Mining Multi-connection Bridging Rules Using Hidden Markov Model

Weiqi Zhang[1], Gang Li[2], Qingfeng Chen[3], and Yuan Jiang[4]

[1] School of Computer Science and Engineering, Southeast University, Nanjing, 211189, China
[2] Deakin University, Austrilia,
ligang@ieee.org
[3] Guangxi University, Guangxi, China
[4] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China

Abstract. A *Multi-connection Bridging Rule (MCBR)* is a sequence of *connections* and each *connection* is a rule whose antecedent and consequent belong to different conceptual classes. MCBR can capture inter-class information among different conceptual classes, at the same time the relationship of items within an identical class. It can cater for various application scenarios such as group-based criminal detecting and correlation detecting in bioinformatics. In this paper, we proposed an approach to discovering interesting MCBRs in database based on *Hidden Markov Model (HMM)*. By presenting data in HMMs, we can estimate the interestingness of a *connection* simultaneously considering the topological information and association relation. The *semi-Markov Random Walk* technique and *all-confidence* are adopted to evaluate the interestingness of a MCBR. Considering that a MCBR is a sequence of *connections*, the problem of finding the most interesting MCBR is then transformed into the task of finding the most likely sequence of pairs of two hidden states in HMMs. A *viterbi* based algorithm, *V-Bridge*, is proposed to mine MCBRs in database. We conduct experiments on synthetic data to demonstrate the effectiveness of our approach.

Keywords: association rule, bridging rule, Hidden Markov Model

1 Introduction

Most existing association rule mining algorithms focus on discovering association rules among items that belong to one conceptual class, while items are usually classified into several classes in real-world applications. If simply ignoring the characteristics of different classes, association rules that contain interesting inter-class information will remain undetected. In many scenarios, especially in molecular analysis in bioinformatics [1], credit card fraud detection [2] and criminal investigative analysis [3], a new kind of association rules, whose antecedent and consequent belong to different conceptual classes, is in need of attention. In 2006, Zhang et al. proposed the concept of *bridging rule* [3]: a *bridging rule*, with its *antecedent* and *consequent* from different conceptual classes, indicates the correlation of items among these classes. Zhang et al. further proposed algorithms [3,4] to discover bridging rules in database.

However, in existing work, the bridging rule contains only one connection between items from different classes, though more connections may exist. For example, in Fig. 1, `carbon steel` belongs to the class of `black metal`, while `nickel` and `chrome` belong to the class of `rare metal`. Each solid line between two classes indicates one single connection, thus, there are two connections: "`carbon steel - chrome`" and "`carbon steel - nickel`". If we consider only one but ignore the other, none of these connections has significant practical value. But when simultaneously taking both connections into consideration, the synthesis of three elements can bring us `stainless steel`, which has been widely used in fields such as metallurgical industry. Another application scenario arises in bioinformatics. In the past decade, researchers have been focused on the discovery of miRNA and the identification of their target mRNAs. According to [5], multiple miRNAs may regulate one mRNA, and one miRNA could have several target mRNAs. Regarding the correspondence between an miRNA and its target mRNA as a connection, we can have connections such as "hsa_miR_107 - `ABHD12`", "hsa_miR_107 - `MRPS16`"and so on. The miRNAs and mRNAs belong to different conceptual classes, and these connections reveal new insights into biological procedures [6].

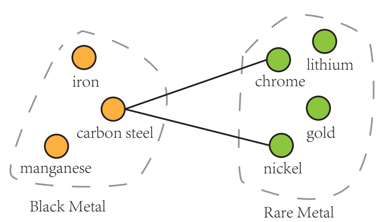


Fig. 1. A Rule with Two Connections

In this paper, we generalize the single connection *bridging rule* into a more general multi-connection *bridging rule*: one such rule could represent more than one connection and reveal the correlation property of these connections. And a multi-connection bridging rule discovering algorithm is proposed based on the *Hidden Markov Model* (HMM) approach. There are two main challenges in our work. The first is *how to represent the dataset in HMMs*. To address it, we construct a graph for each class, and the corresponding HMM can be built based on the graph. The second is *how to evaluate the interestingness between items from different classes*. This can be addressed by executing the *semi-Markov random walk process* [7] on HMMs. The contribution of this paper can be summarized as follows:

