Knowledge-Based Systems 86 (2015) 1–10

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Multi-view attribute reduction model for traffic bottleneck analysis

X.D. Yue [a,*], L.B. Cao [b], D.Q. Miao [c], Y.F. Chen [c], B. Xu [d]

[a] School of Computer Engineering and Science, Shanghai University, Shanghai 201804, China
[b] Advanced Analytics Institute, University of Technology, Sydney, NSW 2007, Australia
[c] School of Electronics and Information, Tongji University, Shanghai 200092, China
[d] School of Transportation Engineering, Tongji University, Shanghai 200092, China

ARTICLE INFO

Article history:
Received 15 October 2014
Received in revised form 19 March 2015
Accepted 20 March 2015
Available online 31 March 2015

Keywords:
Pattern mining
Traffic bottleneck
User preference

ABSTRACT

In the field of traffic bottleneck analysis, it is expected to discover traffic congestion patterns from the reports of road conditions. However, data patterns mined by existing KDD algorithms may not coincide with the real application requirements. Different from academic researchers, traffic management officers do not pursue the most frequent patterns but always hold multiple views for mining task to facilitate traffic planning. They expect to study the correlation between traffic congestion and various kinds of road properties, especially the road properties easily to be improved. In this multi-view analysis, each view actually denotes a kind of user preference of road properties. Thus it is required to integrate user-defined attribute preferences into pattern mining process. To tackle this problem, we propose a multi-view attribute reduction model to discover the patterns of user interests. In this model, user views are expressed with attribute preferences and formally represented by attribute orders. Based on this, we implement a workflow of multi-view traffic bottleneck analysis, which consists of data preprocessing, preference representation and congestion pattern mining. We validate our approach based on the reports of road conditions from Shanghai. Experimental results show that the resultant multi-view mining outcomes are effective for analyzing congestion causes and traffic management.

1. Introduction

In the field of intelligent transportation, *knowledge discovery and data mining* (KDD) methods have been widely utilized to analyze various kinds of traffic data to construct decision support systems for traffic management [7,20,41,42]. For these data-driven traffic management systems, one of the most important tasks is analyzing the causes of traffic bottlenecks and taking action to alleviate congestion [2,17,24]. The data analysis of traffic bottlenecks are generally performed on either spatiotemporal data [15,16] or traffic reports of road conditions [3]. Specifically, for the traffic reports of road conditions, it is expected to analyze traffic bottlenecks through discovering congestion patterns from the table-formed data. To achieve this, most existing works directly apply the methodologies of association rule mining on traffic reports to obtain congestion patterns. Techniques of association rule mining and association analysis are employed to predict traffic network flows [11]. The algorithms of frequent pattern mining are used to discover simultaneously congested link-sets in a road network [28]. Additionally, a strategy of association rule acquisition based on group decision-making is proposed to identify the traffic states of regional road sections [14].

Depending on the above association rule/pattern mining methods, we can obtain abundant data patterns from traffic reports. However, lots of these patterns may be out of users' interests and not actionable enough to support traffic management. Because most existing data mining methods just focus on statistical significance of attributes, such as cooccurrence and discernibility to generate patterns but neglect user preferences and requirements. Without considering user preferences and requirements, the data mining algorithms provide same data patterns for different users and thus actually produce single-view analysis results. For real applications, the single-view analysis results are not sufficient and flexible enough to implement a solution for traffic improvement. Thus it is necessary to design user-oriented data mining methods to analyze traffic reports from multiple views.

Especially for the applications of traffic bottleneck analysis, traffic officers are eager to study the causes of traffic congestion from different perspectives. First, they hold the views to facilitate traffic planning. The discovered patterns should indicate the relationship between traffic congestion and the road properties that are easy to

* Corresponding author.
E-mail address: yswantfly@gmail.com (X.D. Yue).

http://dx.doi.org/10.1016/j.knosys.2015.03.022
0950-7051/

be improved. Applying traditional data mining algorithms to road condition reports, the obtained congestion patterns generally consist of the attributes about road construction, such as road width, length and average delay. But in modern cities, these factors are always difficult to change, because reconstructing roads are really high-cost tasks. Thus officers expect to find the dependence between traffic congestion and the controllable factors, such as car/bike separators, zebra crossing, and bus lanes. Second, officers hold the views to involve experts' experiences into bottleneck analysis. The road properties being considered important for congestion formation should have high priority to emerge in the final patterns. The experts' experiences are helpful to discover, employ and interpret the actionable patterns.

As mentioned above, each view for traffic bottleneck analysis actually denotes a kind of preference of road properties. To achieve the multi-view analysis, it is required to integrate user-defined attribute preferences into pattern mining process. In other words, the desired data mining methods should be able to reduce and organize data attributes according to user preferences to generate the final patterns [8,18]. Aiming at the problems, we propose a multi-view attributes reduction model to discover the congestion patterns for traffic bottleneck analysis. The reduction model formulates user views with attribute preferences and extracts data patterns referring to the user-defined attribute priorities. Applying the multi-view reduction model to the traffic reports of road conditions, we can discover the congestion patterns of various kinds of user interests. The multi-view mining outcomes can induce comprehensive and practicable knowledge and lead to an overall analysis of traffic bottleneck causes. Our contributions are summarized as follows.

- *Propose a multi-view attribute reduction model to discover patterns of user interests.* In the model, user views are expressed with attribute preferences and formally represented by attribute orders. To implement the order-based attribute reduction, a data structure of 2D linked list is designed to storage item-discerning elements and discernibility thresholds are set to filter out trivial attributes in patterns.
- *Propose a workflow for traffic bottleneck analysis based on the multi-view reduction model.* The workflow consists of data collection and preprocessing of road conditions, user preference representation and attribute reduction for congestion pattern mining. It supports users to analyze the causes of urban traffic congestion from the views of traffic management.

The rest of this paper is organized as follows: Section 2 reviews the related work. Section 3 introduces a multi-view attribute reduction model, which includes user view representation, reduction algorithm implementation and theoretic model analysis. Based on the multi-view reduction model, Section 4 proposes a user-oriented workflow to analyze the causes of traffic bottlenecks. In Section 5, experimental results validate our approach is effective for overall traffic congestion analysis and customized knowledge discovery. The work conclusion is given in Section 6.

2. Related work

The basic idea of this work originates from the methodology of *actionable knowledge discovery* (AKD), which aims to find the actionable patterns which are friendly enough for business people to interpret, validate and action [5,33,35]. The key focus of actionable pattern mining is to involve user factors into data mining workflow [1,9]. Different from the traditional pattern evaluation of statistical significance, the evaluation of actionable patterns consists of the objective and subjective interestingness measures, which recognize to what extent a pattern is of interest to particular user preferences [19,23,31]. Specifically, the probability-based belief was used to describe user confidences of unexpected rules and the profit and utility mining frameworks were designed to measure the business values of patterns [6,34]. Based on the revised interestingness measures, domain-driven data mining strategies were further proposed to discover the workable knowledge for real applications [4,40]. In this paper, we expect to instantiate an actionable pattern mining model to cater for multiple user views. This model is constructed based on the process of attribute reduction.

