International Journal of Information Technology and Knowledge Management
July-December 2011, Volume 4, No. 2, pp. 709-711

HIERARCHICAL TILE MODEL FOR FREQUENT ITEM SET MINING

Rajeshwar[1] & Vivek Chandra[2]

In this paper we discuss Hierarchical Tile Model to find concentration of attributes. In this model we consider the data as Zero-One Matrix and record those sub matrices resulting different density from their surroundings. Two implications may be found (i) Matrices do not have attributes and tuples; (ii) Sub matrix is considered interesting if its density differs from its surroundings leads to interchangeability of zeros and ones of the matrix. In this model, sub matrices are contiguous. The ordering of rows and columns may be irrelevant.

1. PRELIMINARIES

In this paper only binary data is considered. For example, the data set can record customer item relationship in a supermarket. What is not recorded is numerical information. Any external information not used in frequent itemset mining is ignored, even knowing the fact that the information may be useful in other Data Mining tasks. Let U be the set of attributes. Each customer represents $T \subseteq U$. In this paper, we consider a data Matrix M with i rows and j columns. The value at intersection of rows and columns is denoted by M[m, n].

2. HIERARCHICAL TILES

Figure 1, indicates a sample data set with one as dot and zero as empty space. The dense area indicates the frequent itemsets. To identify such patterns, modeling of Hierarchical Tiles is required; which are defined in terms of rectangles.

2.1 Rectangle

A rectangle of M is a pair (X, Y), where $X \subseteq [i]$ and $Y \subseteq [j]$. A proper sub rectangle of (X, Y) is a rectangle (X', Y') such that $X' \subseteq X$ and $Y' \subseteq Y$ and $(X, Y) \neq (X, Y)$. The co-ordinate pair (m, n) falls within the rectangle (X, Y) if $m \in X$ and $n \in Y$.

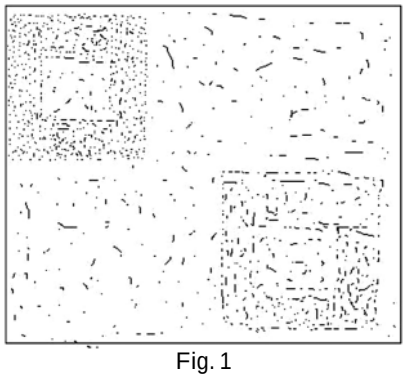

Fig. 1

2.2 Tile

A rectangle coupled with a density parameter is known as Tile. Tile may contain sub tiles and so on. The following definition of Hierarchical Tile is recursive. When the set of contained tile is empty then it is considered as occurrence of base case.

A hierarchical tile of M is a quadruple t = (X, Y, p, T) where

(a) (X, Y) is rectangle of M and also domain of t

(b) p is a number in [0,1]

(c) T is a set of $[t_1, t_2, \ldots t_k]$ of Hierarchical Tiles, whose tails are pairwise disjoint proper subsets of rectangle (X, Y).

Consider a hierarchical tile model as shown in Figure 1,

$$t_0 = (X_0, Y_0, p_0, \{t_1, t_2\})$$

where X_0, Y_0 defines the whole rectangle

$$t_1 = (X_1, Y_1, p_1, t_3)$$

$$t_2 = (X_2, Y_2, p_2, \varnothing)$$

p_j can be filled only after the proper definition of semantics of tile model.

To facilitate we define the concept of exclusive domain. Exclusive Domain of a tile t is the domain of t with the domains of its subtiles is removed. By this definition, a hierarchical tile model t has the prediction p for every co-ordinate pair (m, n) of the Data Matrix. A good model should predict the value t[m, n] which is close to the actual data values M(m, n). The values in the data matrix are considered to be independent with probability of the event M[m, n] = 1 given by the prediction t[m, n].