– Generalize the concept of *bridging rule* into the *multi-connection bridging rule*, which captures multiple relationships between different classes.
– Represent dataset in HMMs. This approach highlights the topology property of dataset and reduces computing complexity of the problem.
– Evaluate the interestingness of a set of items between different classes by performing a semi-Markov random walk on HMMs. This takes two factors into consideration: the correlation between items, and the topology property of each class.
– A mining algorithm is proposed to discover the multi-connection bridging rules based on HMMs. The algorithm can find the top K interesting rules among the classes.

2 Preliminaries and Related Work

The concept of *bridging rule*[3] was first proposed as a derivative of conventional association rule, *bridging rule*'s *antecedents* and *consequents* belong to different classes. Formally, let $I = \{i_1, i_2, \cdots, i_N\}$ be a set of N items, $Attr = \{a_1, a_2, \cdots\}$ is the set of attributes for each item, and D is a set of variable length transactions over I. When all items in I can be partitioned into disjoint classes $C_1, C_2, \cdots, C_m$, we have the following definition [3]:

Definition 1 (bridging rule). *An association rule $A \rightarrow B$ is a* bridging rule*, if and only if items in A are from classes $C_{A_1}, C_{A_2}, \cdots, C_{A_s}$, and items in B are from $C_{B_1}, C_{B_2}, \cdots, C_{B_t}$, where $\{C_{A_1}, C_{A_2}, \cdots, C_{A_s}\} \cap \{C_{B_1}, C_{B_2}, \cdots, C_{B_t}\} = \phi$.*

Three algorithms exist to discover *bridging rule*, they are *agglomeration*-based algorithm, *weighting*-based algorithm and *rough set*-based algorithm [3,4]. Among them the *agglomeration* based algorithm adopts the clustering technique "*CHAMELEON*" to discover the bridging rule. The *weighting* based algorithm uses *support* and *chi-squared value*, and enumeration tree to measure and prune the frequent itemsets. The *rough set*-based bridging rule discovery algorithm uses rough sets to discover bridging sets, then uses the *support-confidence* frame to find out the candidate bridging rules. Zhang et al. regard the bridging rules as channels between classes, thus they evaluate the rules using *information entropy* based metric [4].

There are several limitations in early works. The first limitation is that there's only one connection in the conventional bridging rule*(single connection bridging rules)*. In practice, such as the miRNAs and their targeted mRNAs mentioned in Sec 1, two different classes might have more than one connection and each one can reveal interesting inter-class information. The second limitation lies on the fact that early works neglect the topological property of items when mining the bridging rules. Nevertheless, in the real application the data always contains topological information. For example, the social network of the suspects of the credit card fraud and the topological structure of the protein−protein interaction (PPI) network.

To alleviate these two limits, in this work we generalize the *single connection bridging rule* into *multi-connection bridging rule*, and design a rule mining algorithm based on HMM.

3 Discovering Multi-Connection Bridging Rule

3.1 Multi-Connection Bridging Rule

Let $I = \{i_1, i_2, \cdots, i_N\}$ be a set of N distinct literals called *items*. Assume that items in I are partitioned into two disjoint clusters: C_1 and C_2, with $C_1 \cap C_2 = \emptyset$. A connection and a multi-connection bridging rule (MCBR) can be defined as:

Definition 2 (connection). *For two classes C_i and C_j, let x and y be two items: $x \in C_i$ and $y \in C_j$. A* connection *con is an implication of the form $x \rightarrow y$, where x is the* antecedent *and y is the* consequent *of the connection con.*