As an inductive learning tool, *attribute reduction* extracts valuable patterns from table-formed information systems through reducing redundant attributes to attribute reducts [22,25,27]. Specifically, attribute reduction algorithms are good at processing uncertain data [21,30,36] and efficient to discover the rule-type knowledge from data tables [26,39]. The table-formed information systems are defined as $IS = (U, C, f, V)$, in which U is a finite set of data items, called the universe, C is a finite set of attributes to depict items, V denotes the domain of attribute values, and f is the mapping from U to V, which assigns particular attribute values to items. For classification tasks, we consider the information systems with decision attributes D, $DS = (U, C \cup D, f, V)$. Such systems are called *decision systems*, in which the attributes of C are viewed as conditions. In a decision system DS, for any $x, y \in U$, if $C(x) = C(y) \Rightarrow D(x) = D(y)$, system DS is *consistent*, otherwise it is *inconsistent*. In this paper, we just consider the consistent decision systems and assume D consists of a single attribute $D = \{d\}$, d is a decision attribute for labeling the class of each object. Example 1 illustrates a decision system about customer evaluation of cars.[1]

Example 1. Decision system 'Car evaluation': conditional attributes 'Buying price', 'Maintenance cost', 'Number of doors', 'Capacity of persons', 'Size of luggage boot', 'Safety of cars' depict the properties of cars and the decision attribute 'Accepted' reflects the evaluation of customers (Accepted or Unaccepted). The details are presented in Table 1.

In decision systems, the classification ability of a conditional attribute set A can be evaluated through constructing its *positive region* relative to the decision d. The positive region $POS_A(U/d)$ consists of all the objects that can be correctly classified with the attributes A. Attribute reduction aims to reduce redundant conditional attributes to an *attribute reduct* while preserve a certain classification property. Attribute reduct can be formally defined from the view of positive region preservation.

Definition 1. *Attribute Reduct.* Given a decision system $DS = (U, C \cup \{d\}, f, V)$, an attribute set $R \subseteq C$ is a d-reduct of C, iff

$$\begin{aligned} &(1) \quad POS_R(U/d) = POS_C(U/d) \\ &(2) \quad \forall r \in R,\ POS_{R-\{r\}}(U/d) \neq POS_R(U/d) \end{aligned} \tag{1}$$

Condition (1) requires a reduct R to have the same classification ability as the conditional attribute set C. For an attribute $r \in R$, if $POS_{R-\{r\}}(U/d) = POS_R(U/d)$, r is d-dispensable in R, otherwise r is d-indispensable. If all the attributes in R are d-indispensable, R is independent with respect to d, otherwise R is dependent. Obviously, condition (2) requires that a reduct R should be independent.

According to Definition 1, we notice that the reduct of a decision system may not be unique. The reduct of the minimum attributes is considered the *optimal reduct*. Such as in Example 1, attribute sets {Persons, Safety} and {Buying, Maintenance, Doors, Luggage

[1] The demo decision system is generated from the UCI dataset 'Car evaluation'.

Table 1
Decision system of 'Car Evaluation'.

	Buying	Maint	Doors	Persons	Luggage boot	Safety	Accepted
01	High	High	2	2	Small	High	U
02	High	High	2	4	Med	High	A
03	High	High	2	4	Big	Low	U
04	High	Low	2	4	Small	High	A
05	High	Low	2	4	Big	Low	U
06	High	Low	3	4	Big	Low	U
07	Med	High	2	4	Small	High	A
08	Med	High	3	4	Small	Low	U
09	Med	High	3	4	Med	Low	U
10	Med	High	3	4	Big	High	A
11	Med	Low	3	2	Big	High	U
12	Low	High	2	2	Big	Low	U
13	Low	High	2	4	Small	High	A
14	Low	Low	2	2	Big	Med	U
15	Low	Low	2	4	Small	Med	A

boot} are both reducts, in which {Persons, Safety} is the optimal one. Because computing the optimal reduct of a given system is an NP-hard problem, many heuristic algorithms are designed to find suboptimal solutions. However, the patterns obtained from the reduct of minimum attributes may not coincide with the user requirements. For example, the reduct {Persons, Safety} cannot reflect the relationship between price, maintenance cost and purchase for salesmen.

To involve prior information into attribute reduction, Wang first designed a reduction algorithm based on attribute orders and analyzed its properties. The reduction process is built up on the basis of traditional discernibility matrix [32]. Yao proposed a formal framework to represent the attribute bias for machine learning. This framework formulates the user preferences with different kinds of attribute orders and describes these orders by both qualitative and quantitative judgements [38]. Zhao also utilized attribute orders to represent user requirements and designed an algorithm to find a particular reduct according to user interests. The quality of the output reduct relies on the free attributes selected in reduction process [43]. Moreover, Han and Wang further analyzed the relationship between attribute orders and attribute reducts and proposed Secondary Attribute Theory to judge whether similar attribute orders produce the same reduct [13].

Our approach is based on the works mentioned above. To capture the patterns of user interests efficiently, we improve the existing attribute reduction methods from the aspects of preference representation and reduction implementation. First, to guarantee the reducts independent, we use only strict total orders to represent attribute preferences, and define the priority of attribute sets by their token attributes. Second, we design a novel data structure of 2-dimensional list to storage discerning elements and further compress it with absorption laws. Finally, to avoid overfitting in reduction process, we set discernibility thresholds to filter out trivial attributes. Different from the traditional reduction methods focusing on attribute discernibility, our approach considers both the attribute discernibility and preference to generate reducts. The attribute reduct with preference is still Pawlak reduct, i.e. an independent attribute subset which can keep the classification ability as the whole conditional attributes. The difference is that we do not pursue the optimal (shortest) reduct but try to search a reduct in solution space to contain the attributes of user interests as much as possible.

3. Multi-view attribute reduction model

The user-oriented approach for traffic data analysis is proposed on the basis of multi-view attribute reduction. Through formulating user views by attribute preferences, the attribute reduction process can generate the attribute reducts of user interests and thus induce the data patterns according to user views.

3.1. Modeling user view with attribute preference

The views of user in data analysis actually represent the users' preferences of data attributes. Attributes preferences express various kinds of user interests and lead to the multi-view analysis results. In the case of vehicle purchase, when a customer tends to evaluate cars from the view of safety, it means the customer considers the safety property is more important than the other properties of cars, such as price and comfort. The user bias of data attributes can be formally represented in the form of strict order of attributes, i.e. the attribute priority.[2]

Definition 2. *Attribute Order.* Given a decision system $DS = (U, C \cup D, f, V)$, in which $C = \{a_1, a_2, \ldots, a_m\}$ is the set of conditional attributes and $m = |C|$. A strict attribute order S on set C is defined as,

$$S : a_1 \prec a_2 \prec \ldots \prec a_m \quad (2)$$

The order S is an attribute sequence which defines the priority of attributes, such as $a_1 \prec a_2$ denotes the attribute a_1 is prior to a_2.

Based on the attribute order, we can further define the order of attribute sets. For an attribute set, the set priority is decided by the priority of its member attributes.

Definition 3. *Token Attribute.* Given a decision system $DS = (U, C \cup D, f, V), S$ is an attribute order on set C. For a subset of attributes $A \subset C$, sort the attributes of A according to S and obtain the ordered sequence $A^S = \{a_i, \ldots\}$. Obviously, attribute a_i owns the highest priority in set A and is named the token attribute of A.

Definition 4. *Attribute Set Order.* Given a decision system $DS = (U, C \cup D, f, V)$ and an order S on C, for two attribute subsets $A,\ B \subset C$, sort the attributes of A and B with S and obtain $A^S = \{a_i, \ldots\}$ and $B^S = \{a_j, \ldots\}$, in which a_i, a_j are the token attributes of A and B respectively. According to S, if $a_i \neq a_j$ and a_i is prior to a_j $(a_i \prec a_j)$, the sequence A^S is considered prior to B^S $(A^S \prec B^S)$ and thus the attribute set A is prior to B $(A \prec B)$; if $a_i = a_j$, the attribute sets A and B have the same priority, i.e. $A \cong B$.

