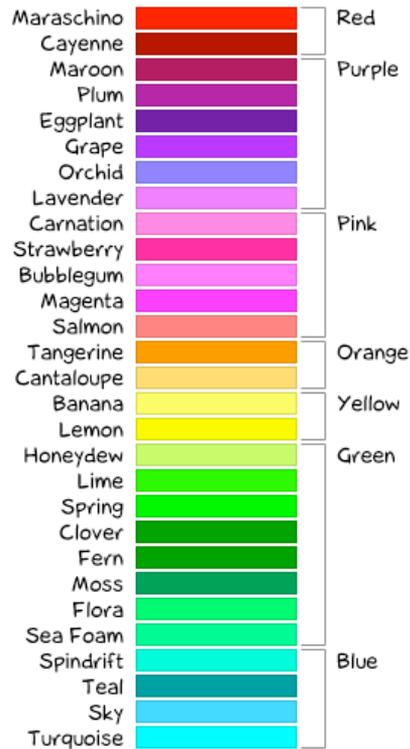

Color names if
you're a girl...



Color names if
you're a guy...

Doghouse Diaries
"We take no as an answer."

Exploring discrete choice model with fuzzy control theory

Ming LU, IVT, BAUG, ETHZ

Conference paper STRC 2012

Exploring Discrete Choice Model with fuzzy control theory

Lu, Ming
IVT, BAUG, ETHZ
Zurich

Phone: +41 633 3801

Email:

ming.lu@ivt.baug.ethz.ch

Date: 2012-04-27

Abstract

Discrete choice data are usually explained by Logit models. Assuming that the error term follows a particular probabilistic distribution, Logit model explain respondents' choices as a specialized regression model. Most of the discrete choice data use different kinds of logit models. While fuzzy control theory are widely used to deal with linguistic variables, which are encountered quite often in decision making process, such as the travel time by train is 'a little slow' but the same trip by air is 'quite expensive'. In this paper, the model applies fuzzy control theory to explore the different preferences among travellers. At first, a heuristic algorithm is used to determine the parameters of the membership functions, which means instead of just using one distribution for a certain attribute, the membership functions decompose it into several levels with several values according to how much the attribute contributes to the decisions. Then the fuzzy results are fed into Logit model to calibrate the parameters in the utility functions.

The model is also applied to a stated choice survey, which was collected in the summer of 2011 to explore Values of Travel Time Savings (VTTS) and travel time Reliability (VOR) in Switzerland. The fuzzilized results are then compared to a regular Logit model, and the results look quite promising.

Keywords

fuzzy control theory –discrete choice models – membership function – travel time reliability

1. Introduction

The random utility theory is usually used to explain discrete choice models by assuming the decision makers have a certain form of utility functions. And the final decisions are the alternative with the largest utility among all possible alternatives. As the attributes in the utility functions cannot cover all, usually an error term is added in the utility function. This error term actually contains three parts: part of decision makers' preference, unobserved variables, perception errors. Part of the decision makers' preferences can be captured by the alternative specific constants (but only in labelled model), and can also be explained by the socio-demographics. It can also partly explained by the mix-logit, which are used to account for preference heterogeneity. These three cannot full explain the preferences, so the error term includes the rest of the preferences. The unobserved variables are known by the decision makers themselves but normally not revealed to the researcher. The unobserved variables, especially in a revealed stated survey, greatly influence the results of the model. While the unobserved variables are from the perspective of the researchers, the perception errors are from the angle of the decision makers. The decision makers, especially for trip related decisions, it is nearly impossible for them to get the perfect information for the network is full of uncertainties. The uncertainty in the random utility theory sometimes means the perception errors, but generally speaking, uncertainty means the error term in the utility function, which contains all the three parts discussed above.

Fuzzy logic theory is widely used in control systems. It fuzzilizes the input into several linguistic levels into membership degrees instead of the normally used two-valued logic: true of false. Fuzzy logic theory is proved very handy to deal with maginal values that are partially true and partially false (Von Altrock, Constantin, 1995). This also works in the domain of sortation or classification when from time to time there will be attributes belongs to several classes. Membership functions, 'if-then' rules and defuzzification consist of the entire fuzzy logic system. And three parts, in the sequence of appearance, is quite similar to the human reasoning process when a decision is confronted to decide. As humans are not machines, when making decisions, precise values of each variable are not necessary. Normally an ambiguous value is enough, then several values trade off among each other and the decision is made. The fuzzy control system works exactly the same way: first the inputs are processed by the membership functions where a single crisp value are decomposed into several values; then the 'if-then' rules act as approximating reasoning functions like humans; and at last the defuzzification process the values into a recognizable format. Applying fuzzy logic theory in discrete choice models has been done by many researchers, which will be discussed in the following section.

In this paper we apply a fuzzy logic system, known as Adaptive Neuro-Fuzzy Inference System (Jang 1993) (ANFIS or Adaptive-Neuro-based Fuzzy Inference System, Adaptive Neural Fuzzy Inference System), to a stated preference survey, where back propagation is used to modify the weight of the network system and genetic algorithm is used to optimize fuzzy 'if-then' rules. In the next section an overview of the general methodology used in discrete choice models is discussed. Then in section 3 the framework of ANFIS will be explained and in section 4 the system is applied to a SP route choice survey and compared the result with Logit models. In the last section, the advantages and disadvantages of ANFIS are discussed and further steps are stated.

2. Literature review

Discrete choice models focus on using observed variables and assumed distribution to estimate the generalized uncertainties. The decision makers, or in the travel behavior domain, trip makers' decisions among available choices reflect their perception of the utilities associated with each. Random utility models are mostly used such as Multinomial Logit, Generalized Extreme Value model and Mixed Logit model (Train 2003, Avineri and Prashker 2005, Chen and Recker 2000, Noland and Polak 2002, De Palma and Picard 2005, etc). As logit models dominate analysis of discrete choice models, there will be no further discussion in this paper.