Definition 3 (multi-connection bridging rule(MCBR)). *For two classes C_i and C_j, a multi-connection bridging rule Bridge is a sequence of connections: $Bridge = \{con_1, con_2, \cdots, con_L\}$, which satisfies: for each connection $con_k \in Bridge$, the* antecedent *of con_k belongs to C_i and the* consequent *of con_k belongs to C_j. L is the size of multi-connection bridging rule. Notice that when $L = 1$, the multi-connection bridging rule degenerates into the original single connection bridging rule.*

Two kinds of databases can provide us information on items and classes. The first is relational database, which provides the attribute values for each item. A sample relational database is in Table 1. Note a_{kj} as value of item i_k on attribute a_j. The second is the transactional database: D is a set of variable length transactions over I. Each record $r \in D$ contains a set of items that simultaneously appear in a certain event. A sample transaction database is in Table 2(a).

A MCBR spans across different item classes, as well contains more connections than single connection bridging rule. These characteristics make discovering MCBR a challenging issue.

3.2 HMM-Based Mining Method

Data Representation in HMM Let $G_1 = (V, E_1)$ be a graph representing the class C_1, where $V = \{v_1, v_2, \cdots, v_{N_1}\}$ is a set of N_1 nodes(items), corresponding to items in the C_1, and $E_1 = \{d_{kj}\}$ is a set of M_1 edges, with d_{kj} indicating the interaction of v_k and v_j. The interaction between v_k and v_j exists only when they satisfying $sim(v_k, v_j) > \theta$, where θ is a threshold to filter uninteresting interactions, and function $sim(\cdot)$ measures the similarity between two items in the same class. Many measurements exist for the similarity between two items in different application scenarios[8,9,10]. In this paper, we adopt the *Euclidean distance* based similarity [11]:

$$sim(i_k, i_j) = \frac{1}{1 + \sqrt{\sum_{p=1}^{M} (a_{kp} - a_{jp})^2}} \tag{1}$$

Similarly, let $G_2 = (U, E_2)$ be the graph representing the class C_2 with N_2 nodes and M_2 edges, and let u denote the nodes in G_2.

Now let us consider a real-world example. In *TripAdvisor*, a worldwide tourism products review website, customers could score their stayed hotels from different aspects For customers v_1 to v_7 who scored up to three stars hotels, we group them into class C_1; and for u_1 to u_6 who scored four or five stars hotels, we group them into class C_2, as shown in Fig. 2(a) and Table 1. Attributes a_1 to a_4 stand for *location, service, sanitation and cost performance* of the hotel respectively.

Table 1. Example: relational data of hotel reviews.

(a) Class 1

	a_1	a_2	a_3	a_4
v_1	1	3	2	3
v_2	1	4	2	3
v_3	2	4	2	3
v_4	3	4	2	2
v_5	2	5	3	3
v_6	2	5	4	3
v_7	3	5	3	4

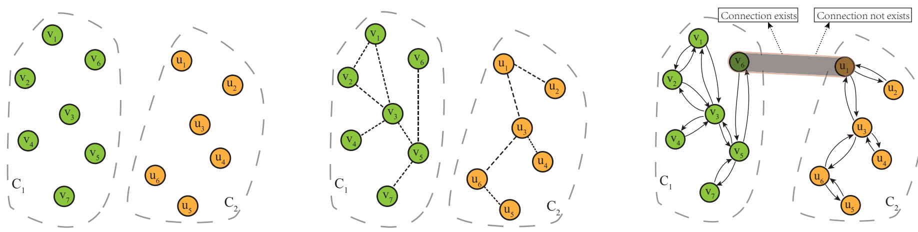
(b) Class 2

	a_1	a_2	a_3	a_4
u_1	2	3	4	3
u_2	1	3	4	4
u_3	3	4	4	3
u_4	4	5	4	3
u_5	3	4	5	5
u_6	3	4	5	4

For two customers whose reviews are similar, we regard them with similar attitude and taste, and set a connection between them. Given threshold $\theta = 0.4$, we can build two graphs corresponding to two classes, as in Fig. 2(b).