Example 2. For the decision system 'car evaluation' shown in Example 1, suppose we have the following attribute order S,

$$Safety \prec Buying \prec Maint \prec Doors \prec Persons \prec Luggage\ boot \quad (3)$$

Order S presents a customer preference for buying cars: safety will be first considered and then the car price and maintenance cost, in the meantime, the property of comfort is considered of no importance. Given two attribute sets {Safety, Doors, Persons} and {Price, Maint}, 'Safety' and 'Price' are the token attributes. Referring to the order S, Safety $\prec$ Price, thus {Safety, Doors, Persons} $\prec$ {Price, Maint}. If we change the attribute 'Safety' to 'Price' in the first set, the two sets have the same priority, i.e. {Price, Maint} $\cong$ {Price, Doors, Persons}.

[2] In decision systems, we just pay attention to the priority of conditional attributes, which is defined as similar as in information systems.

3.2. Reduct algorithm with attribute preference

Based on the definitions of attribute preference, the user bias can be involved into attribute reduction process. This process aims to capture the attributes of user interests in reducts. The reduct algorithm with attribute preference is built on the basis of discerning elements. Each discerning element consists of the attributes to distinguish a pair of items with different class labels.

3.2.1. Data structure

Considering spatial complexity, we design a 2-dimensional list to storage the discerning elements. The first-dimension list represents M conditional attributes of system. On the second dimension, each attribute a_i has a linked list of discerning elements, in which all the elements have the same token attribute a_i. To further simplify the 2-dimensional list, we use the following strategies to remove redundant discerning elements.

- Because the comparison of two items is symmetric, i.e. the discerning element $e(i,j) = e(j,i)$, for each item pair, just one copy of their different attributes is stored in the list.
- Reduce redundant discerning elements based on Absorption Law in Set Theory [37]. For two elements e and e', if $e \subseteq e'$, e' can be replaced by e in the list.

Algorithm 1. Constructing List of Discerning Elements.

Input:
Decision system $DS = (U, C \bigcup \{d\}, V, f),\ |C| = m,\ |U| = n$,
attribute preference $S : a_1 \prec a_2 \prec \ldots \prec a_m$
Output: 2-Dimensional list of discerning elements
Initialize an array of m attributes, in which the kth node points to the linked list of discerning elements of the token attribute a_k;
for $i = 1$ to n **do**
  for $j = i + 1$ to n **do**
    Compare the class of item i and item $j : d(i),\ d(j)$;
    if $d(i) \neq d(j)$ **then**
      Integrate the different attributes between item i and item j to form the discerning element $e(i,j)$;
      Sort the attributes of $e(i,j)$ according to S and obtain the token attribute $a_k,\ 1 \leqslant k \leqslant m$;
      for each discerning element e' in the kth list
        if $e(i,j) \cap e' = e(i,j)$ **then**
          Replace e' with $e(i,j)$;
        else
          if $e(i,j) \cap e' = e'$ **then**
            Break and check the next pair of items;
          end if
        end if
      end for
      If the inclusion conditions are not satisfied, insert the element $e(i,j)$ into the kth linked list;
    end if
  end for
end for
Output 2-dimensional list of discerning elements.

Algorithm 1 shows how to construct a 2-dimensional list of discerning elements for a decision system. Before being inserted into the linked list, the attributes in every discerning element are sorted by the given order. This strategy facilitates the computation of token attributes and the comparison of attributes in further reduction process. Moreover, to balance computational time and spatial complexity, Absorption Law is applied to only the elements having the same token attribute rather than the total discerning elements to remove redundancy. Given the decision system of Example 1 and the attribute preference shown in formula (3), Fig. 1 illustrates the 2-dimensional list of the discerning elements. First we obtain a group of attribute sets (discerning elements) to distinguish all the items of different classes in decision system, such as {Luggage boot, Safety}, {Maint, Luggage boot, Safety}, etc. Ranking the attributes in each discerning element by the given preference, the elements above turn into {Safety, Luggage boot}, {Safety, Maint, Luggage boot}. Because the bigger element includes the smaller one, we just insert {Safety, Luggage boot} into the list.

3.2.2. Reduct algorithm

The attribute reduction algorithm involving attribute preferences is implemented through ranking the discerning elements. Given an attribute order $a_1 \prec \ldots \prec a_m$ for a decision system, referring to Definition 4, the binary relation of equal priority '$\cong$' between two attribute sets depends on their token attributes. Obviously, the relation '$\cong$' is reflexive, symmetric and transitive, thus can partition all the discerning elements (attribute subsets) M into m disjoint equivalence classes, $M/\cong: \{[a_1], \ldots, [a_m]\}$. Each equivalence class $[a_i]$ is denoted by the common token attribute and all the elements of it have the same priority. Thus we also have $[a_1] \prec \ldots \prec [a_m]$. Through partitioning the discerning elements, item discernibility is graded to different levels according to the predefined attribute preference. Recalling the data structure, the second-dimension lists represent the partition of discerning elements. Based on this, the reduction algorithm with attribute preference is designed as follows.

Algorithm 2. Reduct with Attribute Preference (RAP).

Input:
Decision system $DS = (U, C \bigcup D, V, f),\ |C| = m$,
Attribute preference, $S : a_1 \prec a_2 \prec \ldots \prec a_m$
Output: Attribute reduct R based on the preference S
Initialize reduct $R = \emptyset$;
Construct the 2-dimensional list M to partition all the discerning elements. m linked lists represent the equivalence classes of m token attributes, $\{[a_1], \ldots, [a_m]\}$, and class priority $[a_1] \prec \ldots \prec [a_m]$;
while $M \neq \emptyset$ **do**
  Choose the class one by one from low to high priority;
  for $i = m$ to 1 **do**
    Check the irreplaceable discernibility of a_i;
    if $|[a_i]| < T_{id}$ **then**
      Delete all the elements of $[a_i]$ from M;
    else
      Browse M forward and count the number of elements which contain the attribute a_i, the number is denoted by K_{a_i};
      Check the general discernibility of a_i;
      if $K_{a_i} < T_d$ **then**
        Delete all the elements of $[a_i]$ from M;
      else
        Add attribute a_i to reduct R and delete all the elements containing a_i from M;
      end if
    end if
  end for
end while
Output attribute reduct R.

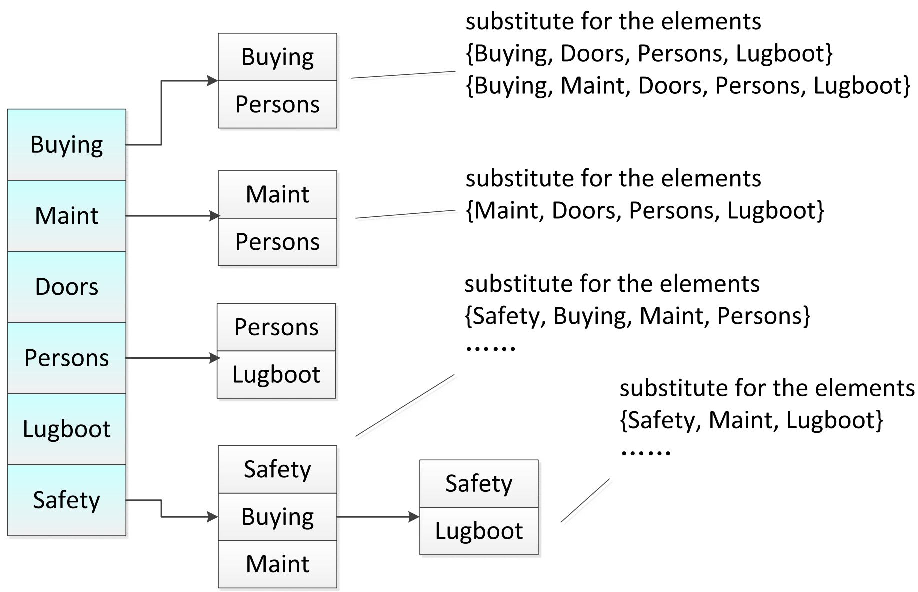

Fig. 1. Data structure of discerning elements.