Although expected utility theory has reigned from the very beginning of the studies on decision-making models under uncertainty, there are also other methods that can be used to explore complicated individual choices. New tendency of research on the travelers' choice models is introducing mathematical methodologies to make them more intelligent and realistic, for example, using Fuzzy Set Theory and Artificial Neural Networks to analyze the behaviors of the decision making process is one of those. Fuzzy set theory was firstly suggested by Zadeh (1965) as an approach to express the different types of uncertainties inherent in human systems. Approximating reasoning of FST is more like the simulation of decision-making process human beings encountered which is difficult to decide. In the mean time, Artificial Neural Networks may either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems without necessarily creating a model of a real biological system. Vythoulikas (2003) combined the two methods as an approach to address discrete choice behaviours. He believes that in reality, decision-makers use a few simple rules that relate their vague perceptions of various attributes to their references towards the available alternatives. Based on this approach by incorporating rule weights, which captures the importance of a particular rule in the decision making process. He presents an approach for calibrating the weights using concepts from neural networks. They are also other researchers who are using these two methods to deal with travelers' choices under uncertainty. Generally speaking, fuzzy theory is used to deal with imprecision vagueness and uncertainty characteristics that appear in models, and artificial neural networks are used as the main frame of the model due to its resemblance to the Logit model. Although the models established by ANN have some achievements, some of the models just use current ANN models, which may have some problems if not dealing with them appropriately. For example, local extreme values are a tough problem when using Back Propagation Neural Networks (BPNN), and once it happens, it will be hard to identify the results.

Most of the research can be classified into the static models, which means the models are established on a fixed database. Yet Arentze and Timmermans (2003) developed a framework

for modelling dynamic choices based on a theory of reinforcement learning and adaptation. According to this theory, which leads the research to a dynamic, intelligent level, individuals develop and continuously adapt choice rules while interacting with their environment (Teodorovi, D. 1994). It is believed to be one of the research directions of the future.

Although travellers' behaviours have been studied for quite a long time, there is still much work left to do. All the current models play significant roles for the explanation of individuals' behaviours, and solve many problems in the real world. However, they may meet some problems because the travellers' choices are too complicated in the actual situation to simulate using just one theory or model. The future work is going to focus on the combination and improvement of the existing model, search for method to simulate the dynamic process of the decision making so as to make models that will be more suitable, more efficient and more intelligent.

3. Model framework

3.1 Fuzzy control theory

When answering stated preference questionnaires, the respondents usually compare the value of the same attribute, and then they will compare the trade-offs between different attributes to make the final choices (Pang, G. K. H, etc., 1999). This decision making process can be captured by the fuzzy control theory. For the following scenario the linguistic value can be expressed as

‘If the xx is (linguistic adjective), and YY difference is (linguistic adjective),
then the choice is zz, and ..’.

where xx, yy indicate attributes, zz and ... indicate consequences.

For every single respondent who is answering this scenario, he might follow this principle:

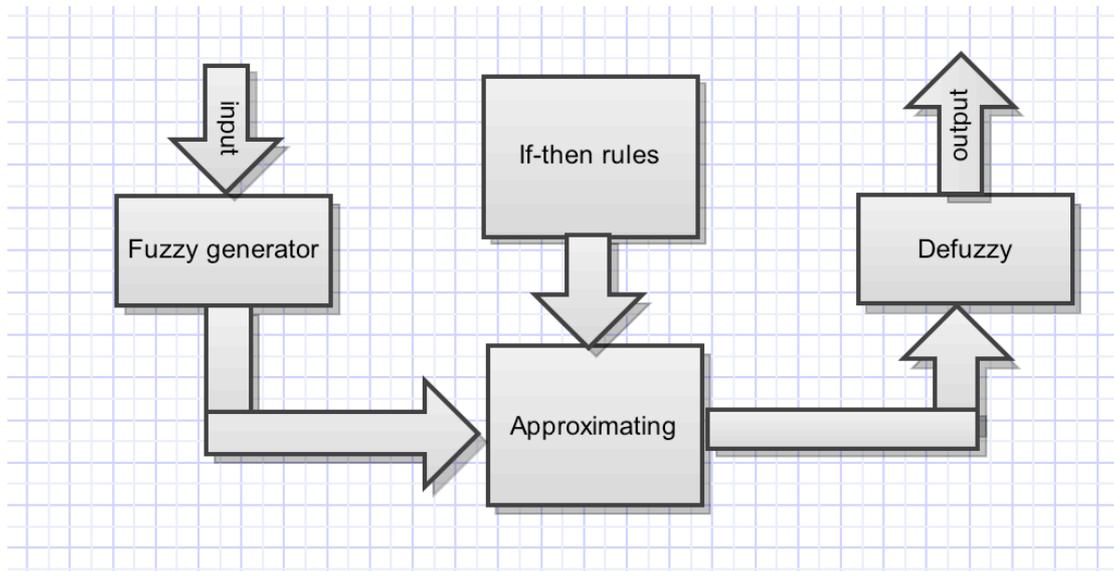
‘If the travel time difference is positively high, the travel cost difference is negatively low, and stop events difference is basically the same, ... then I will choose route 1.’

For a subset of population, their principle to choose the alternatives might be:

‘If the travel time difference is positively high, the travel cost difference is negatively low, and stop events difference is basically the same... then I will definitely not choose route 1 and I will probably choose route 2.’