Once graphs G_1 and G_2 are ready, we construct a HMM for each of them. Then for each HMM, we design the state transition diagram based on the graph's structure. Every node in the graph corresponds to a corresponding hidden state in the HMM. The state transition from one state to another is allowed when their corresponding nodes are connected in the original graph.

Let us take an example for class C_1. Each node $v_k \in V_1$ corresponds to a hidden state in HMM. For convenience, we denote the states as v_k. For each edge $d_{kj} \in E_1$, we add an edge between v_k and v_j in the HMM. Therefore, the HMM of G_1 shares an identical structure with G_1. Fig. 2(c) shows the corresponding HMMs of G_1 and G_2, we note them as G_1' and G_2' respectively. The solid lines with arrows indicate the transitions between states. The transition probability will be calculated later.



(a) (b) (c)

Fig. 2. (a) shows classes of items; (b) shows each class's corresponding graph; (c) shows the corresponding HMMs of G_1 and G_2.

One connection con_k in a MCBR can be viewed as observation emitted by a pair of hidden states v_i and u_j in the respective HMMs. Accordingly, these two HMMs can be regarded as generative models that *jointly* emit, or produce the MCBRs. In Fig. 2(c), the dashed lines with arrow point to the observations of a pair of states $v_6 - u_1$. There are two observations "Connection exists" and "Connection not exists", the joint emit probability will be determined later.

Evaluating the Interestingness of Connections Let $N(v_i)$ be the set of neighbors of v_i in the graph. Hidden state transition probabilities for two HMMs can be determined by:

$$t_1(v_i, v_j) = \frac{sim(v_i, v_j)}{\sum_{k \in N(v_i)} sim(v_i, v_k)} \tag{2}$$

$$t_2(u_i, u_j) = \frac{sim(u_i, u_j)}{\sum_{k \in N(u_i)} sim(u_i, u_k)} \tag{3}$$

$t_1(v_i, v_j)$ means the transition probability from v_i to v_j, and $t_2(u_i, u_j)$ is the transition probability from u_i to u_j.

Once the transition probability is estimated, we then adopt the *All-Confidence* [12] and the *Semi-Markov Random Walk* [7] techniques to determine the emit probability of two HMMs.

Measured by *all-confidence* [12], an association rule is deemed interesting even though its items are not frequent enough but still with a high dependency among each other. In the real application of the MCBR, the interested items are usually infrequent, such as in the credit card fraud detection, the suspects only do very few frauds in a long period of time to conceal themselves. Thus *All-confidence* can better measure the characteristics of the new bridging rules. The *all-confidence* [12] of a set of items T is defined as:

$$all(T) = \frac{|\{d | d \in D \wedge T \subset d\}|}{MAX\,\{i | \forall t (t \in P(T) \wedge t \neq \emptyset \wedge t \neq T \wedge i = |\{d | d \in D \wedge t \subset d\}|)\}} \tag{4}$$

where $P(T)$ is the power set of T, D is the set of all transactional records. Please note that the maximum value will occur when the subset of T consists of a single item.

Referring to the example of hotels and customers again, if two or more customers have visited the same tourist attraction, we put them into a transactional record. The set of transactional records is shown in Table 2(a). An example of

Table 2. (a) shows a tourist transactional database; (b) shows *all-confidence* of itemsets for Tourist data

(a)

$TouristAttractions(Records)$	$Customers(Items)$
r_1:Hong Kong Skyline	v_3, u_2, u_3
r_2:Victoria Peak	v_5, v_6, u_6
r_3:Ocean Park	v_3, v_7, u_4, u_5
r_4:Hong Kong Disneyland	v_6, u_6
r_5:Victoria Harbour	v_1, v_3, u_3, u_5

(b)

T	$\{v_3\}$	$\{u_3\}$	$\{v_3, u_3\}$	$\{v_7, u_5\}$	$\{v_3, u_3, u_7\}$	$\{v_6, u_6, u_7\}$
$all(T)$	1	1	2/3	1/2	1/3	0

item (customer) set T and its *all-confidence* $all(T)$ is shown in Table 2(b).

