Application of Interactive Fuzzy Data Mining to the Analysis of Inter-Vehicle Communication in Traffic Simulations

Yusuke NOJIMA*, Yoshihiro HAMADA* and Hisao ISHIBUCHI*

*Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
nojima@cs.osakafu-u.ac.jp
hamada@ci.cs.osakafu-u.ac.jp
hisaoi@cs.osakafu-u.ac.jp

Abstract: Recently, Inter-Vehicle Communication (IVC) has actively been studied to avoid traffic congestion. Vehicles with IVC can obtain the latest traffic information on the congestion through local communication among neighboring vehicles. In this paper, we examine the effectiveness of IVC through computer simulations. Each vehicle chooses a link (i.e., route) according to the traffic information. The traffic information for each link is represented by a link weight. Each link weight corresponds to the travel time that a vehicle can pass the link. We examine when and where the traffic information is useful for link selections. Thus, we collect travel records (e.g., predicted and actual travel times, travel path, traffic volumes around the path) of vehicles with IVC from traffic simulations. Since we can obtain a huge amount of data, we analyze the data by fuzzy data mining. It can linguistically represent the data by fuzzy rules. We use multiobjective genetic fuzzy rule selection to obtain a set of fuzzy rules with high accuracy and interpretability. For the interpretability, we incorporate user’s preference into our genetic rule selection method. User’s preference is represented by satisfaction levels. The satisfaction levels are interactively modified by its user in our interactive fuzzy data mining. Key words: Inter-vehicle communication, traffic simulation, interactive fuzzy data mining, multiobjective genetic fuzzy rule selection, fuzzy rule-based knowledge extraction.

INTRODUCTION

Vehicles are widely used as a means of useful transportation in the mobility society where the demand for road traffic is expanding year by year. At the same time, chronic traffic congestion has become a big social problem. One solution is advanced transportation information systems (ATIS) such as VICS (see http://www.vics.or.jp/english/). A VICS center gathers road traffic information about congestion and regulations. After processing information, the center broadcasts information. This centralized system needs public infrastructures and time to edit information. On the other hand, local communication systems named Inter-Vehicle Communication (IVC) have been recently proposed and discussed [KAT 03] [JIN 06] [SCH 06] [SHI 06] [OHA 07]. IVC has several advantages: no need of huge public infrastructure investment and little time lag on transmitting traffic information. This is because vehicles can directly communicate traffic information to each other.

In this paper, we examine the effectiveness of IVC through computer simulations. In our simulator, each vehicle chooses a link (i.e., route) according to the traffic information. The traffic information for each link is represented by a link weight. In our simulator, each link weight corresponds to the travel time that a vehicle can pass the link. That is, each vehicle predicts the travel time using the traffic information. We examine when and where the traffic information is useful for link selections. To do that, we collect various kinds of travel records (e.g., predicted and actual travel time, travel path, traffic volumes around the path) of vehicles with IVC from traffic simulations. Since we can obtain a huge amount of data, we analyze the data by fuzzy data mining.

There are a lot of data mining techniques in the literature [WIT 05]. The interpretability of extracted knowledge is an important issue in data mining as well as its accuracy. In knowledge extraction from numerical data, a fuzzy rule-based classifier is a promising knowledge representation form. This is because the antecedent part of a fuzzy rule is linguistically designed with membership functions. Interactive fuzzy data mining has been proposed for the design of fuzzy
rule-based classifiers [NOJ 09a]. Its modification is proposed in [NOJ 09b]. In those papers, it was applied to the benchmark problems available from UCI machine learning repository. It has two main characteristics. One is to use evolutionary multiobjective optimization (EMO) algorithms. The other is to consider user’s preference on the interpretability of fuzzy rule-based classifiers.

In the design of fuzzy rule-based classifiers, evolutionary algorithms have often been used for parameter tuning, input selection, rule generation, and rule selection under the name of genetic fuzzy systems (GFSs) [COR 01] [COR 04] [HER 05]. Since the late 1990s, the importance of interpretability maintenance in the design of fuzzy rule-based systems has been pointed out in the literature (e.g., [ISH 95]). Evolutionary multiobjective optimization (EMO) algorithms have been widely used in the problems with a number of conflicting objectives [DEB 01] [COE 02] [COE 04]. EMO algorithms can obtain a number of nondominated solutions in terms of the conflicting objectives by its single run. That is, we can obtain a number of fuzzy rule-based classifiers with different accuracy and interpretability. A multiobjective genetic fuzzy system (MGFS) is a growing topic in GFSs (e.g., see an MGFS bibliography main tained by M. Cococcioni http://www2.ing.unipi.it/~o613499/emofrbss.html).