The mechanism behind this decision making process consists of three parts:

Figure 1 Fuzzy control system



In the fuzzy set generation part, the inputs are divided into several linguistic levels. After passing by the membership function, the input proceeds to the approximating reasoning part, where normally there is a fuzzy rule base ready for use, and these rules are either from experiences or deduced by heuristic algorithms. At last the defuzzy part, depending on the type of the system, generates the results.

Generally speaking, there are three types of fuzzy logic system: the traditional fuzzy logic system, Takagi-Sugeno fuzzy logic system and generalized fuzzy logic system. The first type, as its name says, has a quite traditional framework. The input passes by membership functions in the fuzzy generator, dividing into several linguistic values; then those values are put into the approximating reasoning box, and according to the ‘if-then’ rules, the format of the output in this part is the same as the fuzzy generator. Humans or machines are normally cannot recognize this format, and the defuzzy part works as the interpolation or translator for the system. For the Takagi-Sugeno system the fuzzy generator part works the same, but instead using membership function in the post part of the ‘if-then’ rule, it uses a linear format of the input values, like:

If x_1 is (linguistic adjective), x_2 is (linguistic adjective) ... x_n is (linguistic adjective),

$$\text{then } y = c_1x_1 + c_2x_2 + c_3x_3 + \dots + c_nx_n.$$

Then the output are calculated for control systems. The difference is that in a Takagi-Sugeno system, the output of the ‘if-then’ rules can be calculated and therefore the fuzzy rules can be

decided. The generalized fuzzy system follows the structure of the basic fuzzy system, but varies in detail.

- **Fuzzy generator**

The fuzzy set generator separates the inputs into the linguistic values. Normally triangular or Gaussian functions are used to process the input. In figure xx is the membership function with three linguistic levels, for both the travel time difference and the travel cost difference, the linguistic values here is ‘negatively low’, ‘indifference’, and ‘positively high’. The reason for separating the attributes values is that the respondents can usually capture the information of the questionnaire vaguely with uncertainty involved.

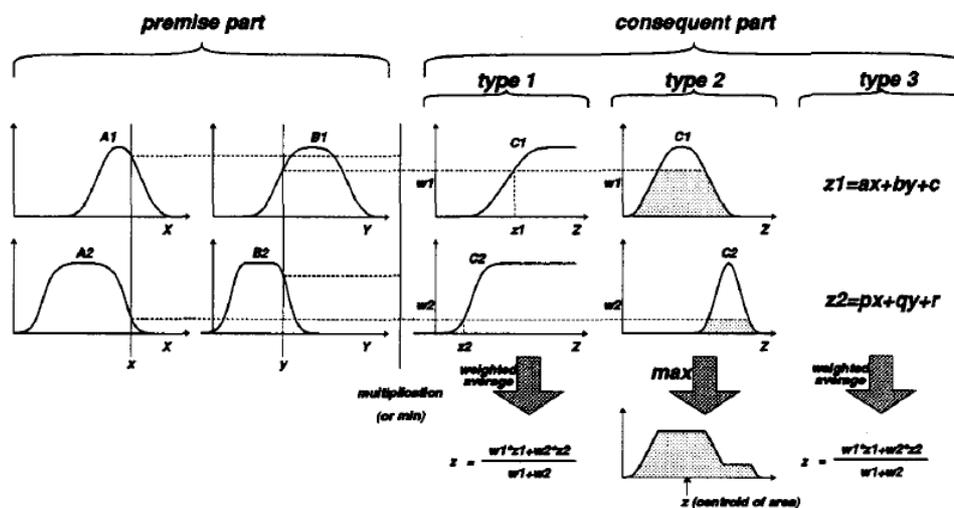
- **Fuzzy rule base**

The fuzzy rule is the most difficult part of the fuzzy interfering system. For the more inputs it has, the more the fuzzy rules it will get. As the number of total fuzzy rules is the number of linguistic levels of each input timing together. So if there are more than three or four inputs, the number of the total fuzzy rules will be massive, and normally this is quite difficult to decide which fuzzy rules should use in the system.

- **Defuzzy layer**

Three are types of defuzzy methods: weighted average, centroid and Takagi-Sugeno system, which are shown in figure 2.

Figure 2 Three types of defuzzification



Source: (Jang 1993)

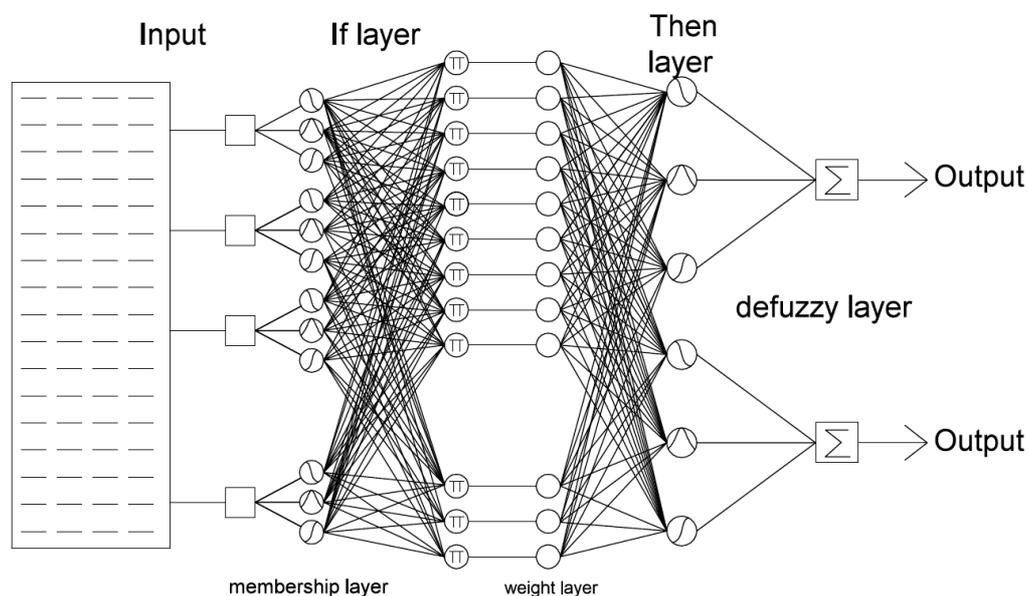
The weighted average is simply the sum of the weighted ‘then-rule’ output divided by the sum of the weights. The centroid method uses a fuzzy max operation to deal with the ‘then-rule’ output. This is quite similar to calculate the centroid of an unregularly combined shape. The third one is different from the former two. Each rule has the combination of a linear form of each input with a constant. Then the results are calculated as the first method.