As shown in Algorithm 2, in reduction process, attributes are selected according to their priority and discernibility. For an attribute with high priority, it takes precedence to distinguish item pairs and thus is more likely to occur in the reduct. On the other hand, for the attribute with low priority, if it is the token attribute of many discerning elements, this indicates the attribute is irreplaceable and should be added to the reduct. Furthermore, the discernibility of an attribute consists of the general and irreplaceable discerning ability. For an attribute a_i, the number of the elements in class $[a_i]$ denotes its irreplaceable discernibility, and the number of all the discerning elements containing a_i represents its general discernibility. To filter out the trivial attributes, we adopt two thresholds of discernibility to guarantee reducts concise. The threshold T_d is used to evaluate general discernibility of an attribute. In each iteration, only the attributes which distinguish more than T_d item pairs are considered as the candidates. The threshold T_{id} is used to measure attribute's irreplaceable discernibility. With the previous selected attributes, irreplaceable discernibility of a candidate attribute is represented by the number of item pairs only can be distinguished by it. In algorithm implementation, we set $T_d = \lceil T/m \rceil$ and $T_{id} = \lceil 0.02T \rceil$, in which m is attribute number and T is the number of all the discerning elements.

Through attribute reduction process, the original conditional attributes are reduced to attribute reduct for classification. Integrating attribute values into the reduct, we can obtain the corresponding data patterns. With different attribute preferences, RAP generates different attribute reducts and thus leads to the patterns of various kinds of user interests.

3.3. Model analysis

Involving the attributes of user interests may lead to attribute reducts including more attributes than the traditional patterns. Thus it is required to analyze the redundancy of the attribute reducts with user preferences. Next we prove that the attribute reducts obtained by the proposed reduction model are still *independent*, i.e. all the attributes in a reduct are necessary for classification [25]. This means the reducts contain no redundant attributes and guarantee the high-quality patterns.

Theorem 1. *Given a decision system $DS = (U, C \cup D, V, f)$, $|C| = m$ and an attribute preference $S : a_1 \prec a_2 \prec \ldots \prec a_m$, suppose the partition of discerning elements induced by S is $\{[a_1], \ldots, [a_m]\}$. For any attributes $a_p, a_q \in C$, if the attribute priority $a_p \prec a_q$, then $\forall e \in [a_q]$, $e \cap \{a_p\} = \emptyset$.*

Proof. For any element $e \in [a_q]$, e is a subset of attributes to discern an item pair and its token attribute is a_q. According to Definition 3, any attribute $a \in e$ cannot be prior to the attribute a_q, $a_q \preceq a$. Because $a_p \prec a_q$, we have $a_p \prec a_q \preceq a$, and $a_p \prec a$, the attribute a_p is prior to any attribute in e. Thus for any element e in class $[a_q]$, does not contain the attribute a_p, i.e. $e \cap \{a_p\} = \emptyset$. □

As shown in Theorem 1, for a class of discerning elements, all its members cannot contain the attributes which are prior to its token attribute. This indicates that for an attribute a, if its discerning class $[a]$ is not empty, a has some discernibility cannot be replaced by the attributes prior to it. Generally, the more elements $[a]$ has, the more important attribute a is for discerning item pairs. Even owning low priority, attribute a should be selected into reduct due to its irreplaceable discernibility. Based on Theorem 1, we can further check the dependency of the reduct attributes.

Theorem 2. *Given a decision system $DS = (U, C \cup D, V, f)$, $|C| = m$ and an attribute preference $S : a_1 \prec a_2 \prec \ldots \prec a_m$, the attribute reduct obtained by algorithm RAP is independent.*

Proof. Suppose R be a reduct obtained by algorithm RAP, R consists of k attributes, $R : \{r_1, r_2, \ldots, r_k\}, r_i \in C$, $1 \leqslant i \leqslant k$, and the attribute priority $a_1 \preceq r_1 \prec r_2 \prec \ldots \prec r_k \preceq a_m$. In order to prove a reduct R is independent, referring to the definition of attribute reduct [25], we should demonstrate that every attribute in reduct R is indispensable.

First we demonstrate attribute r_k, which has the lowest priority in R, is indispensable in reduct R. Since RAP selects attributes from low to high priority, r_k is the first selected attribute and $r_k = \min \text{priority}\{a_i | [a_i] \neq \emptyset\}$. $\forall r \in R - \{r_k\}$, $r \prec r_k$, according to Theorem 1, we have $\forall e \in [r_k]$, $e \cap \{r\} = \emptyset$. This means the item pairs discerned by r_k cannot be distinguished by the other attributes in reduct, thus r_k has irreplaceable discernibility and is indispensable in R.

Next we demonstrate the other attributes in R indispensable. $\forall r \in R - \{r_k\}$, suppose M^r be the set of discerning elements before selecting attribute r, referring to RAP algorithm, all the elements containing the reduct attributes of the priority lower than r have been removed from M^r, i.e. $\forall r' : r' \in R \wedge r \prec r',\ \forall e \in M^r, e \cap \{r'\} = \emptyset$. Since $[r] \subset M^r$, we have $\forall e \in [r],\ e \cap \{r'\} = \emptyset$. For the reduct attributes prior to r, $\forall r^* : r^* \in R \wedge r^* \prec r$, according to Theorem 1, $\forall e \in [r],\ e \cap \{r^*\} = \emptyset$. As mentioned above, for the reduct attributes whether prior or posterior to r, r has the irreplaceable discernibility. Thus r is indispensable in R.

To sum up, all the reduct attributes $\{r_1, r_2, \ldots, r_k\}$ are indispensable in R. Thus the reduct obtained by RAP is independent. $\square$

From Theorems 1 and 2, we know that adding the attribute preference as prior information to attribute reduction model is helpful to not only discover patterns from user views but also guarantee the patterns without redundancy.

4. Multi-view traffic bottleneck analysis

Based on the multi-view attribute reduction model, a user-oriented workflow for traffic bottleneck analysis can be created. The workflow consists of data collection and preprocessing of road conditions, user preference representation, road attribute reduction and congestion pattern generation. Involving user preferences in pattern mining process, this workflow supports users to analyze the causes of traffic bottlenecks from multiple views of traffic management.

4.1. Data preparation

The target of traffic bottleneck analysis is to study the causes of road congestion. Urban traffic congestion is not only related to road construction but also city planning and traffic management. As is well known, in modern cities, road reconstruction generally costs too much in both money and time. Thus we cannot solve urban traffic problems just through widening roads. A reasonable solution to traffic congestion should consider multiple aspects, which include city layout, infrastructure construction, policies and laws, traffic programming and people traveling mode. To achieve this solution, first of all, we should know the factors which may lead to traffic block.