There are a large number of papers on the interpretability issue of fuzzy rule-based systems [ABO 03] [CAS 03] [CAS 05] [GOM 07] [JIM 01] [JIN 00] [NAU 99] [SET 98] [SET 00] [YEN 98]. There exist various interpretability measures because the interpretability of fuzzy rule-based systems is very subjective for its user. That is, it is hard to specify an interpretability measure without its user’s opinion. Furthermore, the user also doesn’t know how to specify the interpretability measure before obtaining some knowledge. Thus the interpretability measure should be interactively modified by the user during the search process. For this type of problems, interactive evolutionary computation (IEC) [CHO 02] [TAK 01] seems to be a promising approach. IEC is useful or necessary to incorporate user’s preference into evolutionary optimization.

In this paper, we examine the relationship between the situation that each vehicle faces and the validity of IVC. From a huge amount of collected data, we extract a large number of fuzzy rules. The antecedent parts and consequent parts of fuzzy rules represent the road conditions and the validation of IVC, respectively. Then we apply multiobjective genetic fuzzy rule selection [ISH 97] [ISH 04] to obtain a number of accurate and interpretable fuzzy rule-based classifiers. Our interactive fuzzy data mining is based on IEC and EMO like [BRI 06] [BRI 08]. We use the number of correctly classified training patterns and the number of fuzzy rules as quantitative objective functions. In addition, as another qualitative objective function, we consider user’s preference. In our method, user’s preference is represented by six satisfaction level functions related to the interpretability and accuracy. In our method, these three objective functions are optimized by an EMO algorithm. During the evolution, the satisfaction level functions are interactively updated by the user with our user evaluation interface.

This paper is organized as follows. First we explain our traffic simulator in Section 2. Next we explain multiobjective genetic rule selection and the incorporation of user’s preference in Section 3. In Section 4, we demonstrate a case study and examine the effectiveness of IVC. Finally Section 5 concludes this paper.

1. Traffic Simulator

Traffic flow models can be divided into macroscopic and microscopic models. In macroscopic models, a traffic flow is treated as a phenomenon based on fluid dynamics [LIG 55] [TRE 99]. On the other hand, a traffic flow is treated as the interaction among vehicles in microscopic models [DIA 02] [MAE 92] [SCH 95]. Yikai et al. proposed a traffic flow model based on fuzzy estimation of each vehicle’s behavior [YIK 93a] [YIK 93b]. Tamaki et al. proposed a traffic flow model [TAM 03] [TAM 04] using cellular automata where a stochastic velocity model [NAG 92] was utilized. These studies are examples of microscopic models.

In this paper, we develop a microscopic traffic flow model using cellular automata as in the above mentioned papers. A cellular automaton is a discrete model which has been studied in computability theory, mathematics, and theoretical biology. It is based on a regular grid of cells. Each cell assumes one of a finite number of states. Discrete time is usually used. The state of a cell at time $t+1$ is a function of the states of a finite number of neighboring cells at time $t$. In this section, we explain our traffic flow model based on cellular automata. We also explain our IVC method examined in this paper.

1.1. Outline

We first explain the outline of our traffic flow model. Figure 1 shows the road map of our traffic flow model. The simulation area is divided into squared cells. In our traffic model, we assume that the road map is treated as a directed graph where a node and a link correspond to an intersection and a road between intersections, respectively. An Intersection is assigned randomly to any cell on any link in Fig. 1.