3.2 Model Framework

ANFIS is quite similar to a RBF (Radial Basis Function) neural network (Celikoglu, H. B., 2006), but differs in detail. The model is basically in integrate fuzzy logic into a neural network model where the weight can be updated according to the actual output. The connection of the ‘if-rule’ layer and the ‘then-rule’ layer can be completely connected, but this is not necessary for two reasons: firstly, if the dimension of the input is larger than three, then the if-then rules will be enormous, this will massively influence the computing power; and secondly, as the model tries to simulating the decision making process of humans, normally not every rule is needed. As the model explore trade-offs among all attributes, each ‘if rule’ is combined by every attribute. Generally speaking ANFIS cannot pick up ‘if-then’ rules by itself. This is normally the case but not absolutely true. It depends on which updating methods are using, certain kinds of networks can choose ‘if-then’ rules automatically. Some researchers use a full connection network (when the dimension of the input is small); others specify the rules using empirical or heuristic methods. Here a simplified genetic algorithm is used to get the ‘if-then’ rules, which will be discussed in the following section.

The following figure illustrate the structure of the model (Celikoglu, H. B., 2006, Vythoukcas, 2003)):

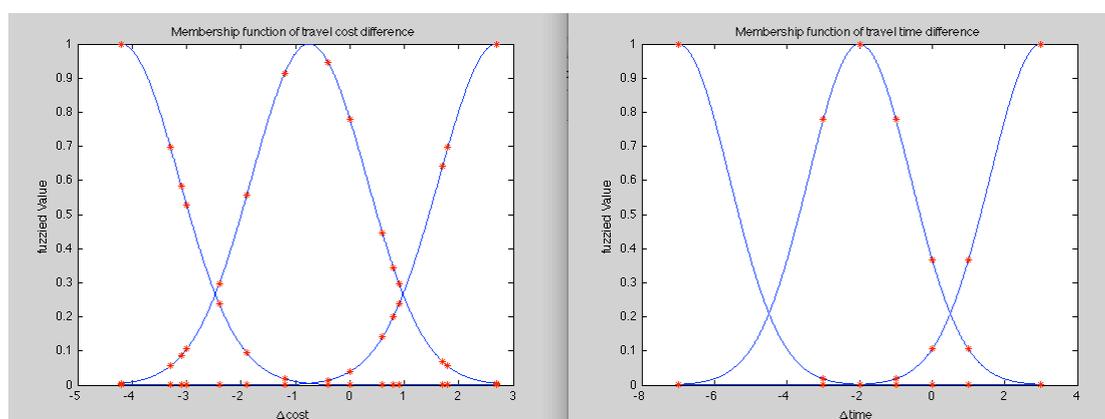
Figure 3 ANFIS structure



The input are the output are shown in the first layer and the last layer respectively. Basically, the input are fed into specified membership function to prepare the ‘if’ rules, which is shown in the second layer. After the input are transferred by the membership functions, the third layer acts as the ‘if rule’ layer. In order to improve the prediction results, an extra layer was added to weight the ‘if rule’s. Then the weighted values pass to the ‘then-rule’ layer, where the linguistic decisions are deduced. At last, in the de-fuzzy layer, the second method is used to get the final output, where the centroids of the ‘then-rule’ layer are calculated. Detailed calculations of each layer’s input and output can be seen in table 1.

For the membership layer, each input is decomposed into several levels, and Gaussian function is used for each linguistic variable (Cheng, H., 1997). For the first and the last linguistic level, s-shaped functions are used (The function is still Gaussian function, but only half of it, see figure 4).

Figure 4 Membership functions



For the 'if layer' part, as mentioned before, the model tries to explore the over all trade-off among all the attributes, so the 'if rule' will contains all the input at a certain linguistic level. Actually there is a hidden layer after each of the linguistic level times together, and the weights in this are binary values, where if the 'if-rule' is accepted, the weight is set to 1, otherwise a zero value is assigned.

In a tradition ANFIS, the weight layer does not exist. The purpose of adding this layer is to improve the model results. The number of the weights in this layer is the full combination of the inputs. In the weight updating process, if the 'if rule' is unaccepted, then corresponding weight doesn't change for the value before is set to zero. This means all the unaccepted 'if-rule' nodes are deactivated.

In the 'then-rule' layer, each 'then-rule' output accepts all the 'if-rule' times a binary weight, which if the 'then-rule' is activated, the value is 1 otherwise 0. Each of the weight layer output could only connect to each output once. To put it another way, each 'if rule' can only pass to one of the linguistic levels of a certain output once. The membership function for the 'then-rule' also uses Gaussian function, of which parameters are used later to calculate the centroid.

For the defuzzy layer, each output of the 'then rule' layer is calculated separately as they have dependent centroids, as shown in figure 2.