In this paper, we adopt 14 attributes as candidate factors to analyze urban traffic congestion. These attributes fully depict the properties of a road from the views of construction, programming and environment respectively. Table 2 lists the attributes and the corresponding descriptions. We can see that the attributes 1–5 are inherent road properties about construction and traffic capacity, attributes 6–11 are related to road programming and infrastructure setup, and attributes 12–14 reflect region environment.

Using attribute reduction model to discover data patterns of traffic congestion, it is necessary to discretize attribute values to obtain symbolic descriptions of road properties. The criteria of discretization are made through referring to China Urban Road Traffic Performance Index and expert experiences. Table 3 shows the discretization criteria of all attributes. Besides the conditional attributes of road properties, the decision attribute value to judge whether a road is a traffic bottleneck is assigned by domain experts.

4.2. User preference representation

The user preferences of road properties for traffic bottleneck analysis come from either application requirements or domain knowledge. On one hand, because it is difficult to change road construction, traffic officers pay more attention to the effects of traffic environment and programming on road congestion. On the other hand, if experts have the experience that some factors are critical to cause congestion, it is required to enhance the priority of these factors in bottleneck analysis.

As introduced in Section 3, the user preferences of road properties are formally represented by attribute orders. Thus we can model the bias of users in traffic bottleneck analysis through setting the priority of road properties. To facilitate the priority setting, we divide 14 road properties into 5 groups from the aspects of road construction, infrastructure setup, traffic environment, traffic capacity and bus station. To obtain a total order of properties, users can first rank property groups and then adjust property order within each group. This strategy is implemented through a friendly user interface.

Table 2
Properties of road.

	Property	Description
01	Length	Length of road (km)
02	Lane number	Number of road lanes
03	Joint roads number	Number of connecting roads
04	Capacity difference	Difference of traffic capacity to connecting roads (pcu/h)
05	Average delay	Average time delay of a road, computed by the formula $T_{delay} = T/L$, in which T is the time delay at road junction and L is road length (s/m)
06	Middle separator	Type of middle separator of a road
07	Car/bike separator	Type of car/bike separator of a road
08	Zebra strips	Whether a road has zebra strips
09	Exit/entrance number	Number of exits and entrances in a road
10	Bus station	Type of bus station
11	Bus-only lane	Whether a road has bus-only lane
12	Region	Location of region
13	Land type	Type of region land
14	Traffic volume	Traffic volume of a region (10,000/Day)

Table 3
Discrete values of road properties.

Length	Value	Middle separator	Value	Car/bike separator	Value
0–200	1	No separator	0	No separator	0
200–500	2	Divider	1	Divider	1
500–800	3	Pier		Pier	
>800	4	Strip	2	Strip	2
				No vehicle lane	3
Zebra strips	Value	Traffic volume	Value	Region	Value
No	0	0–18.75	1	Outer middle-ring	1
Yes	1	18.75–37.5	2	Around middle-ring	2
		>37.5	3	Within inner-ring	3
Lane number	Value	Exit/entrance	Value	Bus lane	Value
1	1	0 or 1	1	No	0
2	2	2	2	Yes	1
>2	3	>2	3		
Bus station	Value	Land type	Value	Capacity difference	Value
No station	0	Entertainment	1	0	0
Bus bay	1	Residential	2	0–1300	1
Roadside stop	2	Business	4	1300–5400	2
		Government	8	>5400	3
Joint roads number	Value	Average delay	Value		
0–5	1	0–0.08	0		
6	2	0.08–0.14	1		
>6	3	>0.14	2		

For example, if traffic officers are interested in road infrastructure setup and expect to study the ways to avoid traffic congestion through optimizing it, the following road property orders can be used to express this type of user preferences.

(1) Preference for *infrastructure setup* and *environment*: {Car/bike separator $\prec$ Middle separator $\prec$ Zebra strips $\prec$ Exit/entrance} $\prec$ {Region $\prec$ Land type $\prec$ Traffic volume} $\prec$ {Bus lane $\prec$ Bus station} $\prec$ {Joint roads number $\prec$ Capacity difference} $\prec$ {Lane number $\prec$ Length $\prec$ Average delay}
(2) Preference for *infrastructure setup* and *bus station*: {Car/bike separator $\prec$ Middle separator $\prec$ Zebra strips $\prec$ Exit/entrance} $\prec$ {Bus lane $\prec$ Bus station} $\prec$ {Region $\prec$ Land type $\prec$ Traffic volume} $\prec$ {Joint roads number $\prec$ Capacity difference} $\prec$ {Lane number $\prec$ Length $\prec$ Average delay}
(3) Preference for *infrastructure setup* and *traffic capacity*: {Car/bike separator $\prec$ Middle separator $\prec$ Zebra strips $\prec$ Exit/entrance} $\prec$ {Joint roads number $\prec$ Capacity difference} $\prec$ {Bus lane $\prec$ Bus station} $\prec$ {Region $\prec$ Land type $\prec$ Traffic volume} $\prec$ {Lane number $\prec$ Length $\prec$ Average delay}
(4) Preference for *infrastructure setup* and *road construction*: {Car/bike separator $\prec$ Middle separator $\prec$ Zebra strips $\prec$ Exit/entrance} $\prec$ {Lane number $\prec$ Length $\prec$ Average delay} $\prec$ {Joint road number $\prec$ Capacity difference} $\prec$ {Bus lane $\prec$ Bus station} $\prec$ {Region $\prec$ Land type $\prec$ Traffic volume}

See the property orders above, through combining with other kinds of road properties, the effects of infrastructure setup on traffic congestion can be fully analyzed from multiple views. In property order (1), the road properties related to separator setting and region environment are endowed with high priority. This indicates a user preference for discovering the dependency between these factors and traffic congestion. Thus the data mining task is to find the patterns to show how road infrastructure setup influences traffic condition in different types of traffic environments. Similarly, based on the orders (2–4), the discovered data patterns can reflect the comprehensive effects of infrastructure setup on traffic congestion coupling with bus station, traffic capacity and road construction respectively. The experimental results in Section 5 will further validate this.

4.3. Extract patterns of traffic bottleneck

Given a user preference of road properties, we can use attributes reduction model to extract the traffic bottleneck patterns of user interests. For a decision system of traffic congestion, RAP algorithm is first used to obtain an attribute reduct of road properties. Next candidate patterns of traffic congestion are generated through integrating attribute values into the reduct. Finally, based on pattern evaluation criteria, the redundant patterns in candidate ones are filtered out. The workflow of discovering traffic bottleneck patterns is shown in Algorithm 3.

Algorithm 3. Extracting Traffic Bottleneck Patterns.

1: Preprocess the data of road conditions;
2: Formulate user preferences with the priority orders of road properties;
3: For a priority order, utilize RAP algorithm to compute an attribute reduct of road properties;
4: For every item of class ‘congestion’, integrate its property values into attribute reduct to form a candidate pattern;
5: Remove redundant patterns and filter out the candidate patterns according to their Confidences and Lifts.

Example 3. Suppose users have a preference for region environment and infrastructure setup in traffic bottleneck analysis as shown in road property order (1), the attribute reduct obtained from RAP algorithm consists of the following road properties: Traffic volume, Land type, Region, Zebra strips and Middle separator. For the items of class ‘congestion’, integrate property values to produce the candidate patterns as

(1) Traffic volume = 1 $\wedge$ Land type = 2 $\wedge$ Region = 2 $\wedge$ Zebra strips = 0 $\wedge$ Middle separator = 2 [Conf: 0.714, Lift: 2.187]
(2) Traffic volume = 1 $\wedge$ Land type = 4 $\wedge$ Region = 3 $\wedge$ Zebra strips = 0 $\wedge$ Middle separator = 1 [Conf: 1, Lift: 3.061]
(3) Traffic volume = 2 $\wedge$ Land type = 2 $\wedge$ Region = 2 $\wedge$ Zebra strips = 0 $\wedge$ Middle separator = 2 [Conf: 0.308, Lift: 0.942]
(4) Traffic volume = 2 $\wedge$ Land type = 2 $\wedge$ Region = 2 $\wedge$ Zebra strips = 1 $\wedge$ Middle separator = 0 [Conf: 1, Lift: 3.061]
(5) ......