1.2. Transition Rules

In this subsection, we explain transition rules in our traffic flow model. Our model is a deterministic one, which follows the Wolfram’s rule 184 (CA-184)
[WOL 86] except for the intersection. The state of each cell is empty or occupied by a vehicle. The positions of all vehicles running in the model are updated synchronously. At every state transition time, each vehicle stays at the current cell or jumps to its next cell. Our local transition rule is simply stated as “a vehicle moves only when its next cell towards its destination is empty”. This means that the drivers are short-sighted. That is, they do not know whether the vehicle in front can move or is stuck by another vehicle. Therefore, the state of each cell $s_i$ is entirely determined by the occupancy of the cell itself and its two nearest neighbors $s_{i-1}$ and $s_{i+1}$ along the route. Figure 2 summarizes our local transition rule for all the eight possible configurations $(s_{i-1}, s_i, s_{i+1}) \rightarrow (s_i)_{t+1}$. Empty and occupied cells are shown by white and black squares, respectively. In Fig. 2, the state $(s_i)_{t+1}$ of the center cell at the next time step $t+1$ is specified based on the states $s_{i-1}, s_i$ and $s_{i+1}$ at the current time step $t$. All vehicles are assumed to move right in Fig. 2.

![Figure 1. Road map of our traffic model.](image)

![Figure 2. Illustration of our local transition rule for the state $s_i$ of the $i$-th cell under the motion rule of CA-184 (all vehicles are assumed to move right).](image)

1.3. Inter-Vehicle Communication (IVC)

In this subsection, we explain a route selection method based on available information for vehicles through IVC. Our method chooses a route from its origin to its destination based on available information, and revises the selected route whenever the vehicle approaches an intersection. In this paper, we represent the traffic information for each link by a link weight. Each link weight corresponds to the travel time that the vehicle can pass the link. Thus, if a link weight is large, a vehicle on the link needs long travel time to pass the link. We use Dijkstra’s algorithm [DIJ 59] to search for the route with the minimal sum of link weights (i.e., the fastest route).

Each vehicle has its own weight for each link. The actual travel time of the vehicle is assigned as the weight to the corresponding link. Figure 3 shows an example in which a vehicle $A$ passes on another vehicle $B$ on the opposite lane. They can communicate with each other through IVC. Traffic information to be shared by these two vehicles consists of the travel time (i.e., weight) and the update time for each link. It should be noted that each vehicle has its own travel time and update time for each link. More specifically, the newer information for each link is shared by these two vehicles by updating the older one for each link. Closely adjacent vehicles in the same lane also communicate directly with each other in the same manner as in the above-mentioned situation. We assume that the communication range of each vehicle is one cell. That is, a vehicle $C$ can directly communicate with a vehicle $E$ in Fig. 4. Besides, the vehicle $C$ can indirectly communicate with a vehicle $F$ through vehicles $E$ and $D$ in Fig. 4.

![Figure 3. An example of inter-vehicle communication.](image)

![Figure 4. An example of inter-vehicle communication.](image)
2. Multiobjective Fuzzy Rule Selection

In this section, we explain fuzzy rule-based classifiers and multiobjective genetic fuzzy rule selection. We also explain user’s preference and its incorporation into multiobjective genetic fuzzy rule selection.

2.1. Fuzzy Rule-based Classifiers

Let us assume that we have \( m \) training (i.e., labeled) patterns \( x_p = (x_{p1}, \ldots, x_{pn}), p = 1, 2, \ldots, m \) from \( M \) classes in an \( n \)-dimensional pattern space where \( x_{pn} \) is the attribute value of the \( p \)th pattern for the \( i \)th attribute \( (i = 1, 2, \ldots, n) \). For the simplicity of explanation, we assume that all the attribute values have already been normalized into real numbers in the unit interval \([0, 1]\). Thus the pattern space of our classification problem is an \( n \)-dimensional unit-hypercube \([0, 1]^n\).

For our \( n \)-dimensional pattern classification problem, we use fuzzy rules of the following type:

Rule \( R_q \): If \( x_1 \) is \( A_{q1} \) and \ldots and \( x_n \) is \( A_{qn} \) then Class \( C_q \) with \( CF_q \).

where \( R_q \) is the label of the \( q \)th fuzzy rule, \( x = (x_1, \ldots, x_n) \) is an \( n \)-dimensional pattern vector, \( A_{qi} \) is an antecedent fuzzy set (with \( i = 1, 2, \ldots, n \) and \( C_q \) is a class label, and \( CF_q \) is a rule weight. We denote the antecedent fuzzy sets of \( R_q \) as a fuzzy vector \( A_q = (A_{q1}, A_{q2}, \ldots, A_{qn}) \).

We use 14 fuzzy sets in four fuzzy partitions with different granularities in Fig. 5. In addition to 14 fuzzy sets, we also use the domain interval \([0, 1]\) as an antecedent fuzzy set in order to represent a don't care condition.