3.3 Genetic algorithm for searching if then rules

As mentioned above, the most difficult part of ANFIS is to get the ‘if-then’ rules. As the data used in this paper are more than 3 dimensions, it is far too complicated if full combinations of the ‘if-then’ rules are used. So genetic algorithm (GA) is used here to choose the ‘if-then’ rules (Almejalli, K, etc., 2007, Lin, C.J., 2004).

GA is a heuristic algorithm, which is used to search for optimum solutions. It is a specific algorithm from a more generalized algorithm called Evolutionary Algorithm. These algorithms are inspired from natural evolution, where the first generation is usually randomly generated or from empirical data. Then a fitness function is defined and individuals in this generation is selected according to the fitness function. Those selected individuals are chosen as the parents of the next generation, which are inheriting, mutating or crossing each other’s gene over to generate a number of offspring. This process continues until the fitness function converges or reaches to a certain generation. A detailed description of how the algorithm works is not discussed here. To see how GA optimizes ANFIS, please go to Appendix B.

4. Application

4.1 Data collection and first step analysis

From May 2011 to August 2011 a survey was launched, aiming to capture travellers' perceptions of travel time variations based on an SP data. The recruited area was from the entire Switzerland. The data collection process consisted of two periods. In the first period the data is collected by LINK institute, which is integrated in the KEP survey sponsored by SBB. In this period, information collected include: daily trips, social-demographics, GA ownerships, car ownerships etc., which is included in the original KEP survey, and also additional questions are asked about routine trips of the respondents. In the end of the regular KEP survey, the respondents were asked if they are interested in participating another survey. For those who agreed to continue the survey, a paper and pen survey about the trip mentioned in the additional questions will be mailed to the respondent a week or two weeks later. In this pen and paper survey (see appendix C), two scenarios are designed each with 8 situations. Here only scenario 2 is interested and the form of the scenarios is shown.

4.2 ANFIS application

As there are only two alternatives in the model, and also for the purpose of reducing input dimensions, the differences between same attributes are calculated as input of ANFIS. Therefore the inputs are: travel time difference, travel cost difference, slowdown-event difference, stop-event difference, distribution parameter sigma difference and distribution parameter mu difference.

The membership function divides the inputs above into three linguistic levels respectively, expressed as 'negatively low/less', 'indifferent' and 'positively high/more'. These three levels literally mean the difference of each attribute between the two routes. For the outputs, the membership function divides each of them into five levels: 'definitely not', 'probably not', 'don't care', 'probably', 'absolutely'. All the membership functions use Gaussian function with two parameters, μ for the expected value and σ for the standard deviation.

Once the input and output membership functions are defined, then form of the 'if-then' rule is decided as:

'If the travel time difference is negatively low, travel cost difference is positively high, slowdown events difference is negatively less, stop events difference are positively more, sigma is negatively smaller and mu is positively larger, then the respondents will probably choose route1 and probably not choose route 2.'

As many researchers mentioned before, ANFIS appears as a black box. Due to the structure of the model, it is impossible for ANFIS to calculate marginal values of the attributes. However, hit ratio can be calculated to predict if the output is close to the actual choices, and indicate if the model is suitable or not. The results are shown in table 1.

Table 1 ANFIS result

Hit Ratio	Initial error	Final error	Total if-then rules	Selected if-then rules	Learning rate	GA generations	GA offspring
65.26%	932.962	636.609	6075	520	0.1	100	100

Note: Hit ratio is the number of correctly predicted choices divided by total sample.

The hit ratio indicates that ANFIS can predict almost two thirds of the choices correctly. The total it-then combinations are all the linguistic levels of each fuzzy input and output times together. As there are six inputs and two outputs, the total rules reach to 6075. But after optimization, only less than 1/10 rules remain. Due to the limitation of the computer memory, GA only iterates for 100 generations, and for each generation, there are 100 children reproduced.

4.3 Exploring heterogeneity using modified membership function

For the purpose of exploring the travel time reliability, three types of modes are usually used: mean variance model, schedule model and schedule late model. The mean variance model, as its name implies, uses mean travel time and variance or standard deviations of travel time to explore the travel time reliability. The reliability is the marginal value of standard deviation of travel time divided by the marginal value of travel cost. The schedule model basically divided the variance in the mean-variance model into scheduled earliness and scheduled lateness with an extra attribute to indicate being early or late. As there are two variables to explain the effects of disutility on trips, the reliability is also divided into two parts. They are the partial derivatives of the earliness and lateness divided by partial derivative of travel cost, respectively. Based on the fact that the disutility of being late is more disliked by that of being early, the schedule late model (sometimes it called mean lateness model or lateness model) is proposed. It is only a simplified version the schedule model.

For the probability in the questionnaire, it is assumed that the travel time follows a log-normal distribution, which is shaped positively skewed with a long tail. Given mean travel time $E(t)$, and variance $Var(t)$, then for Log-normal distribution $\ln \mathbb{N}(\mu, \sigma^2)$:

$$f(t) = \frac{1}{t \cdot \sqrt{2\pi \ln\left(1 + \frac{Var(t)}{[E(t)]^2}\right)}} e^{-\frac{\{t - [\ln(E(t)) - \ln\left(1 + \frac{Var(t)}{[E(t)]^2}\right)]\}^2}{2 \ln\left(1 + \frac{Var(t)}{[E(t)]^2}\right)}}$$

$$\mu = \ln(E(t)) - \frac{1}{2} \ln\left(1 + \frac{Var(t)}{E^2(t)}\right), \sigma^2 = \ln\left(1 + \frac{Var(t)}{E^2(t)}\right)$$

where t – travel time;

$f(t)$ – density function for travel time;

$E(t)$ – expected travel time of the trip;

$Var(t)$ – travel variance of the trip;

μ, σ – parameters of lognormal distribution;

Instead of using mean and variance in the model, the two parameters of the travel time distribution are used here. The results are shown in table 2. Even though ANFIS is not made to apply to logit models, the fuzzilized models show better-adjusted values. Most parameters show correct signs, where negative values indicate disutility and positive value mean utility. There is only one parameters, the cost of 'positively higher' shows the incorrect sign. The reason may lie in the other two linguistic levels. Here three linguistic travel time levels share the same parameter. The value of travel time savings and travel time reliability are not calculated as the linguistic levels then incorporate with each so it does not make any sense to calculate these values.