After filtering out the candidate patterns by Confidence and Lift, finally we obtain the patterns of traffic congestion as follows.

(1) Traffic volume = 1 $\wedge$ Land type = 4 $\wedge$ Region = 3 $\wedge$ Zebra strips = 0 $\wedge$ Middle separator = 1 [Conf: 1, Lift: 3.061]
(2) Traffic volume = 2 $\wedge$ Land type = 2 $\wedge$ Region = 2 $\wedge$ Zebra strips = 1 $\wedge$ Middle separator = 0 [Conf: 1, Lift: 3.061]

The data patterns above indicate the coupling effects of separator setting and region environments on traffic congestion. They reveal that within the middle-ring city area, especially in business and residential areas, there exists a correlation between traffic congestion and deficiency of road separator setting. Users can study the dependence between congestion and other road properties through reformulating their preferences.

5. Experimental results

To analyze the patterns of traffic bottlenecks, we build up a decision system of road conditions in urban areas of Shanghai. Each record in the decision system has 14 conditional attributes (see Table 2) to present the properties of an urban road and 1 decision attribute to judge congestion. We select 300 representative road sections from thousands of ones to form the experimental data. These road sections cover 9 districts in the urban area of Shanghai, which include Huangpu, Jing’an, Changning, Hongkou, etc. 17% of these road sections locate within the inner-ring city area and 83% locate around the middle-ring. The region types of these road sections involve Entertainment (2.7%), Residential(64.3%), Business(26%) and Government(7%). The data of road conditions were collected in a period of 12 months and thus reflect the city traffic situation comprehensively.

To validate the capability of RAP for traffic data analysis, we evaluate patterns from the aspects of *accuracy*, *concision* and *user interests*. The popular measures of Confidence and Lift are used to evaluate pattern accuracy. The pattern concision is measured by the compression ratio of conditional attributes. Given a pattern P, its concision is quantified by

$$Conc(P) = |P|/|A| \tag{4}$$

in which $|P|$ and $|A|$ are respectively the attribute numbers of pattern and conditional attribute set.

Finally, a strategy is designed to measure how the discovered data patterns coincide with user interests. As introduced above, the user interests are represented by attribute preferences. Suppose users have a preference of m attributes $a_1 \prec a_2 \prec \ldots \prec a_m$, it is natural to assign attributes scores according to their positions in the attribute order. The higher priority an attribute has, the higher score it is assigned. Based on the attribute scores, we can define the degree of user interests of a pattern P as

$$Int(P) = \sum_{a_k \in P} score(a_k)/|P| \tag{5}$$

Obviously, the degree-of-interest of a pattern is the average of preference scores of all pattern attributes.

The data experiment work consists of two parts. First, we present an overall evaluation of RAP through comparing with the following popular pattern mining algorithms: Apriori [12], Predictive Apriori [29], Tertius [10] and Order-based Attribute Reduction (OAR) [32]. Table 4 lists the best 3 congestion patterns generated by different pattern mining methods. To measure the degree-of-interest of patterns, we suppose the user in this bottleneck analysis task holds a preference for road infrastructure setup and environment (see preference (1) in Section 4.2).

As shown in Table 4, RAP can extract precise congestion patterns with high Confidences and Lifts as most traditional pattern mining methods. Focusing on the cooccurrence of items, the patterns mined by traditional algorithms always consist of the attributes of road construction. They just reflect the dependence between road construction and traffic congestion. From the patterns, we find that the roads of long length and high average delay are more likely to be congested. But in a modern city, it is difficult to change these road conditions directly. Thus these patterns are not applicable to support traffic planning. Specifically, for a traffic officer who is interested in the roles of road separator setting and region environment in traffic congestion, these patterns cannot coincide with his expectation. Through integrating attribute preferences into pattern mining process, RAP patterns indicate the effects of separator setting and region environment on congestion. In the business and residential regions within the middle-ring city area, the deficiency of road separator setting easily leads to traffic block even the surrounding traffic volume is normal. Meanwhile, in the central city area of high traffic volume, the effective way to ease the traffic congestion is crowd decentralization.

Table 4
Evaluation of traffic bottleneck patterns.

Methods	Traffic bottleneck patterns	Conf	Lift	Conc (%)	Int
Apriori	Length = 3 ∧ aveDelay = 1 → Congestion	1	3.061	14.3	1.5
	Length = 3 ∧ trfVol = 2 ∧ landType = 2 → Congestion	1	3.061	21.4	6.3
	Length = 3 ∧ Region = 3 ∧ aveDelay = 1 → Congestion	1	3.061	21.4	4.3
Predictive Apriori	Length = 3 ∧ aveDelay = 1 → Congestion	1	3.061	14.3	1.5
	Length = 3 ∧ trfVol = 2 ∧ landType = 2 → Congestion	1	3.061	21.4	6.3
	Length = 3 ∧ CapDif = 3 → Congestion	1	3.061	14.3	3
Tertius	Length = 3 ∧ aveDelay = 1 → Congestion	1	3.061	14.3	1.5
	Length = 2 ∧ busLane = 0 ∧ aveDelay = 2 → Congestion	0.810	2.478	21.4	3.3
	busLane = 0 ∧ aveDelay = 2 → Congestion	0.622	1.904	14.3	4
OAR	Length = 2 ∧ laneNum = 2 ∧ CapDif = 0 ∧ jointSec = 1 ∧ busLane = 0 ∧ trfVol = 2 ∧ landType = 8 ∧ Region = 3 ∧ exitEnt = 1 ∧ zStrips = 0 ∧ carStrip = 0 → Congestion	1	3.061	78.5	7.7
	Length = 2 ∧ laneNum = 2 ∧ CapDif = 2 ∧ jointSec = 3 ∧ busLane = 1 ∧ trfVol = 3 ∧ landType = 2 ∧ Region = 3 ∧ exitEnt = 3 ∧ zStrips = 0 ∧ carStrip = 1 → Congestion	1	3.061	78.5	7.7
	Length = 3 ∧ laneNum = 2 ∧ CapDif = 2 ∧ jointSec = 3 ∧ busLane = 0 ∧ trfVol = 3 ∧ landType = 4 ∧ Region = 3 ∧ exitEnt = 1 ∧ zStrips = 0 ∧ carStrip = 1 → Congestion	1	3.061	78.5	7.7
RAP	trfVol = 1 ∧ landType = 4 ∧ Region = 3 ∧ zebStrip = 0 ∧ midSep = 1 → Congestion	1	3.061	35.7	10.4
	trfVol = 2 ∧ landType = 2 ∧ Region = 2 ∧ zebStrip = 1 ∧ midSep = 0 → Congestion	1	3.061	35.7	10.4
	trfVol = 3 ∧ landType = 4 ∧ Region = 3 ∧ zebStrip = 1 ∧ midSep = 2 → Congestion	1	3.061	35.7	10.4

The order-based reduction method selects attributes according to their priorities and guarantees high classification precision of reducts. But due to the redundant discernibility matrix and overfitting in reduction process, for real traffic data, OAR generally induces the patterns of many attributes which are difficult to capture the critical factors for analyzing congestion. The theoretical foundation of our approach is similar to the order-based method, through improving the data structure and filtering out the trivial discernibility, RAP can generate concise patterns which contain the attributes of the most user interests. It should also be noticed that no attribute in RAP pattern is redundant, because the attribute reduct obtained by RAP is independent (see Theorem 2).