Semi-Markov random walk is the process that makes state transitions based on a Markov chain. At each time point, the random walker moves to one of the current node's neighbors in both HMMs. We perform *semi-Markov random walk* on G'_1 and G'_2, according to the hidden state transition probability t_1 and t_2. Note $\pi_1(v_1)$ as the stationary probability of visiting node v_1 in the Markov random walk on G'_1 , $\pi_2(u_1)$ for u_1 in G'_2. A node in G'_1 or G'_2 with higher stationary probability, can be considered has greater importance in the conceptual class it belongs to, and also could better reveal the topological characteristics of that conceptual class. Assume the random walker spends a random amount of time with mean $\mu_\wp$ on an arbitrary pair of states $\wp$, then the time the random walker spends on a specific pair of states $\wp = \{v_i, u_j\}$ is $\pi_1(v_i)\pi_2(u_j)\mu_\wp$. If the mean time $\mu_\wp$ corresponds to $all(\wp)$, then the proportion of time that the random walker spends at $\wp = \{v_i, u_j\}$ can be computed as:

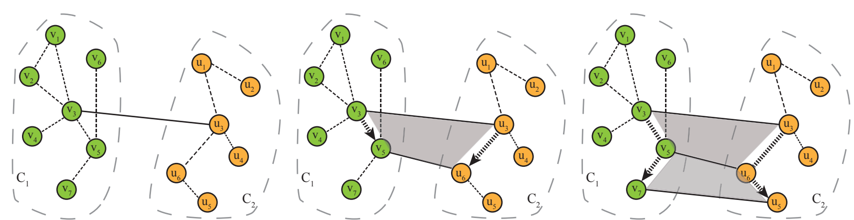
$$e(v_i, u_j) = \frac{\pi_1(v_i)\pi_2(u_j)all(\{v_i, u_j\})}{\sum_{m=1}^{N_1}\sum_{n=1}^{N_2}\pi_1(v_m)\pi_2(u_n)all(\{v_m, u_n\})} \tag{5}$$

This equation shares a similar with the equation in [13], it gives us the long-run proportion of time it spends on a specific pair of items.

We adopt the proportion of time $e(v_i, u_j)$ as the probability that the pair of hidden state (v_i, u_j) emits a connection $v_i \to u_j$. It can be regarded as the interestingness measure of the connection $v_i \rightarrow u_j$. This approach provides us an effective way to evaluate the interestingness of a connection, by combining the nodes' transactional and topological correlations.

Mining Multi-Connection Bridging Rule Considering that a MCBR is a sequence of connections, the problem of finding the most interesting MCBR is then transformed into the task of finding the most likely sequence of pairs of two hidden states whose observation are "Connection exists".

Viterbi algorithm[14] is frequently adopted to find the most likely sequence of hidden states in HMMs. We propose a algorithm *V-Bridge* based on *Viterbi* algorithm to mine MCBRs in database. Before presenting the algorithm, we demonstrate the process of mining the MCBR in Fig. 3.



(a) (b) (c)

Fig. 3. The process of discovering MCBR with size $L = 3$. Fig. 3(c) is the final state of the MCBR $Bridge = \{con_1, con_2, con_3\}$, in which con_1 is $v_3 \rightarrow u_3$, con_2 is $v_5 \rightarrow u_6$ and con_3 is $v_7 \rightarrow u_5$. The dashed line with arrow suggests the antecedent and consequence's transitions from one connection to another connection.

Let l be the size of the MCBR, v_i and u_j be the antecedent and consequent of the last connection $v_i \rightarrow u_j$ in the MCBR. Then the log-probability of the most interesting MCBR $\gamma(l, v_i, u_j)$ can be recursively computed as:

$$\gamma(l, v_i, u_j) = \max_{m,n} \left[\gamma(l-1, v_m, u_n) + \log t_1(v_m, v_i) + \log t_2(u_n, u_j) + \log e(v_i, u_j)\right] \tag{6}$$

where $\gamma(l-1, v_m, u_n) + \log t_1(v_m, v_i) + \log t_2(u_n, u_j) + \log e(v_i, u_j)$ indicates that at a certain time point, the candidate MCBR has already contained $l-1$ connections. It calculates the log probability of all these things occurred: the last candidate connection is $v_m \rightarrow u_n$, v_m transfers to v_i, u_n transfers to u_j and the next candidate connection is $v_i \rightarrow u_j$. The max function helps us get the most possible choice at each step.