Table 2 Modified mean-variance model before and after fuzzilization (Bierlaire, M. 2003)

Parameters	Before		After	
	MNL	MMNL	MNL	MMNL
β_{cost}	-0.615	-0.151	-0.304	-0.435
β_{cost2}			-0.542	-0.605
β_{cost3}			0.158	-0.348
σ_{cost}		0.378		1.11, -0.0275, 1.69
Route 1 & Route 2 β_{time}	-0.444	-0.586	-2.44	-0.726
σ_{time}				0.225
β_{delay}	-0.124	0.541	-0.132	-0.0918
β_{delay2}			-0.367	-0.373
β_{delay3}			-0.0769	-0.0125
β_{mu}	0.0633*	-0.0366*	0.610	0.119
β_{mu2}			0.618	-0.328
β_{mu3}			0.572	0.228
β_{sigma}	-0.128	-0.189	-0.0287	-0.189
Adjusted- ρ^2	0.118	0.182	0.155	0.182

Note: parameters with * mean the parameter is insignificant and did not pass the T-test.

5. Conclusion and problems

The logit results of the model are not perfect, yet it is somehow promising. As ANFIS is mostly applied to the control systems, it is usually better at predicting. Even though a genetic algorithm is used to find optimal if-then rules, the final if-then rules are still more than expected. This can also demonstrate that the decision making process of human beings is far more complicated and normally a simple model cannot simulate it.

Most of the ANFIS models avoid using more than three inputs, yet here six inputs are used. From the angle of computing powers, the speed of the calculation is relatively not too slow. On the other hand, due to the structure of ANFIS, it needs a huge amount of RAM. For six inputs, 100 generations with 100 offspring at each generation, the model needs more than ten Gigabits of memory and the whole process needs nearly 18 hours.

For the structure of the model, there is still much room to improve. The weight updating process presently only uses fastest gradient descent algorithm, while in the future a hybrid method will be used to accelerate the convergence. And for both the weight updating and 'if-then' optimizing, currently random number generation is used to avoid local minimum, but this will be done by adding more sophisticated algorithms.

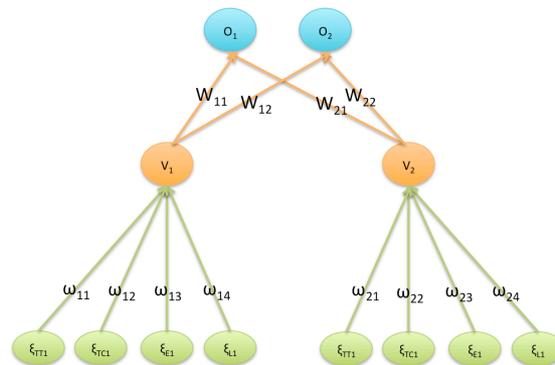
Appendix A: Back propagation for weight modifying

The back propagation algorithm (Pfeifer R, etc., 2010) is usually used in supervised neural networks where the weights can be modified according to the actual output. The principle of the back propagation is that it tries to converge to the direction of the actual output, and it must have input-output pair to ‘learn’ the weight. As ANFIS is a special form of neural networks, the back propagation works quite similar to the other neural network. And for the model in this paper, the weight modified by BP is the weight layer weights, sigma from both input and output membership functions.

A typical back propagation algorithm is composed of 3 steps: system output calculation, error term back propagation and weight updating. The generalized form of this process is illustrated as follows:

- **System output calculation**

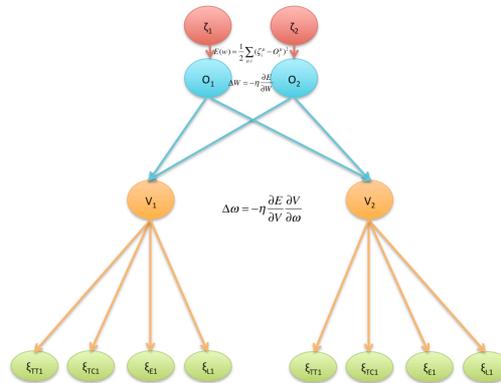
Figure 5 Forward propagation



As shown in figure, ξ_{mn} is the input of the system, in ANFIS, this ξ_{mn} is the output of the membership function. All the input pass forward by the middle layer V_k (in ANFIS this layer consist of the if layer, weight layer and then layer, and the links between the input and middle layer are cross linked) and system output O_i is calculated (This represents the defuzzy layer in ANFIS).

- **Error term back propagation**

Figure 6 Error backward propagation



Source: (can be deleted if not used)

After the outputs are calculated, they are compared with the actual results and an error term is

calculated as

$$E(w) = \frac{1}{2} \sum_{\mu,j} (\zeta_i^\mu - O_i^\mu)^2$$

where $E(w)$ is a function of the weights to be updated (or modified, for ANFIS is the weight layer weights and the parameter sigma for both input and output membership functions).