To further validate the ability of RAP in multi-view data analysis, we suppose 4 users with different preferences for traffic bottleneck analysis. The references (1–4) of road properties are shown in Section 4.2. Fig. 2 illustrates the average degree-of-interest of the patterns obtained by different pattern mining methods. Obviously, without considering user interests, traditional algorithms generate the same patterns for different users thus result in low degree-of-interest. Especially for the users being interested in the road properties not occurring frequently in congestion records, such as bus lane, traditional algorithms over focus on road construction and cannot provide the satisfactory results. We also find that when a user has an attribute preference for road construction (Pref 4), the degree-of-interest of traditional patterns will be close to RAP patterns. This indicates that with specific attribute preferences, RAP can obtain the patterns similar to the traditional ones. Extracting the patterns from multiple user views, RAP actually provides a flexible tool for data analysis.

In the second experiment, we briefly present how to use RAP patterns to analyze the causes of urban traffic bottleneck and further support the traffic programming. Due to the length limit, we just present a part of experimental results here. As shown in Section 4.2, if someone wants to know the effects of road separators, RAP patterns can offer us the multiform dependencies between road separators and other kinds of road properties and their coupling effects on congestion. Table 5 shows the congestion/smooth patterns under the preferences (3, 4). From the results, we can find the following interesting rules. For the roads with multiple connections, standard middle separator setting is helpful to ease traffic congestion while zebra strips are not. The bus lane has positive effects on smoothing traffic. The long roads are more likely to be congested than the short ones. For the narrow roads, the separator setting may aggravate the traffic jam. For the wide roads with multiple lanes, middle separator setting is an effective way to smooth traffic. Synthesizing these rules, we can construct the comprehensive knowledge on road separator setting. RAP patterns facilitate traffic officers to study how to set the road separators under various road conditions. Comparing with the

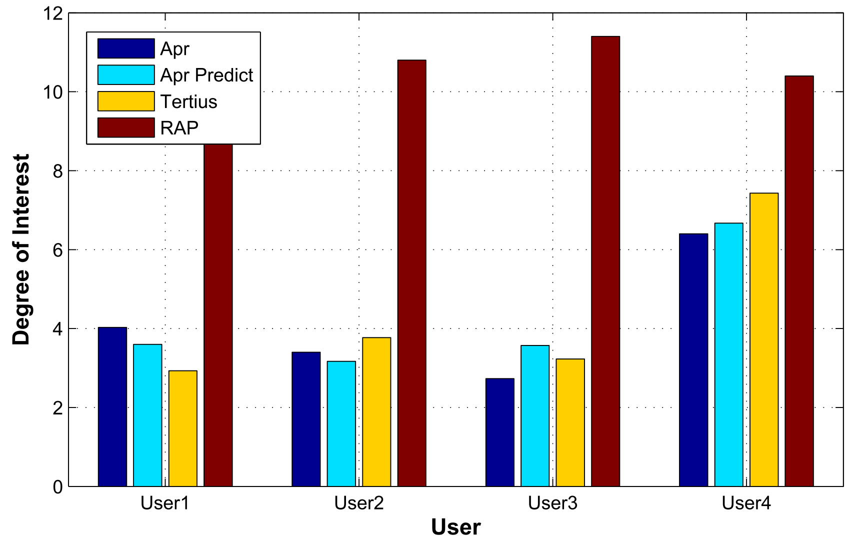

Fig. 2. Degree-of-interest with multiple views.

Table 5
Traffic bottleneck patterns with user preferences.

Preferences	Congestion/Smooth patterns
Pref 3	busLane = 0 ∧ jointRoads = 3 ∧ zebStrip = 1 ∧ midSep = 0 ∧ carSep = 1 → Congestion
	busLane = 0 ∧ jointRoads = 3 ∧ zebStrip = 1 ∧ midSep = 0 ∧ carSep = 2 → Congestion
	busLane = 0 ∧ jointRoads = 2 ∧ zebStrip = 1 ∧ midSep = 1 ∧ carSep = 1 → Congestion
	busLane = 1 ∧ jointRoads = 1 ∧ zebStrip = 0 ∧ midSep = 2 ∧ carSep = 1 → Smooth
	busLane = 1 ∧ jointRoads = 2 ∧ zebStrip = 0 ∧ midSep = 1 ∧ carSep = 2 → Smooth
	busLane = 1 ∧ jointRoads = 2 ∧ zebStrip = 0 ∧ midSep = 2 ∧ carSep = 1 → Smooth
Pref 4	aveDelay = 2 ∧ Length = 2 ∧ laneNum = 1 ∧ zebStrip = 1 ∧ midSep = 1 → Congestion
	aveDelay = 1 ∧ Length = 3 ∧ laneNum = 1 ∧ zebStrip = 1 ∧ midSep = 0 → Congestion
	aveDelay = 1 ∧ Length = 3 ∧ laneNum = 3 ∧ zebStrip = 1 ∧ midSep = 0 → Congestion
	aveDelay = 2 ∧ Length = 1 ∧ laneNum = 2 ∧ zebStrip = 0 ∧ midSep = 2 → Smooth
	aveDelay = 2 ∧ Length = 1 ∧ laneNum = 3 ∧ zebStrip = 0 ∧ midSep = 2 → Smooth
	aveDelay = 2 ∧ Length = 3 ∧ laneNum = 2 ∧ zebStrip = 1 ∧ midSep = 2 → Smooth

traditional pattern mining algorithms, RAP is more actionable and consistent with the real application requirements.

6. Conclusions

To bridge the gap between the diverse user expectations in traffic bottleneck analysis and pattern mining algorithms, we propose a multi-view attribute reduction model and apply the model to extract congestion patterns according to user interests. In the reduction model, user views are expressed by attribute preferences, which are formally represented by attribute orders. We validate our approach based on the reports of road conditions from Shanghai. Experimental results show that the proposed model is effective in analyzing congestion causes from the views of traffic management. Our future work will focus on the following issues. First, the efficiency of the proposed model need to be further demonstrated in both theory and data experiments. Second, Confidence and Lift may not be the best measure to filter out the candidate patterns. Finally, the strategy of integrating the RAP patterns into knowledge should be formally designed.

Acknowledgments

This work was supported by Innovation Foundation of Shanghai University, National Natural Science Foundation of China (Serial Nos. 61103067, 61103070, 61170155, 61273304), Young Teachers Program of Universities in Shanghai and The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University.