We repeat the above iterations from $l = 1$ till $l = L$, then get the most interesting MCBR with size L. The pseudo code of this ***V****iterbi based Multi-connection* ***Bridging*** *Rule Discover* (V-Bridge) algorithm is shown in Alg.1.

In the algorithm, $\Gamma[l, i, j]$ stores the probability that $con_l = (i, j)$ in $Bridge = \{con_1, con_2, \cdots, con_L\}$. $R[l, i, j]$ stores the previous pair of sequence number

Algorithm 1 *Viterbi based MCBR Discover(V-Bridge)* Algorithm

Require: T_1 is HMM of G_1' transition matrix with size $N_1 * N_1$, $T_1[i,j] = t_1(v_i, v_j)$;
T_2 is HMM of G_2' transition matrix with size $N_2 * N_2$, $T_2[i,j] = t_2(u_i, u_j)$;
E is the connection interestingness matrix with size $N_1 * N_2$, $E[i,j] = e(v_i, u_j)$;
L is the desire size of the multi-connection bridging rule.

Ensure: *Bridge* is he most interesting multi-connection bridging rule, i.e., a sequence of connections: $Bridge = \{con_1, con_2, \cdots, con_L\}$. con is a two-tuples, $con_k = (i,j)$ indicates the connection $v_i \to u_j$ and $con_k[1] = i$, $con_k[2] = j$.

1: **for** each node $v_i \in G_1$ **do**
2:    **for** each node $u_j \in G_2$ **do**
3:       $\Gamma[1,i,j] = \log E[i,j]$
4:       $R[1,i,j] = 0$
5:    **end for**
6: **end for**
7: **for** $l = 2$ to L **do**
8:    **for** each node v_i in G_1 **do**
9:       **for** each node u_j in G_2 **do**
10:          $\Gamma[l,i,j] = \max_{m,n}[\Gamma(l-1,m,n) + \log T_1(m,i) + \log T_2(n,j) + \log E(i,j)]$
11:          $R[l,i,j] = \arg\max_{(m,n)}[\Gamma(l-1,m,n) + \log T_1(m,i) + \log T_2(n,j) + \log E(i,j)]$
12:       **end for**
13:    **end for**
14: **end for**
15: $con_L = \arg\max_{(m,n)}(\Gamma[L,m,n])$
16: **for** $l = L$ to 2 **do**
17:    $con_{l-1} = R[l, con_l[1], con_l[2]]$
18: **end for**
19: return *Bridge*

con_{l-1} in $Bridge = \{con_1, con_2, \cdots, con_L\}$ when $con_l = \{i,j\}$. From line 1 to 6, the algorithm initializes the matrix Γ and R. From line 7 to 14, the algorithm recursively calculates the probability that con_l equals to (i,j) and stores it in Γ in line 10, and records the corresponding path in R in line 11. From line 15 to 19, the algorithm finds and returns the most interesting MCBR using the path records in R.

Assume the number of edges in G_1, G_2 are M_1 and M_2, then the computational complexity of this Algorithm is $O(LM_1M_2)$. We can extend it to find the top K interesting MCBRs by replacing the *max* operator in the algorithm by an operator that returns the top K values, and replace *argmax* by an operator returns the corresponding arguments of the top K values. Theoretically, this will enable us to find all MCBRs in the database.