3. ALGORITHMS

Our first algorithm is to discover hierarchical tiles. In this stage of our research, the algorithm only discover single tile with high contribution to the log-likelihood of the

[1]Assistant Professor, Shri Parshuram College, Kuruskhetra.
[2]Head, Department of IT, MP East Zone dis.com, H14, Jabalpur (M.P.)
E-mail: [1]rzaildaar@yahoo.com,

model. According to Algorithm 1, The log likelihood of an hierarchical tile that has no subtiles is computed from the data in constant time after a linear time preprocessing step. The preprocessing consist of finding the commutative sum

$$M^*[m, n] = \sum_{k=0}^{m}\sum_{l=0}^{n} M(k, l)$$

For all $0 \le m \le i$ and $0 \le n \le j$

Algorithm 1

Input: A data matrix M of dimensions i × j

Output: A commutative sum matrix M*

For k = 0 to i

M*[k, 0] = 0

For l = 0 to j

M*[0, l] = 0

For k = 1 to i

For l = 1 to j

M*[k, l] = M[k, l] + M*[k – 1, l] + M*[k, l – 1] – M*[k – 1, l – 1]

Once these sums are known, the number N of ones inside the tile {a,….b} × {c, …. , d} is simply

$$N1 = \sum_{k=a}^{b}\sum_{l=c}^{d} M(k, l) = M^*[b, d] - M^*[b, c-1] - M^*[a-1, d] = + M^*[a-1, c-1]$$

The log likelihood of the tile is given by

log L(X, Y) = N.H(N1/N)

where N = |X|.|Y|, H(p) = p log p + (1 – p)log (1 – p)

The log likelihood of the surrounding tiles is taken care into the account by the algorithm.

The number of dimensions in a data set of the size i × j is $\theta(i^2 j^2)$. Hence an exhaustive search is feasible in polynomial time.

Algorithm 2

Input : A data matrix M of dimensions i × j

Output : A hierarchical rectangle (X, Y)

M* = commutative sum (M)

Best = – ∞

For X_1 = 1 to i

For $X_2 = X_1$ to i

For Y_1 = 1 to j

For $Y_2 = Y_1$ to j

L = loglikelihood (X_1, X_2, Y_1, Y_2, M*)

X = {X_1, X_1 + 1,……, X_2}

Y = {Y_1, Y_1 + 1,……., Y_2}

With large sized data set, quardratic time may be too much. The next algorithm describes a local search which in most of the cases search good hierarchical tiles.

Algorithm 3

Input: A data matrix M of dimensions i × j

Output: A hierarchical rectangle (X, Y)

(X_1, X_2), random number such that $X_1 \le X_2$

(Y_1, Y_2), random number such that $Y_1 \le Y_2$

best = – ∞

Repeat

L = loglikelihood (X_1, X_2, Y_1, Y_2, M*)

If L ≤ best

Return B

best = L

B = ({X_1,……. X_2},{Y_1,…….Y_2})

C = ({$X_1 - 1, X_2, Y_1, Y_2$}, {$X_1 + 1, X_2, Y_1, Y_2$},{$X_1, X_2 - 1, Y_1, Y_2$},{$X_1, X_2 + 1, Y_1, Y_2$},

{$X_1, X_2, Y_1 - 1, Y_2$}, {$X_1, X_2, Y_1 + 1, Y_2$},{$X_1, X_2, Y_1, Y_2 - 1$}, {$X_1, X_2, Y_1, Y_2 + 1$})

(X1, X2, Y1, Y2) = max likelihood (R, M*)

$R \in C$

B is the corresponding tile.

This algorithm moves all edges of the rectangle by one step and selects the one with best likelihood.

4. CONCLUSION

Local search may not stuck the local optima. It is also a fact that when data matrix is sampled from a model and the size of the data approaches infinity, the method does have a high probability of ending up at a global optima. To achieve the same log likelihood of both the tiles and rest of the matrix is taken into account.

REFERENCES

[1] Foto Afrati, Aristides Gionis, and Heikki Mannila., "Approximating a Collection of Frequent Sets", In Ronny Kohavi, Johannes Gehrke, William DuMouchel, and Joydeep Ghosh, editors, Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), page 12–19. ACM Press, 2004.

[2] Ella Bingham, Heikki Mannila, and Jouni K. Seppänen., "Topics in 0–1 Data", In Proceedings of the ighth ACM SIGKDD InternationalConference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada, pages 450–455. ACM, 2002.

[3] Toon Calders. "Deducing Bounds on the Support of Frequent Itemsets", In Database Support for Data Mining Applications: Discovering Knowledge with Inductive Queries, pages 214–233. Springer-Verlag, 2004.

[4] Linchun Gao and András Prékopa, "Lower and Upper Bounds for the Probability that at Least r and Exactly r Out of n Events Occur", Mathematical Inequalities & . . . . , 5(2):315–333, 2002.

[5] Inderjit S. Dhillon., "Co-clustering Documents and Words Using Bipartite Spectral Graph Partitioning", In Knowledge Discoveryand Data Mining, pages 269–274, 2001.