![Figure 5. Membership functions used in this paper.](image)

The consequent class \( C_q \) and the rule weight \( CF_q \) of each fuzzy rule \( R_q \) are specified from training patterns compatible with its antecedent part \( A_q = (A_{q1}, A_{q2}, \ldots, A_{qn}) \) in the following heuristic manner. First we calculate the confidence of each class for the antecedent part \( A_q \) as \([\text{HON } 01]\):

\[
c(A_q \Rightarrow \text{Class } h) = \frac{\sum_{p=1}^{m} \mu_{A_q}(x_p)}{\sum_{p=1}^{m} \mu_{A_q}(x_p)},
\]

\[ h = 1, 2, \ldots, M, \quad (2) \]

where \( \mu_{A_q}(x_p) \) is the compatibility grade of \( x_p \) with the antecedent part \( A_q \) of each fuzzy rule \( R_q \). The rule weight \( CF_q \) is calculated by the product operation as:

\[
\mu_{A_q}(x_p) = \mu_{A_{q1}}(x_{p1}) \cdot \ldots \cdot \mu_{A_{qn}}(x_{pn}). \quad (3)
\]

Then the consequent class \( C_q \) is specified by identifying the class with the maximum confidence:

\[
c(A_q \Rightarrow \text{Class } C_q) = \max_{h=1,2,\ldots,M} [c(A_q \Rightarrow \text{Class } h)]. \quad (4)
\]

In this manner, we generate the fuzzy rule \( R_q \) with the antecedent part \( A_q \) and the consequent class \( C_q \). We do not generate any fuzzy rules with the antecedent part \( A_q \) if there is no compatible training pattern with \( A_q \).

The rule weight \( CF_q \) of each fuzzy rule \( R_q \) is specified by the confidence values \([\text{ISH } 05a]\):

\[
CF_q = c(A_q \Rightarrow \text{Class } C_q) - \sum_{h=1, h \neq C_q}^{M} c(A_q \Rightarrow \text{Class } h). \quad (5)
\]

We do not use the fuzzy rule \( R_q \) as a candidate rule if the rule weight \( CF_q \) is not positive (i.e., if its confidence is not larger than 0.5).

As confidence, support is also often used for evaluating the interestingness of individual rules. Support can be calculated as follows:

\[
s(A_q \Rightarrow \text{Class } C_q) = \frac{\sum_{p=1}^{m} \mu_{A_q}(x_p)}{m}. \quad (6)
\]

Let \( S \) be a set of the fuzzy rules of the form in (1). When an input pattern \( x_p \) is to be classified by \( S \), first we calculate the compatibility grade of \( x_p \) with the antecedent part \( A_q \) of each fuzzy rule \( R_q \) in \( S \) using the product operation in (3). Then a single winner rule is identified using the compatibility grade and the rule weight of each fuzzy rule as:
\[ \mu_A(x_p) \cdot CF_w = \max \{ \mu_A(x_p) \cdot CF_q \mid R_q \in S \}. \quad (7) \]

The input pattern \( x_p \) is classified as the consequent class of the winner rule \( R_w \). When multiple fuzzy rules with different consequent classes have the same maximum value in (7), the classification of \( x_p \) is rejected. If there is no compatible fuzzy rule with \( x_p \), its classification is also rejected. For more detail, see [ISH 05b].

### 2.2. Multiobjective Genetic Fuzzy Rule Selection

Multiobjective genetic fuzzy rule selection is a two-step method [ISH 97] [ISH 04]. In the first step, a prespecified number of promising fuzzy rules are generated from training patterns as candidate rules. In the second step, an EMO algorithm is used to search for non-dominated fuzzy rule-based classifiers (i.e., non-dominated subsets of the generated candidate rules in the first step).

Since we use the 14 antecedent fuzzy sets in Fig. 5 and a don’t care for each attribute of our n-dimensional classification problem, the total number of possible fuzzy rules is \( 15^n \). Among these possible rules, we examine only short fuzzy rules with a small number of antecedent conditions (i.e., short fuzzy rules with many don’t care conditions) to generate candidate rules. In this paper, we examine fuzzy rules with three or less antecedent conditions. For prescreening candidate rules, we use the product of the support \( s(R_q) \) and the confidence \( c(R_q) \). That is, we choose a prespecified number of the best candidate rules for each class with respect to \( s(R_q) \cdot c(R_q) \).

