0.61	0.70/0.63	0.69/0.59	**0.80**/0.56
iris	0.80/0.75	0.80/0.61	0.80/0.61	0.79/0.67	0.92/0.86	0.79/0.67	**0.96**/0.53
led7	0.40/0.44	0.33/0.32	0.33/0.32	**0.50**/0.19	0.37/0.36	0.23/0.22	0.43/0.42
mushroom	0.82/0.76	0.82/0.79	0.83/0.79	0.85/0.70	**0.87**/0.84	0.78/0.75	0.81/0.00
nursery	0.45/0.44	0.45/0.44	0.45/0.44	0.45/0.44	0.42/0.41	0.45/0.44	**0.58**/0.53
page blocks	0.82/0.35	0.82/0.46	0.84/0.43	0.78/0.33	0.80/0.51	**0.83**/0.51	0.80/0.74
pima	0.70/0.43	0.68/0.49	0.68/0.48	0.69/0.44	0.68/0.61	0.67/0.45	**0.77**/0.73
wine	0.66/0.40	0.69/0.57	0.68/0.56	0.67/0.49	0.84/0.77	0.64/0.50	**0.88**/0.66
average	0.72/0.53	0.71/0.58	0.71/0.57	0.73/0.53	0.74/**0.67**	0.68/0.56	**0.78**/0.65

well (i.e. it is less likely to overfit). Almost the same quality of factors, but in the unsupervised settings, was described in [4].

4.2 Factors as Ensemble of Classifiers

The modern state-of-the-art classifiers, e.g. Random Forests, Multilayer Networks, Nearest Neighbour classifiers, are comprised of a set single classifiers, i.e. the single classifiers make ensembles. In this section we examine a set of factor-classifiers as an ensemble and evaluate its accuracy.

It should be noticed that some factor sets are incomplete, in other words, they do not contain factors for several classes. It is caused by unbalanced training sets, where some classes contain only few objects. Here we examine the datasets where there are enough factors for every class, namely `iris`, `mushroom`, `pima` and `wine` datasets. We study rule-based ensembles.

As it was mentioned in Section 3.1 the responses of classifiers can be taken into account in several ways. We focus on two strategies, where the responses of all voted classifiers or the best one are considered, we call them "all-votes" and "best-vote", respectively. For a rule-based ensemble of factor-classifiers $\mathcal{C} = \{(f_i, c) \mid j = 1, \dots, k\}$, where k is the number of factors, the class label is assigned to object g as follows:

$$\textit{all-votes-class}(g, \mathcal{C}) = \arg\max_{c \in \mathcal{Y}} \sum_{\substack{(f_i,c) \in \mathcal{C} \\ \mathbf{B}_{i_} \cdot g = \mathbf{B}_{i_}}} w^g_{(f_i,c)},$$

$$\textit{best-vote-class}(g, \mathcal{C}) = \arg\max_{c \in \mathcal{Y}} \max_{\substack{(f_i,c) \in \mathcal{C} \\ \mathbf{B}_{i_} \cdot g = \mathbf{B}_{i_}}} w^g_{(f_i,c)}.$$

The results of the experiments, given in Table 3, show that the most accurate ensembles are those that are based on the precision-weighed votes. According to

Table 3. The average accuracy of classifier ensembles computed on `iris`, `mushroom`, `pima` and `wine` datasets. Best values are highlighted in bold.

	all-votes		best-vote	
	precision	accuracy	precision	accuracy
`iris`	0.84/0.82	0.84/0.82	0.84/0.82	0.84/0.82
`mushroom`	0.93/0.93	0.89/0.89	0.99/0.99	0.88/0.88
`pima`	0.66/0.66	0.67/0.66	0.72/0.70	0.73/0.73
`wine`	0.77/0.75	0.76/0.75	0.79/0.75	0.76/0.75
average	**0.80/0.79**	0.79/0.78	**0.83/0.81**	0.80/0.79

the examined datasets, the "best-vote" scheme (where the response of the best classifier is considered) provides best results.

5 Conclusion

In this paper we examine the factors computed on unlabeled data in the supervised settings. We provided an experimental justification that in case of factors the data explanation problem is closely related to the class explanation problem, i.e. a factor is able to explain specificity of a particular (sub)class. Based on the results of the supervised factor evaluation we propose several models of factor-based ensembles of classifiers. We show that factor-based classifiers can achieve accuracy comparable to the state-of-the-art ensembles of classifiers.