4 Experimental Results

The experiments were carried out on the computer with 2.5 GHz i3 CPU and 4GB of main memory. The transactional dataset we experiment with is generated by IBM *Quest Market-Basket Synthetic Data Generator* (`http://www.cs.`

loyola.edu/~cgiannel/assoc_gen.html). It contains 1000 transaction records, with an average number of items per transaction 5. Altogether there are 100 items in the transactional dataset, which are manually separated into 2 classes: class C_1 contains 60 items, each with 5 attributes; class C_2 contains 40 items, each with 4 attributes. The attribute of items is generated following the *Gaussian* distribution.

4.1 Performance Evaluation

The performance is measured by *Information entropy*, which quantitizes the information that a channel, signal or event carries [4]. Considering connections in a MCBR as channels connecting two classes, we can measure the significance of the connections in MCBR by the *joint entropy* [4].

For a connection $con = v_i \to u_j$, its joint entropy is defined as:

$$\begin{aligned} H(con) = &- \Sigma_{v_k \in N(v_i)} t(v_i, v_k) \log t(v_i, v_k) \\ &- \Sigma_{u_k \in N(u_j)} t(u_j, u_k) \log t(u_j, u_k), \end{aligned} \tag{7}$$

where $t(\cdot)$ is the transition probability between two items's corresponding states in HMMs.

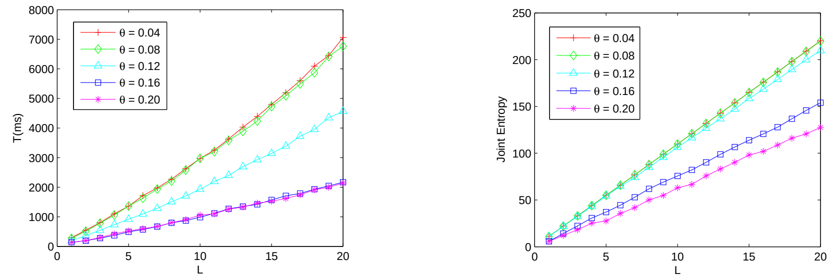
The *joint entropy* of a MCBR is the sum of *joint entropy* of its connections. Formally, for a MCBR $Bridge = \{con_1, con_2, \cdots, con_n\}$, its joint entropy is defined as: $H(Bridge) = \Sigma_{i=1}^{n} H(con_i)$. A MCBR $Bridge$ with larger $H(Bridge)$ value is regarded as more significant.

4.2 Performance under Different Parameters

The time complexity of *V-Bridge* algorithm is $O(LM_1M_2)$, where M_1 and M_2, the number of edges in G_1 and G_2, are highly related to the threshold θ when constructing the HMMs for two classes. Hence the numbers of edges in G_1 and G_2 are affecting the running time of the algorithm.

Fig. 4(a) shows how the changes θ and L can impact on the running time of *V-Bridge* algorithm. When threshold $\theta = 0.04$ or 0.08, it filters few edges between items and keeps most edges in the G_1 and G_2. The algorithm's running time increases sharply as L becomes larger, and reaches 7000 ms when $L = 20$. When threshold $\theta = 0.16$ or 0.20, it only keeps edges between items that are highly similar to each other in G_1 and G_2. The running times of the algorithm stay below 2000 s for different L from 1 to 20.

This experiment shows that a larger threshold θ will filter weak edges in G_1 and G_2 and reduce the complexity. However, it's possible that a large threshold θ filters edges that might support connection transitions in interesting MCBRs. In consideration of this, next we will investigate how the changes of θ and L will affect the quality of the discovered result.



(a) (b)

Fig. 4. (a) shows running time of *V-Bridge* algorithm with different θ and L; (b) shows average joint entropy of *MCBRs* found by *V-Bridge* algorithm with different θ and L.

4.3 Quality of Results under Different Parameters

Here we investigate how the changes of θ and L will affect the quality of the discovered result. Fig. 4(b) shows the average *joint entropy* of the MCBRs found by *V-Bridge* algorithm with different θ and L.

It can be observed that, the average *joint entropy* is increasing as L becomes larger. This is because that the *MCBR* with larger size contains more connections and can give us more information. The significance of *MCBRs* will decrease when θ becomes larger. This could be because that the average degree of each item in G_1 and G_2 declines and some interesting edges are filtered.