Let us assume that we have \( N \) candidate rules (i.e., \( N/M \) candidate rules for each of \( M \) classes). Any subset \( S \) of the \( N \) candidate rules can be represented by a binary string of length \( N \): \( S = s_1 s_2 \ldots s_N \) where \( s_j = 1 \) and \( s_j = 0 \) mean the inclusion and the exclusion of the \( j \)th candidate rule \( R_j \) in the subset \( S \), respectively (\( j = 1, 2, \ldots, N \)). Such a binary string \( S \) is used as an individual (i.e., a fuzzy rule-based classifier) in an EMO algorithm of our method.

Each fuzzy rule-based classifier \( S \) is evaluated by the following three objectives:

- \( f_1(S) \): the number of correctly classified training patterns by \( S \),
- \( f_2(S) \): the number of selected fuzzy rules in \( S \),
- \( f_3(S) \): overall user preference for \( S \).

The first and second objectives have been frequently used and correspond to accuracy maximization and complexity minimization, respectively. The third objective \( f_3 \) is the newly proposed objective in this paper. We explain it in the next subsection.

The problem formulation of multiobjective genetic fuzzy rule selection is written as

Maximize \( f_1(S) \) and \( f_2(S) \), and minimize \( f_3(S) \). \quad (8)

We use NSGA-II of Deb et al. [DEB 02] to search for a number of non-dominated fuzzy rule-based classifiers with respect to these three objectives. In this paper, uniform crossover and bit-flip mutation are used in NSGA-II. We also use three problem-specific operations in this paper. One is that the unnecessary fuzzy rules which are not selected as a winner rule are removed from \( S \) after calculating the first objective. Another is that we assign the worst rank to the fuzzy rule-based classifiers with less than two rules, because these fuzzy rule-based classifiers give us less information. The other is that we remove overlapping solutions in the decision space from offspring population [ISH 05c] [NAR 05].

### 2.3. User Preference

As we mentioned in Section 1, interpretability is so subjective and hardly specified without the actual user’s opinion. One approach may be to use various interpretability measures as objective functions and try to obtain a number of non-dominated classifiers in terms of many-objectives. But current EMO algorithms are not appropriate for the problems with more than four objectives [HUG 05] [ISH 08] [JAS 04] [KHA 03] [PUR 03]. Another approach may be to implicitly build a user evaluation model through actual user evaluations by using neural networks or similar models. But the model will be a black box and hardly understandable. For these reasons, we combine several satisfaction level functions into an overall preference function. The satisfaction level functions represent the preferred level and the priority of criteria related to user’s preference on a fuzzy rule-based classifier. Each satisfaction level function is interactively modified by the user during the evolution in our genetic rule selection.

In this paper, we use four interpretability criteria for representing user’s preference: average confidence, average coverage, the number of used attributes, and the maximum number of used granularities. Confidence and support have been often used to examine the interestingness of individual rules [BAY 99]. In [NOJ 09a], we used confidence and support, but there is sometimes a big gap between the support values of majority and minority classes. For this problem, we use coverage instead of support in this paper. The coverage is calculated as almost the same form of support as:

\[ \text{Cover}(A_q \Rightarrow \text{Class } C_q) = \frac{\sum \mu_{A_q}(x_p)}{\text{sup}(C_q)}, \quad (9) \]

where \( \text{sup}(C_q) \) is the consequent support, which is equal to the number of patterns whose class is \( C_q \). This measure has the same property as support. This is often used for partial classification [ALI 97] [IGL 06]. Average confidence and coverage values are calculated over fuzzy if-then rules in \( S \).
Figure 5 shows an example of the antecedent parts of three rules in a fuzzy rule-based classifier. The closed triangle represents one of the membership functions in Fig. 4. Open rectangles represent don’t care conditions. The number of used attributes in the classifier in Fig. 2 is five, because at least one fuzzy set is assigned to \( x_2, x_4, x_5, x_6, \) and \( x_8 \). The maximum number of used granularities of the classifier in Fig. 5 is two, because two granularities (i.e., two kinds of partitions) are used for the second and sixth attributes. Some researchers have pointed out that this similarity among different granularities is not appropriate for knowledge from the interpretability point of view.

In addition to the above four interpretability criteria, we also use two quantitative criteria: \( f_1 \) and \( f_2 \) for user-specified requirement levels.

