Sperner System Descriptions in ARM

Levon Aslanyan
Institute for Informatics and Automation Problems
Yerevan, Armenia
e-mail: lasl@sci.am

Hasmik Sahakyan
Institute for Informatics and Automation Problems
Yerevan, Armenia
e-mail: hsahakyan@sci.am

***Abstract*—Data structures used in associated rule mining algorithms are presented in terms of Sperner systems. This helps to minimize the necessary algorithmic resource, also making transparent the data sets and their interrelations.**

***Keywords*—Computer science, data mining, monotone Boolean function.**

I. Introduction

Let f be an arbitrary monotone Boolean function. We notice that $Z(f) \cup O(f)$, the union of all upper zeros and lower ones of f is the only deadlock resolving set $G(f)$ of function f. All necessary terms and definitions can be found in [2, 4,10-14]. Thus, in order to determine the values of function f at all points of E^n [4-5,8,11,20], knowing its monotonicity, it is sufficient to determine them only at the points of the deadlock resolving set $G(f)$ [12], or in some interpretable area of its extension.

Let us transfer these well-known concepts of Boolean domain to the domain of partially defined Boolean functions. *Partially defined Boolean function* $\check{f}$ is determined on $h + l$ vertices of the n-dimensional unit cube E^n, $h + l < 2^n$. $\check{f}$ is a part of some function f but the complete function f is unknown, and such functions can be different. h vertices correspond to a set $H \subseteq \mathcal{N}_f$ of vertices that accept value "one", and l vertices of the set $L \subseteq \mathcal{N}_{\bar{f}}$ accept "zero" value (so they are out of the set $\mathcal{N}_f$). The reminder part $E^n - L - H$ of vertices is the area of indeterminedness of $\check{f}$, which means that the values of $\check{f}$ are not determined or, equivalently, that they are defined, but we do not know these values. When $\check{f}$ belongs to a specific class of Boolean functions, then the sets H and L cannot be arbitrary. E.g., for monotone f or $\check{f}$, H will be upward monotone and L will be downward monotone. The two sets $H = \mathcal{N}_{\check{f}}$ and $L = \mathcal{N}_{\bar{\check{f}}}$ might be given with the help of two Sperner systems: $Z(\check{f})$ and $O(\check{f})$. $Z(\check{f})$ is the set of all upper zeros of $\check{f}$. Then L is the set of all vertices of E^n majored by at least one of the vertices of $Z(\check{f})$. Similarly, $O(\check{f})$ is the set of all lower ones of $\check{f}$, and the set H must include all vertices of E^n that are majoring at least one vertex of $O(\check{f})$. And the union $G(\check{f}) = Z(\check{f}) \cup O(\check{f})$ is the only deadlock-resolving set of a partially defined monotone function $\check{f}$ (in its part $H \cup L$ in initial definition).

II. Association rule mining dataset representation by Sperner systems

The partial definedness situation of a function may have different causes. In pattern recognition, the learning set is a partially defined function $\check{f}$ because of the natural limitation of information about the subject area and its samples. On the contrary, indefiniteness is a temporal situation in the oracle-based function reconstruction, being a current result of the algorithm. Consider the step k of the association rule mining (ARM) algorithm *APRIORI* [1-3]. L_{k-1} be the list of large $k-1$ itemsets (frequent sets) at the beginning of the k-th step, and $\check{f}_{k-1}$ be the partially defined function, formed at that stage. Zeros of $\check{f}_{k-1}$ are represented by the sets $L_{k-i}, i = 1,2,\ldots$ on layers. Correspondingly, we will denote attributes of $\check{f}_{k-1}$ by L_{k-1}, H_{k-1}, $Z(\check{f}_{k-1})$ and $O(\check{f}_{k-1})$.

Lemma. For the target function $\check{f}$, at the beginning of the k-th step of the frequent sets growing procedure, the following conditions hold: $L_{k-1} \supseteq E^n_{k-1} \cap Z(\check{f})$ and $E^n_{k-1} - L_{k-1} \subseteq H$, where E^n_{k-1} denotes the $(k-1)$-th layer of E^n. (1)

We left the proof as an exercise. The idea behind the Lemma says that L_{k-1} is not a final but a temporary knowledge. *APRIORI* uses the following principle of frequent sets candidate generation. In searching candidates, having the set L_{k-1} at the beginning of the step k, it considers an arbitrary element $\alpha \in E^n_k$. All neighbours of α from E^n_{k-1} are obtained from α by zeroing one of its unit coordinates. Let us do that for the last two unit coordinates. We obtain 2 vertices with common $k-2$ prefix. This is the vertex pair used in candidate α generation. This is a weak check for candidate α. The full check requires two check parts: check that *all* $k-1$ layer neighbours of α belong to L_{k-1}, and check of the required support level at vertex α. The second check is

stronger but the first check is necessary to form the support computation samples set. These samples area must be at least L_k, which is unknown at that stage, but the smaller area is preferable due to forthcoming costly computations of support values.