An important direction of further work is to study factors computed in supervised settings for each class separately rather than for the whole dataset. Incorporating precision or accuracy to a BMF objective might improve accuracy of the model as well as provide a deeper insight on a class structure.

Acknowledgment

The work of Tatiana Makhalova was supported by the Russian Science Foundation under grant 17- 11-01294 and performed at National Research University Higher School of Economics, Moscow, Russia.

References

1. Belohlavek, R., Baets, B.D., Outrata, J., Vychodil, V.: Inducing decision trees via concept lattices. Int. J. General Systems **38**(4), 455–467 (2009)
2. Belohlavek, R., Grissa, D., Guillaume, S., Nguifo, E.M., Outrata, J.: Boolean factors as a means of clustering of interestingness measures of association rules. Ann. Math. Artif. Intell. **70**(1-2), 151–184 (2014)
3. Belohlavek, R., Outrata, J., Trnecka, M.: Impact of Boolean factorization as preprocessing methods for classification of boolean data. Ann. Math. Artif. Intell. **72**(1-2), 3–22 (2014)
4. Belohlavek, R., Outrata, J., Trnecka, M.: Toward quality assessment of Boolean matrix factorizations. Inf. Sci. **459**, 71–85 (2018)

5. Belohlavek, R., Trnecka, M.: From-below approximations in Boolean matrix factorization: Geometry and new algorithm. J. Comput. Syst. Sci. **81**(8), 1678–1697 (2015)
6. Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. J. Comput. Syst. Sci. **76**(1), 3–20 (2010)
7. Coenen, F.: The LUCS-KDD discretised/normalised ARM and CARM data library (2003), http://www.csc.liv.ac.uk/ frans/KDD/Software/LUCS_KDD_DN
8. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
9. Dixon, W.: BMDP statistical software manual to accompany the 7.0 software release, vols 1–3. (1992)
10. Ene, A., Horne, W.G., Milosavljevic, N., Rao, P., Schreiber, R., Tarjan, R.E.: Fast exact and heuristic methods for role minimization problems. In: Ray, I., Li, N. (eds.) 13th ACM Symposium on Access Control Models and Technologies, SACMAT 2008, Estes Park, CO, USA, June 11-13, 2008, Proceedings. pp. 1–10. ACM (2008)
11. Ganter, B., Wille, R.: Formal Concept Analysis Mathematical Foundations. Springer-Verlag, Berlin, Heidelberg (1999)
12. Ganter, B., Kuznetsov, S.O.: Hypotheses and version spaces. In: Conceptual Structures for Knowledge Creation and Communication, 11th International Conference on Conceptual Structures, ICCS 2003 Dresden, Germany, July 21-25, 2003 Proceedings. pp. 83–95 (2003)
13. Kueti, L.T., Tsopzé, N., Mbiethieu, C., Nguifo, E.M., Fotso, L.P.: Using Boolean factors for the construction of an artificial neural networks. Int. J. General Systems **47**(8), 849–868 (2018)
14. Kuznetsov, S.O.: Machine learning and formal concept analysis. In: Eklund, P.W. (ed.) Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings. Lecture Notes in Computer Science, vol. 2961, pp. 287–312. Springer (2004)
15. Lucchese, C., Orlando, S., Perego, R.: A unifying framework for mining approximate top-k binary patterns. IEEE Trans. Knowl. Data Eng. **26**(12), 2900–2913 (2014)
16. Makhalova, T., Trnecka, M.: From-below boolean matrix factorization algorithm based on mdl. arXiv preprint arXiv:1901.09567 (2019)
17. Makhalova, T.P., Kuznetsov, S.O., Napoli, A.: A first study on what MDL can do for FCA. In: Ignatov, D.I., Nourine, L. (eds.) Proceedings of the Fourteenth International Conference on Concept Lattices and Their Applications. CEUR Workshop Proceedings, vol. 2123, pp. 25–36 (2018)
18. Outrata, J.: Preprocessing input data for machine learning by FCA. In: Kryszkiewicz, M., Obiedkov, S.A. (eds.) Proceedings of the 7th International Conference on Concept Lattices and Their Applications, Sevilla, Spain, October 19-21, 2010. CEUR Workshop Proceedings, vol. 672, pp. 187–198. CEUR-WS.org (2010)
19. Xiang, Y., Jin, R., Fuhry, D., Dragan, F.F.: Summarizing transactional databases with overlapped hyperrectangles. Data Min. Knowl. Discov. **23**(2), 215–251 (2011)