We define the third objective function for an overall user preference as:

\[
f_3(S) = U(u_1(g_1(S)), u_2(g_2(S)), u_3(g_3(S)), u_4(g_4(S)), u_5(f_1(S)), u_6(f_2(S))),
\]

where \( u_i(S) \) is the satisfaction level function on the \( i \)th criterion of user’s preference. \( g_1, g_2, g_3 \) and \( g_4 \) mean average confidence, average coverage, the number of used attributes, and the maximum number of used granularities, respectively. In this paper, the overall user preference \( f_3 \) is calculated by the simple sum of the values of the satisfaction level functions. Of course, we can use other criteria in the preference function.

Each satisfaction level function on a criterion is represented by a membership function in Fig. 7. In [NOJ 09a], we used triangular-type member functions, but a trapezoidal-type membership function seems to be more suitable for representing a requirement level on a criterion. The function (a) in Fig. 7 is used for \( g_1, g_2, \) and \( f_1 \) (i.e., average confidence, average coverage, and the number of correctly classified training patterns). The function (b) in Fig. 7 is used for \( g_3, g_4, \) and \( f_2 \) (i.e., the number of used attributes, the maximum number of used granularities, and the number of rules).

Each function in Fig. 7 is specified by two segments. A user can modify the position of the point \( B(u_a, u_b) \) in \( g_{\text{min}} \leq u_a \leq g_{\text{max}} \) and \( 0 \leq u_b \leq 1 \) during the evolution. That is, the user can specify the search direction according to his/her own preference on fuzzy rule-based classifiers.

2.4. Procedure of the Proposed Method

Figure 8 shows the whole procedure of the proposed method. We specify an interval (i.e., the number of generations) for internal evaluations. During this interval, the satisfaction level functions are not changed. After the interval, the user checks some of non-dominated classifiers and modifies the satisfaction level functions in the preference function. Then another internal evaluation process starts. By repeating this interactive process, the user can specify his/her preference and find the classifier with the high user preference value.
2.5. Interactive User Interface

We developed a user interface for presenting a fuzzy rule-based classifier to the user and incorporating his/her preference (Fig. 9). The antecedent part of each fuzzy rule is shown together with its consequent class, confidence, and support. Closed triangles and open rectangles mean membership functions and don't care conditions, respectively. The accuracy of the classifier is shown at the right-bottom of the classifier.

The bottom gray zone of the interface is a user manipulation area. The user can simply specify the position of the point B in Fig. 7 by clicking in the ranges on each criterion in Fig. 9.

Each vertical dashed line in each graph for a satisfaction level function in Fig. 9 represents the actual values of the corresponding criterion for the displayed classifier. Thus, the user can refer to this information and modify the position of the point B of the trapezoidal functions in Fig. 7. That is, the user can specify the preference function according to their impression from some displayed fuzzy rule-based classifiers.

There are three buttons at the right-bottom corner in the interface in Fig. 9. The button “Best” is to show the best classifier in terms of the overall user preference. The button “Alt” is to show five alternative classifiers randomly selected among non-dominated ones. The button “Evolve” is to start another internal evaluation process with a prespecified number of generations.

![Figure 9. A user interface for the proposed method.](image)

3. Computer Simulations

3.1. Data Preparation

In this subsection, we explain how to prepare training data with class labels from the travel records in our traffic simulator. There exist 300 vehicles in our simulation environment in Fig. 1. The termination condition of traffic simulations was that each vehicle reached the goals at least 50 times. Each vehicle can communicate with another vehicle in the eight neighborhood cells.

Let us assume that a vehicle travels from node A to B and then chooses a route from node B to C in Fig. 10. In this case, a training pattern \( x = (x_1, \ldots, x_{10}) \) is collected at the node B. In the following, each element of this training pattern is explained in detail.

The first three elements \( x_1, x_2, \) and \( x_3 \) are link connection density of node A, B, and C, respectively. Link connection density is the sum of the number of links that the neighboring nodes have. For example, \( x_2 \) is link connection density of node B, which is a sum of the number of links that the neighboring nodes (i.e., A, C, D, and E) have. That is, \( x_2 = 13 \) (i.e., \( 4+3+3+3 \)). When the link connection density of a node is high, the node can be viewed as the hub of the neighboring nodes. That is, a large number of vehicles must be likely to pass the node.