Weight updating

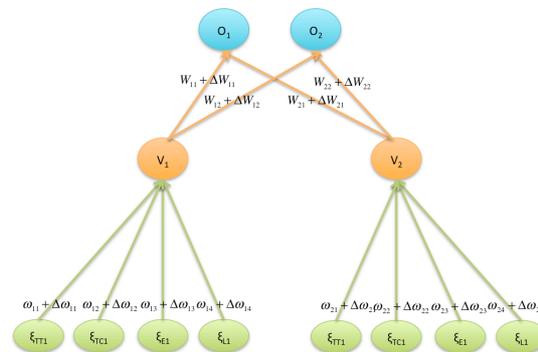
For the weight-updating step, the partial derivatives of the weight are calculated. For the weights between the output and the middle layer, the value of the weight changed is

$$\Delta W = -\eta \frac{\partial E}{\partial W}$$

, and for the weights between the input and the middle layer, the value of the

weight changed is $\Delta \varpi = -\eta \frac{\partial E}{\partial V} \frac{\partial V}{\partial \varpi}$, where η is the learning rate as shown in figure xx. Then these values are added to the original weights, as shown in figure xx, and the network is ready for the next iteration.

Figure 7 Forward propagation after weight updating



• Weight updating

For the weight updating process, there are two versions to update the weights: the on-line version and the off-line version. For the on-line version, all weights in the network remain the same until all the input data are calculated. Then the error is calculated as the total error of all inputs. The weights are updated as described above. The other version is off-line learning, in which the weights are updated each time an output pair is calculated according a single input pair. This means that the order of the inputs may influence the value of the weights. Under this circumstance, especially when the learning rate is inappropriately specified, it may cause the weights oscillate in a certain range. Due to this reason, the ANFIS uses the on-line learning version to update the weights.

Appendix B: Genetic algorithm for optimizing if-then rules

The genetic algorithm is used in this paper to optimize the ‘if-then’ rules. The basic steps is shown as follows:

- **Initialization**

There are 100 offsprings in each generation, and the value of each offspring is the if-layer-weight and the then-layer-weight. If the rule is activated in ANFIS, then it is set to 1, otherwise it is set 0. For initialization, the parents are sets randomly, as the rules are too many (729 for only the if-layer). The number of activated and deactivated ‘if rule’ and ‘then rule’ are set to 1:5 to stimulate the optimizing process. As random number is binary value, so the distribution of the random numbers follows a uniform distribution.

Selection

The fitness function of this genetic algorithm is the total error of ANFIS, which takes the form

$$E = \frac{1}{2} \sum_{m=1}^m \sum_{i=1}^i (\zeta_i^u - O_i^u)^2$$

of:

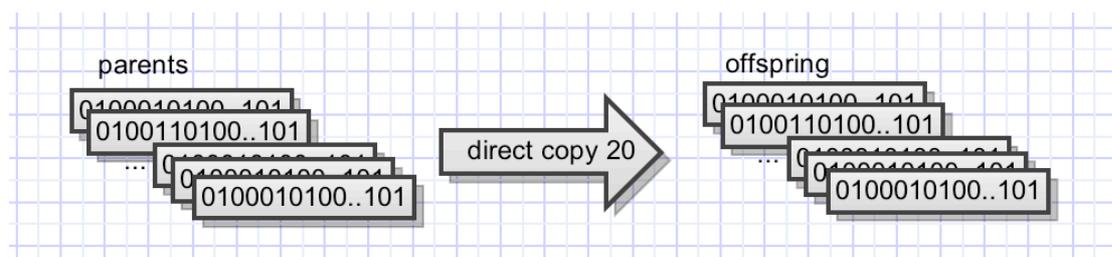
This is the same as one of the ceasing conditions of the ANFIS system.

- **Offspring generation**

Before the reproduction begins, the offspring is sorted in ascending order according to the fitness function. Then the next generation is divided into 5 subpopulations.

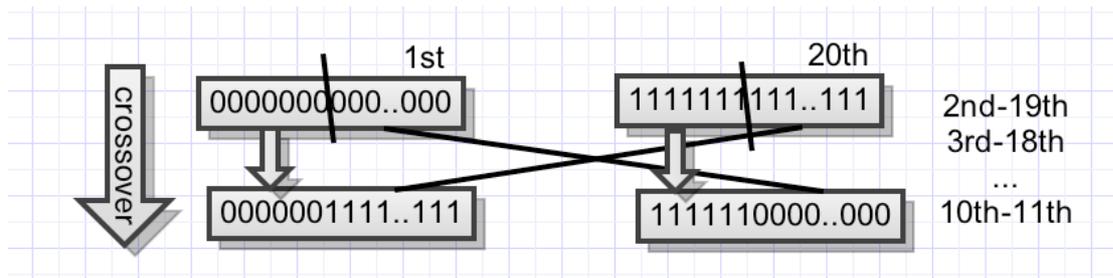
Reproduction The top twenty values (with smaller system errors) are still kept in the next generation, the other 80 parents are discarded.

Figure 8 Direct inheriting



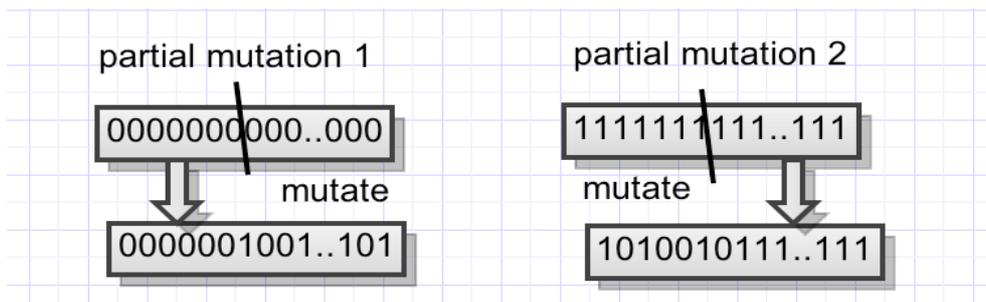
Crossover From the 20 parents, the next generation is reproduced taken half of the parents' gene. To do this, the gene is cut into half, and then first half of the keeps in their original position while the second half is from another parents in a descending order. And this will generate 20 offspring.