Denote by $Z(\check{f}_k)$ and $O(\check{f}_k)$, correspondingly, the sets of upper zeroes and lower ones of function $\check{f}$ at the step k of *APRIORI*. When *APRIORI* finishes its work at some step k_0, then it must be $Z(\check{f}_{k_0}) = Z(f)$ and $O(\check{f}_{k_0}) = O(f)$. Sperner systems $Z(\check{f}_k)$ and $O(\check{f}_k)$ define nested monotone Boolean functions that converge to the frequent itemset function. We also will consider complementary Sperner systems $\acute{Z}(f)$ and $\acute{O}(f)$ to $Z(f)$ and $O(f)$. In general, $\acute{Z}(f) = O(f)$ and $\acute{O}(f) = Z(f)$. But for partially defined functions these sets are different.

The trade-off between the number of candidates and the number of elements of computation of the support value is complexity critical parameter in ARM. Basically, candidate generation is a process in RAM, so that these sets can be analyzed and minimized in their complete form. When candidate set is determined, then an additional I-O of transaction database is required for computing support values. In this context, number of candidates is an important but not critical value. It is clear that this trade-off problem cannot have an unambiguous solution and because of this it has been the subject of a number of studies [6-7,15-19,21]. The partitioning algorithm [19], for example, allows the construction of an extended set of candidates by passing this set to the one last step of reading database, with counting supports. Another approach - clustering [18,21] also builds an extended set of candidates, but this is based on minimizing the number of iteration steps when constructing candidates, trying to analyze and better use the information from the set L_{k-1}.

Consider an arbitrary subset $\Delta \subseteq E_p^n$. We call an interior q-point of elements of the set Δ all-those-vertices $a \in E_q^n$, for which all vertices of the layer E_p^n comparable to them, belong to the set Δ. Similarly, $a \in E_q^n$ is a shadow q-point for Δ, if at least one vertex of the layer E_p^n comparable to a belongs to the set Δ. Insert a notation to these concepts. For an arbitrary $\Delta \subseteq E_p^n$ we denote by $I^l(\Delta)$ the set of all interior points of the layer $p + l$ of E^n, determined by the set Δ. Similarly, $S^l(A)$ is the set of all shadow vertices of the layer $p + l$ of E^n that are determined by the set Δ. l is an integer, and when $l = 0$, then $I^l(\Delta) = S^l(\Delta) = \Delta$. For given p the value domain of l is determined by relation $-p \le l \le n - p$.
According to these definitions, the k-th step of *APRIORI* can be presented by the following picture (Figure1)

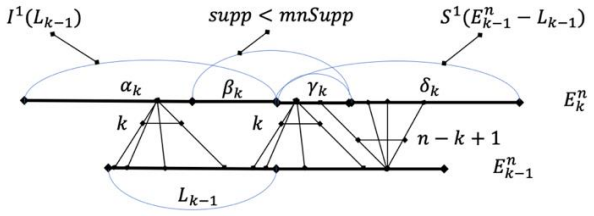

Figure1

Interval composed of all vertices of type α_k plus all vertices of type β_k corresponds to the real set of candidate itemsets on layer k: all their $k-1$ subsets are large subsets, they belong to L_{k-1}. The part of vertices β_k of this interval consists of low support elements so that they will not be included into the L_k. Interval of vertices γ_k consists of those vertices, that reduced in 2 last elements become an item of L_{k-1}. They are members of the *APRIORI* candidates set, but the collection of their $k-1$ subsets have intersection with $E_{n-1} - L_{k-1}$, so that their support can't be satisfactory and these points can't be included in L_k as well. The reminder interval δ_k is not part of the set of candidates.

Analyze the diagram in terms of lower ones and upper zeros. Consider $I^1(L_{k-1})$ and its parts α_k and β_k. $\alpha_k + \beta_k + \gamma_k = C_k$, and $\alpha_k = L_k$. $S^{-1}(\alpha_k)$ is the set of all large $k-1$ itemsets that are covered by the new large k itemsets. While the elements of α now have to be included into the set $Z(\check{f}_k)$, the vertices of $S^{-1}(\alpha_k)$, being upper zero at the step $k-1$ have to be eliminated because they are covered by elements of α_k.

Theorem 1. In *APRIORI* algorithm the set $Z(\check{f}_k)$ can be obtained by the recurrent formula
$$Z(\check{f}_k) = Z(\check{f}_{k-1}) - S^{-1}(\alpha_k) + \alpha_k. \quad (2)$$

In term $(\alpha_i - S^{-1}(\alpha_i)$ the two compounds belong to different layers. This term might be applied to the set $Z(\check{f}_{i-1})$, where the part on the layer i is reduced by $S^{-1}(\alpha_i)$, and the new layer k is involved into the process adding the set α_i. To compute this formula it is necessary to determine the sets α_i, and the required technique is demonstrated in Figure 1 above. Computation involves several transformations into the shadows and interior vertices, and computation of supports at the same time. $O(\check{f}_k)$ is more straightforward:

Theorem 2. In *APRIORI* algorithm the set $O(\check{f}_k)$ can be obtained by the recurrent formula $O(\check{f}_k) = O(\check{f}_{k-1}) + \beta_k$. Alternatively, $O(\check{f}_k) = (E_1^n - L_1) + \sum_{i=2}^{k} \beta_i$. (3)

III. Conclusion

Concluding the points (1), (2), (3), it should be mentioned that *APRIORI*, in a slight modification, may compute and return the sets $Z(\check{f}_k)$ and $O(\check{f}_k)$ of upper zeroes and lower ones, which in step k are wider and narrower than the corresponding final sets. Extension of information of $Z(\check{f}_k)$ and $O(\check{f}_k)$ into the final sets of frequencies is straightforward by one read of the database, passing the set out of monotone domains determined by $Z(\check{f}_k)$ and $O(\check{f}_k)$.

Acknowledgment

The authors would like to thank Committee of Science of Republic of Armenia for supporting their research.

References

[1] Agrawal R., Imielinski T., Swami A., "Mining association rules between sets of items in large databases", *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 207-216, Washington D.C., May 1993.

[2] Agrawal R., Srikant R., "Fast algorithms for mining association rules", P*roceedings of 20th International Conference Very Large Data Bases*, VLDB, Santiago, Chile, pp. 487-499, 1994.

[3] Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo I., "Fast Discovery of Association Rules, A*dvances in knowledge discovery and data mining",* 12 (1), pp. 307-328, 1996.

[4] Alekseev V., "On deciphering of some classes of monotone many valued functions", *USSR Computational Mathematics and Mathematical Physics*, vol. 16, no. 1, pp. 189-198, 1976.

[5] Aslanyan L., Khachatryan R., "Association rule mining enforced by the chain decomposition of an n-cube", *Mathematical Problems of Computer Science*, XXX, ISSN 0131-4645, 2008.

[6] L. Aslanyan, H. Sahakyan et al, "Managing Risk and Safety" (chapter 1), *Intelligent Data Processing in Global Monitoring for Environment and Security*, ITHEA, Sofia (Bulgaria) and Kiev (Ukraine), Editors: K. Markov and V. Velichko, ISBN: 978-954-16-0045-0 (printed), 410 p., 2011.

[7] Aslanyan L., Sahakyan H., "The Splitting technique in monotone recognition", *Discrete Applied Mathematics*, no. 216, pp. 502–512, 2017.

[8] Bodon F., Ronyai L., "Trie: An Alternative Data Structure for Data Mining Algorithms", *Mathematical and Computer Modelling* 38, pp. 739-751, 2003.

[9] Boros, E., Hammer P. L., Ibaraki T., Makino K., "Polynomial-time recognition of 2-monotonic positive Boolean functions given by an oracle", *SIAM Journal on Computing* 26, pp. 93–109, 1997.

[10] Engel K., Encyclopedia of Mathematics and its Applications 65, *"Sperner Theory"*, Cambridge University Press, MA, 1997.

[11] Hansel G., 1966, "Sur le nombre des foncions Booleenes monotones de n variables", *C. R. Acad. Sci.* Paris 262, serie A, pp. 1088–1090 (in French).

[12] Korobkov V., "On monotone functions of algebra of logic", *Problemy Kibernetiki*, no. 13, 1965.

[13] Korshunov A. D., "On the number of monotone Boolean functions", *Problemy Kibernetiki*, no. 38, 5–108 (in Russian), 1981.

[14] Kovalerchuk B., Triantaphyllou E., Deshpande A. S., "Interactive learning of monotone Boolean functions", *Information Sciences*, 94. pp. 87–118, 1996.

[15] Li H. F., Lee S. Y., Shan M. K., "DSM-PLW: single-pass mining of path traversal patterns over streaming web click-sequences", *Proc. of Computer Networks on Web Dynamics*, pp. 1474–1487, 2006.

[16] Makino K., Ibaraki T., "A fast and simple algorithm for identifying 2-monotonic positive Boolean functions", *Proceedings of ISAACS'95, Algorithms and Computation*, Springer-Verlag, Berlin, Germany, pp. 291–300, 1995.

[17] Makino K., Suda T., Ono H., Ibaraki T., "Data analysis by positive decision trees", *IEICE Transactions on Information and Systems* E82-D, pp. 76–88, 1999.

[18] Makino K., Uno T., "New algorithms for enumerating all maximal cliques", Algorithm Theory: SWAT 2004, Lecture Notes in Computer Science, 3111, Springer-Verlag, pp. 260–272, 2004.

[19] Savasere A., Omiecinski E., Navathe S., "An efficient algorithm for mining association rules in large databases", *In 21st VLDB Conf.*, 1995.

[20] Tonoyan G., "Chain decomposition of n dimensional unit cube and reconstruction of monotone Boolean functions", *JVM&F*, vol. 19, no. 6, 1532-1542, 1979.

[21] Zaki M. J., Parthasarathy S., Ogihara M., Li W., "New algorithms for fast discovery of association rules", *TR 651, CS* Dept., University of Rochester, 1997.