The fourth element \( x_4 \) is the traffic volume in the current lane (i.e., A to B), \( x_5 \) is the traffic volume in the current opposite lane (i.e., B to A), \( x_6 \) is the traffic volume in the next lane (i.e., B to C), and \( x_7 \) is the traffic volume in the next opposite lane (i.e., C to B). A large traffic volume of a link means heavy traffic where each vehicle can communicate with each other very often. It also suggests possible traffic congestion.

The other elements \( x_8, x_9, x_{10} \) are the number of links of nodes A, B, and C, respectively. They are related to the traffic volume and the frequency of communication.

![Figure 10. An example of link connection density.](image)

Next we explain how to define the class label of each training pattern, which shows the effectiveness of IVC. We focus on the accuracy of the predicted travel time from available information. We use the following two class labels:

**Class 1:** The vehicle could not correctly predict the travel time.

**Class 2:** The vehicle could correctly predict the travel time.

If the difference between the predicted travel time and the actual travel time is less than three steps, the pattern is regarded as Class 2. Otherwise Class 1. To rebalance the number of patterns for each class, we randomly extracted 500 patterns for each class from the collected patterns.
3.2. Experimental Results

First we generated 500 fuzzy rules for each class from the extracted patterns. Those fuzzy rules were used as candidate rules in multiobjective genetic fuzzy rule selection where NSGA-II with the population size 200 was executed for 1000 generations. Every 100 generation, the interactive user interface showed a fuzzy rule-based classifier with the highest user preference, and five non-dominated fuzzy rule-based classifiers in terms of three objectives. We assume that a user prefers simple fuzzy rule-based classifiers.

Figure 11 shows the simplest fuzzy rule-based classifier and the most accurate one among non-dominated ones at 100th generation. That is, these classifiers were not evaluated by the third objective (i.e., user preference). The simplest one seems to be too low accuracy. The most accurate one seems to be too complicated to understand it.

The selected fuzzy rules in Fig. 13 are linguistically interpreted as follows:

- **R₁**: If a vehicle is on a link with moderate traffic volume, the vehicle can correctly predict the travel time.
- **R₂**: If a vehicle is about to go to a node with high link connection density and a large number of links, the vehicle can correctly predict the travel time.
- **R₃**: If a vehicle is on a link with a light traffic and is about to go to a node with a small number of links, the vehicle cannot predict the travel time correctly.

From these fuzzy rules in Fig. 13, we can see that IVC can work well under the moderate travel volume. The effectiveness of IVC depends on the conditions of the link and node just around the vehicle.

![Figure 11. Two non-dominated fuzzy rule-based classifiers at 100th generation.](image)

![Figure 12. The user-specified satisfaction level functions at the final generation.](image)

![Figure 13. A fuzzy rule-based classifier with the highest user preference at the final generation.](image)

The knowledge is based on the user’s preference that simpler classifiers are better while maintaining the accuracy. If a user doesn’t mind the simplicity of fuzzy rule-based classifiers, the user could obtain more accurate knowledge from the collected data.

4. Conclusion

We examined the effectiveness of IVC through computer simulations. We applied our interactive fuzzy data mining to the analysis of the collected data. Through a case study, we demonstrated that we can obtain linguistic knowledge from fuzzy rules about the characteristic features of IVC with respect to the accuracy of predicted travel times for vehicles. We also demonstrated that we can incorporate user’s preference into multiobjective genetic fuzzy rule selection.

As future studies, we will further examine the ef-
fectiveness of the IVC settings (e.g., communication range and frequency, different environments). We will also consider additional attributes (e.g., change in traffic volume). Furthermore, we will discuss what kinds of satisfaction level functions are needed in interactive fuzzy data mining for traffic simulations.

Although we have utilized the data obtained from the traffic simulator we developed, it could be possible to apply our system to the real traffic data. But it is not easy to obtain the IVC data, since IVC is still a developing technology now. We would like to implement our system for actual traffic data mining near future.

ACKNOWLEDGMENT

This work was partially supported by Grand-in-Aids for Young Scientists (B): KAKENHI (18700228) and Exploratory Research: KAKENHI (18650055).

REFERENCES


[TRE 99] M. TREIBER, A. HENNECKE, and D. HELBING, “Derivation, properties, and simulation of a gas-kinetic-