Figure 9 Crossover



Partial mutation 1 &2 in these two parts, each will have 20 children. In partial mutation 1, the first half of the gene is kept, while the other half are generated randomly. In partial mutation 2, the second half of the gene is reproduced while the first half is dropped and randomly filled up.

Figure 10 Partial mutation



Total mutation One of the drawback for genetic algorithm is that sometimes it can be trapped in a local minimum. There are a lot of algorithms developed to avoid this issue, such as simulated annealing. But here in order to simplify the model, the last 20 'children' (there are not actually children anymore) are generated randomly as the initialization part. This also cannot guarantee that it will get the global minimum value but for the moment, the results are best enough.

- **Drawback of GA**

As the above genetic algorithm is only a simplified version modified for optimize the ‘if-then’ rule of the ANFIS system, unfortunately it has several disadvantages:

As mentioned before, as GA is only a heuristic algorithm, it cannot grantee to find the best solution. But for an approximating reasoning system, there is probably no ‘best’ solution.

If the dimension of the input is more than 3, then the algorithm converges really slow, but as the dimension is relatively large, it also consume tremendous computing power. This is the reason that in most literatures, the premier part of the approximating reasoning seldom appears in more than 3 combinations.

Appendix C: Route choice scenario

Situation X

route 1	route 2
slowing down events ¹ 8 events	slowing down events ¹ 11 events
stopping events ² 5 events	stopping events ² 6 events
travel time 24 min	travel time 22 min
you have the following chance of arriving destinations	you have the following chance of arriving destinations
20% more than 10 min early	15% more than 10 min early
13% 5~10 min early	14% 5~10 min early
39% on time	45% on time
9% 5~10 min late	9% 5~10 min late
19% more than 10 min late	15% more than 10 min late
travel cost 9.8 Fr.	travel cost 10.1 Fr.
your choice <input type="radio"/>	<input type="radio"/>

Note: 1. Slowing down events mean junctions where you need to slow down the vehicle speed.

2. Stopping events mean junctions or zebras where you need to stop your car and let other cars or pedestrians pass.

6. Literature

- Almejalli, K., Dahal, K., & Hossain, a. (2007). GA-Based Learning Algorithms to Identify Fuzzy Rules for Fuzzy Neural Networks. Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007), 289-296.
- Arentze, T.A., and H.J.P., Timmermans (2003) Modelling learning and adaptation processes in activity-travel choice, *Transportation*, 30 (1) 37–62.
- Avineri E, and J.N. Prashker (2005) Sensitivity to travel time variability: Travellers' learning perspective, *Transportation Research Part C*, 13 (2) 157–183.
- Bierlaire, M. (2003) BIOGEME: A free package for the estimation of discrete choice models, paper presented at the 3rd Swiss Transportation Research Conference, Ascona, March, 2003.
- Celikoglu, H. B. (2006). Application of radial basis function and generalized regression neural networks in non-linear utility function specification for travel mode choice modelling, 44, 640-658.
- Celikoglu, H. B. (2006). Application of radial basis function and generalized regression neural networks in non-linear utility function specification for travel mode choice modelling. *Mathematical and Computer Modelling*, 44(7-8), 640-658.
- Chen, A. and W. Recker(2000) Considering risk taking behaviour in travel time reliability, Working Paper UCI-ITSWP-00-24, Institute of Transportation Studies, University of California, Irvine.
- Cheng, H. (1997). Automatically determine the membership function based on the maximum entropy principle. *Information Sciences*, 182(1997), 163-182.
- De Palma, A. and N. Picard (2005) Route choice decision under travel time uncertainty, *Transportation Research Part A*, 39 (4) 295-324.
- Jang, JSR. 1993. "ANFIS : Adaptive-Network-Based Fuzzy Inference System." *Systems, Man and Cybernetics*, IEEE Transactions 23 (3).
- Lin, C.-J. (2004). A GA-based neural fuzzy system for temperature control. *Fuzzy Sets and Systems*, 143(2), 311-333.
- Noland R. B. and J. W. Polak (2002) Travel Time variability: a review of theoretical and empirical issues, *Transport Reviews*, 22 (1) 39-54.
- Pang, G. K. H., Takahashi, K., Yokota, T., & Takenaga, H. (1999). Adaptive Route Selection for Dynamic Route Guidance System Based on Fuzzy-Neural Approaches, 48(6), 2028-2041.
- Pfeifer, R., Damian, D., & Fuechslin, R. (2010). *Neural Networks* (p. 29-32).

Teodorovi, D. (1994). Fuzzy sets theory applications in traffic and transportation, 74, 379-390.

Train K.E. (2003) Discrete Choice Methods with Simulation, Cambridge University Press, London.

Von Altrock, Constantin (1995). Fuzzy logic and NeuroFuzzy applications explained. Upper Saddle River, NJ: Prentice Hall PTR. ISBN 0-13-368465-2.

Vythoulkas, P.C (2003) Modelling discrete choice behavior using concepts from fuzzy set theory, approximate reasoning and neural networks, Transportation Research, 11 (1) 51-73.

Zadeh, L.A. (1965) Fuzzy sets, Information and Control, 8 (3) 338–353.