From the experiment result we can know that, a proper threshold θ should be carefully selected according to different properties of different datasets. A good threshold θ can reduce the complexity of the problem and the algorithm's running time, meanwhile, without the risk of filtering edges that may play a role in an interesting MCBR. For example, when simultaneously consider Fig. 4(a) and Fig. 4(b), we find that $\theta = 0.12$ is good for our synthetic dataset. It helps to reduce the running time at the same time sacrifices little significance of the MCBRs.

5 Conclusion

In this paper, multi-connection bridging rule(MCBR) is proposed to discover important information between different conceptual classes. It can reveal the association relationship among items both from the same class and from different classes. We proposed an effective algorithm *V-Bridge* to discover MCBRs in database with the following contributions:

- We generalize the single connection bridging rule into *multi-connection bridging rule(MCBR)* to meet the real-life application scenarios.
- We represent the data in HMMs to reduce the problem's complexity and highlight the topology property of dataset.

– We evaluate the interestingness of connections in MCBR by performing a semi-Markov random work on HMMs.

Experiments on synthetic dataset show that the proposed *V-Bridge* algorithm can efficiently find interesting MCBRs.

In the future, we plan to improve the proposed algorithm to adapt to multiple classes and furthermore examine its effectiveness with real data, such as the miRNA expression database and *protein-protein interaction* (PPI) network.

References

1. Chad Creighton and Samir Hanash. Mining gene expression databases for association rules. *Bioinformatics*, 19(1):79–86, 2003.
2. Daniel Sánchez, MA Vila, L Cerda, and José-María Serrano. Association rules applied to credit card fraud detection. *Expert Systems with Applications*, 36(2):3630–3640, 2009.
3. Shichao Zhang, Feng Chen, Xindong Wu, and Chengqi Zhang. Identifying bridging rules between conceptual clusters. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 815–820. ACM, 2006.
4. Shichao Zhang, Feng Chen, Zhi Jin, and Ruili Wang. Mining class-bridge rules based on rough sets. *Expert Systems with Applications*, 36(3):6453–6460, 2009.
5. Anton J Enright, Bino John, Ulrike Gaul, Thomas Tuschl, Chris Sander, Debora S Marks, et al. Microrna targets in drosophila. *Genome biology*, 5(1):R1–R1, 2004.
6. Praveen Sethupathy, Molly Megraw, and Artemis G Hatzigeorgiou. A guide through present computational approaches for the identification of mammalian microrna targets. *Nature methods*, 3(11):881–886, 2006.
7. Sayed Mohammad Ebrahim Sahraeian and Byung-Jun Yoon. Fast network querying algorithm for searching large-scale biological networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 6008–6011. IEEE, 2011.
8. Longin Jan Latecki and Rolf Lakamper. Shape similarity measure based on correspondence of visual parts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1185–1190, 2000.
9. Haiying Wang, Francisco Azuaje, Olivier Bodenreider, and Joaquín Dopazo. Gene expression correlation and gene ontology-based similarity: an assessment of quantitative relationships. In *Computational Intelligence in Bioinformatics and Computational Biology, 2004. CIBCB'04. Proceedings of the 2004 IEEE Symposium on*, pages 25–31. IEEE, 2004.
10. Qingfeng Chen, Gang Li, and Yi-Ping Phoebe Chen. Interval-based distance function for identifying rna structure candidates. *Journal of theoretical biology*, 269(1):280–286, 2011.
11. Toby Segaran. Programming collective intelligence. 2008.
12. Edward R Omiecinski. Alternative interest measures for mining associations in databases. *Knowledge and Data Engineering, IEEE Transactions on*, 15(1):57–69, 2003.
13. Xiaoning Qian, Sayed ME Sahraeian, and Byung-Jun Yoon. Enhancing the accuracy of hmm-based conserved pathway prediction using global correspondence scores. *BMC bioinformatics*, 12(Suppl 10):S6, 2011.

14. G David Forney Jr. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.