References

[1] M. Ankerst, Report on the SIGKDD-2002 panel the perfect data mining tool: interactive or automated?, ACM SIGKDD Explor Newslett. 4 (2) (2002) 110–111.
[2] Y.L. Bai, Z.R. Wu, S.Y. Sun, C.N. Wang, Automatic identification algorithm for freeway bottleneck, in: Proceedings of International Conference on Transportation, Mechanical, and Electrical Engineering, 2011, pp. 1857–1860.
[3] S. Barai, Data mining applications in transportation engineering, Transport 18 (5) (2003) 216–223.
[4] L. Cao, P.S. Yu, C. Zhang, Y. Zhao, Domain Driven Data Mining, Springer Science+Business Media, New York, USA, 2009.
[5] L. Cao, H. Zhang, Y. Zhao, D. Luo, C. Zhang, Combined mining: discovering informative knowledge in complex data, IEEE Trans. SMC Part B 41 (3) (2011) 699–712.
[6] R. Chan, Q. Yang, Y. Shen, Mining high utility itemsets, in: Proceedings of the Third IEEE International Conference on Data Mining (ICDM03), 2003, pp. 19–26.
[7] S.Y. Chen, W. Wang, H.V. Zuylen, A comparison of outlier detection algorithms for ITS data, Expert Syst. Appl. 37 (2010) 1169–1178.
[8] Y. Chen, Y.Y. Yao, Multiview intelligent data analysis based on granular computing, in: Proceedings of 2006 IEEE International Conference on Granular Computing, 2006, pp. 281–286.
[9] U. Fayyad, G. Piatetsky-Shapiro, R. Uthurusamy, Summary from the KDD-03 panel: data mining: the next 10 years, ACM SIGKDD Explor. Newslett. 5 (2) (2003) 191–196.
[10] P.A. Flach, N. Lachiche, Confirmation-guided discovery of first-order rules with tertius, Mach. Learn. 42 (1999) 61–95.
[11] X. Gong, X. Liu, A data-mining-based algorithm for traffic network flow forecasting, in: Proceedings of 2003 IEEE Intelligent Transportation Systems, 2003, pp. 193–198.
[12] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, third ed., Morgan Kaufman Publishers, San Francisco, CA, USA, 2011.
[13] S. Han, J. Wang, The second attribute, in: RSFDGrc05, Lecture Notes in Computer Science, vol. 3641, 2005, pp. 156–165.
[14] X.F. Ji, W. Cheng, Y.G. Chen, Traffic state identification methods based on vague sets, in: Proceedings of International Conference on Computational Intelligence and Software Engineering, 2009, pp. 11–14.
[15] B.S. Kerner, Control of spatiotemporal congested traffic patterns at highway bottlenecks, IEEE Trans. Intell. Transport. Syst. 8 (2) (2007) 308–320.

[16] W.H. Lee, S.S. Tseng, J.L. Shieh, H.H. Chen, Discovering traffic bottlenecks in an urban network by spatiotemporal data mining on location-based services, IEEE Trans. Intell. Transport. Syst. 12 (4) (2011) 1047–1056.
[17] W.H. Lee, S.S. Tseng, S.H. Tsai, A knowledge based real-time travel time prediction system for urban network, Expert Syst. Appl. 36 (2009) 4239–4247.
[18] T.Y. Lin, Y.Y. Yao, L.A. Zadeh, Data Mining, Rough Sets and Granular Computing, Physica-Verlag, Heidelberg, Germany, 2002.
[19] B. Liu, W. Hsu, S. Chen, Y. Ma, Analyzing subjective interestingness of association rules, IEEE Intell. Syst. 15 (5) (2000) 47–55.
[20] S. Ma, Y. Zheng, O. Wolfson, T-share: a large scale dynamic taxi ridesharing service, in: Proceedings of IEEE International Conference on Data Engineering (ICDE13), 2013, pp. 410–421.
[21] X.A. Ma, G.Y. Wang, H. Yu, T.R. Li, Decision region distribution preservation reduction in decision-theoretic rough set model, Inf. Sci. 278 (2014) 614–640.
[22] D.Q. Miao, Y. Zhao, Y.Y. Yao, Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model, Inf. Sci. 179 (2009) 4140–4150.
[23] E. Omiecinski, Alternative interest measures for mining associations in databases, IEEE Trans. Knowl. Data Eng. 15 (1) (2003) 57–69.
[24] L.X. Pang, S. Chawla, W. Liu, Y. Zheng, On mining anomalous patterns in road traffic streams, in: Proceedings of the 7th International Conference on Advanced Data Mining and Applications, 2011, pp. 237–251.
[25] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, Boston, USA, 1991.
[26] J. Qian, P. Lv, X.D. Yue, C.H. Liu, Z.J. Jing, Hierarchical attribute reduction algorithms for big data using MapReduce, Knowl.-Based Syst. 73 (2015) 18–31.
[27] Y.H. Qian, J.Y. Liang, W. Pedrycz, C.Y. Dang, Positive approximation: an accelerator for attribute reduction in rough set theory, Artif. Intell. 174 (9-10) (2010) 597–618.
[28] J.T. Ren, Y. Zhang, G.H. Zhang, C.G. Zong, Frequent simultaneously congested link-sets discovery and ranking, in: Proceedings of 2003 IEEE Intelligent Transportation Systems, 2003, pp. 624–627.
[29] T. Scheffer, Finding association rules that trade support optimally against confidence, in: Proceedings of 5th European Conf. on Principles of Data Mining and Knowledge Discovery, 2001, pp. 424–435.
[30] W.H. Shu, W.B. Qian, A fast approach to attribute reduction from perspective of attribute measures in incomplete decision systems, Knowl.-Based Syst. 72 (2014) 60–71.
[31] P. Tan, V. Kumar, J. Srivastava, Selecting the right interestingness measure for association patterns, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD02), 2002, pp. 32–41.
[32] J. Wang, J. Wang, Reduction algorithms based on discernibility matrix: the ordered attributes method, J. Comput. Sci. Technol. 16 (6) (2001) 489–504.
[33] K. Wang, Y. Jiang, A. Tuzhilin, Mining actionable patterns by role models, in: Proceedings of the 22nd International Conference on Data Engineering (ICDE06), 2006, pp. 16.
[34] R. Wong, A. Fu, K. Wang, Mpis: Maximal-profit item selection with cross-selling considerations, in: Proceedings of the Third IEEE International Conference on Data Mining (ICDM03), 2003, pp. 371–378.
[35] Q. Yang, J. Yin, C. Ling, R. Pan, Extracting actionable knowledge from decision trees, IEEE Trans. Knowl. Data Eng. 19 (1) (2007) 43–56.
[36] Y.Y. Yang, D.G. Chen, Z. Dong, Novel algorithms of attribute reduction with variable precision rough set model, Neurocomputing 139 (2014) 336–344.
[37] Y.Y. Yao, Y. Zhao, Discernibility matrix simplification for constructing attribute reducts, Inf. Sci. 179 (7) (2009) 867–882.
[38] Y.Y. Yao, Y. Zhao, J. Wang, S. Han, A model of user-oriented reduct construction for machine learning, Trans. Rough Sets VIII 5084 (2008) 332–351.
[39] M.Q. Ye, X.D. Wu, X.G. Hu, D.H. Hu, Knowledge reduction for decision tables with attribute value taxonomies, Knowl.-Based Syst. 56 (2014) 68–78.
[40] S.C. Yoon, L.J. Henschen, E.K. Park, S. Makki, Using domain knowledge in knowledge discovery, in: Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM99), 1999, pp. 243–250.
[41] J. Yuan, Y. Zheng, and X. Xie. Discovering regions of different functions in a city using human mobility and POIs, in: Proceedings of 18th SIGKDD conference on Knowledge Discovery and Data Mining (KDD12), 2012, pp. 186–194.
[42] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu, C. Chen, Data-driven intelligent transportation systems: a survey, IEEE Trans. Intell. Transport. Syst. 12 (4) (2011) 1624–1639.
[43] K. Zhao, J. Wang, A reduction algorithm meeting users requirements, J. Comput. Sci. Technol. 17 (5) (2002) 578–593.