Motion Prediction in Mobile Communication Systems from a Historical and Geographical Perspective

Zhiwei Cen, Hong Chen, Xiaomei Liu
cenzhiwe@msu.edu, chehon2@msu.edu, liuxiaom@msu.edu
CSE812 Group Project Report, Fall 2002.
Computer Science and Engineering Department, Michigan State University

Abstract

Motion prediction serves to improve quality of service for mobile hosts in mobile communication systems. Much work has been done to make motion prediction using statistical method or shadow clusters. In this study we try to attack this problem through the statistical method and ANFIS (adaptive neural-fuzzy inference system) method. In the simulation process, we generated mobile host motion pattern data with specified geographic information, which is much closer to the real case compared with probabilistic models. Based on the generated data, simulations of both statistical and ANFIS model have been implemented and comparison of the two methods shows their advantages and disadvantages.

Keywords

Motion prediction, ANFIS (adaptive neural-fuzzy inference system), Mobile communication

1. Backgrounds

In recent years we witnessed a tremendous growth of wireless communication systems. With the help of wireless communication system, people can theoretically reach every corner of the earth. Everyday millions of people are using pagers, cellular telephones and other wireless communication devices to exchange information. No longer bound by the cumbersome wired networks, people are expecting an age where information can be exchanged with maximal freedom.

A mobile communication system is a network where mobile nodes are involved in data communication. Since both mobile communication and wireless communication are used in the literature, we would use these terms interchangeably in the following text (though they are almost referring to the same thing). The goal of mobile systems is to provide easy-to-use functionalities to mobile users or *Mobile Hosts* (MH). Mobile communication system differs from wired communication system in both the physical layer and medium access control layer. In upper layers mobile communication might demonstrate different characteristics from wired communication, but the techniques used in these layers are almost the same. The physical layer, which is responsible for transmitting data from one node to another, may use radio or infrared technology in wireless communication systems. The media access control layer deals with issues of using the shared physical medium.

There are two ways to configure a wireless communication system: ad-hoc and infrastructure. As defined in the chapter of MANET working group of IETF ([MANET]), "a 'mobile ad hoc network' (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links -- the union of which forms an arbitrary graph." In ad-hoc network, there are no fixed points, and every node is able to communicate with every other node. Of course the mobile hosts can also connect to the wired networks through some way.

The second type is the infrastructure. In the infrastructure mode, mobile hosts must communicate with each other through stationary access points. These access points are sometimes called *Base Stations* (BS). The coverage of each base station is called a *cell.* Several BSes are connected to a *Mobile Switching Center* (MSC). MSC will process data communication for a large area and is also connected to wide area network or public switched telephone network.

In mobile communication systems, if an in-service MH moves from one cell to another cell, the MH must be connected to the BS of the new cell and rerouting is also involved. This is called *hand-off*. Clumsy hand-off mechanisms often cause broken connections because it is possible that the BS of the destination cell has no channel to allocate to the incoming MH. One possible solution is to predict the motion of the MH to guess its next cell. Then the cells involved will be informed to reserve resources for the upcoming MH.

Motion prediction involves studying the motion pattern of the MH. Several motion prediction methods are presented ([CS98], [LAN95]) and most of them are based on an aggregate history of hand-offs observed in each cell. In [CS98], the behavior of a MH is assumed to be probabilistically similar to the MHs which came from the same previous cell and are now in the current cell. [LAN95] introduced the concept of prediction using shadow clusters. Simulation shows these approaches works fairly well in general geographical areas. However, their adaptability to new environment and graphical situation is no so ideal.

In this study we try to solve the problem through the statistical method and ANFIS (adaptive neural-fuzzy inference system) method. In the simulation process, we generated mobile host motion pattern data with specified geographic information, which is much closer to the real case compared with probabilistic models. Based on the generated data, simulations of both statistical and ANFIS model have been implemented and comparison of the two methods shows their advantages and disadvantages.

The paper is organized as follows. In section 2 and 3, the statistical motion prediction and neural network motion prediction methods are described. Section 4 is about an important part of the simulation process: data generation. Section 5 focuses on the simulation process and results. A comparison of the simulation results of the two methods is also given in this section. Section 6 is the conclusion.

2. Statistical Motion Prediction Solution

One trend of motion prediction solution is via statistical method. Here, the problem of motion prediction can be abstracted into the following model. Time is treated as discrete clicks. A mobile host *MH[i]*, has an initial position *P[i][0](x, y)* at time 0, and is controlled by base station *BS[i][0]*. By being controlled by a base station, we mean that the host is under the signal coverage cell of that base station and the MH should do any communication through that base station. Since the base station residing in a cell can represent that cell totally, we may use *BS[i]* to denote the cell *MH[i]* resides. At time *t*, *MH[i]* would move to a new position *P[i][t](x, y)*, and be controlled by base station *BS[i][t]*. Motion prediction is to find the next base station *BS[i][t+1]* at time *t+1* for mobile host *MH[i]*, or find the probability of mobile host *MH[i]* entering certain base stations at time *t+1*.

2.1 Previous Work

In [CS98], a hand-off estimation solution is introduced. For each mobile which moves into an adjacent cell from current cell 0, the cell 0's BS caches the mobile's quadruplet, (T_{event}, *prev*, *next*, T_{soj}), called a *hand-off event quadruplet*. Here T_{event} is the time when the mobile departed from the current cell, *prev* is the index of the previous cell the mobile had resided in before entering the current cell, *next* is the index of the cell the mobile entered after departing from the current cell, and T_{soj} is the sojourn time of the mobile in the current cell, that is, the time span between the entry into and departure from the current cell. From the cached quadruplets, the BS builds *hand-off estimation function* (HOE), which describes the estimated next cell and sojourn time of a mobile, depending on the cell the mobile stayed before. Later on the HOE would be used in resource reservation mechanisms.

In [LAN95], an *active mobile probabilities* set is calculated for all cells, describing the possibilities that the mobile host would enter that cell in the next time click.

2.2 An age based prediction algorithm

Based on the solution in [CS98], we presented a new prediction algorithm. We explicitly defined the hand-off estimation mechanism, which is not given in [CS98]. We also simulated this algorithm and compared its performance with other prediction algorithms.

In this algorithm, each base station maintains a prediction cache, which would reflect the historical pattern of the motion. Each cache entry has five elements: (*age, pre1, pre2, speed, pred). age* denotes the freshness of the cache entry, the larger the age, the fresher the entry is. Initially *age* is assigned to a large number. In each time click, *age* is decreased by one. When it reaches zero, this entry would be treated as an invalid cache entry. *pre1* and *pre2* are indexes to the most two recent previous cells this host has traversed. *speed* is the estimated current moving speed of the mobile host. We extended the quadruplet in [CS98] by adding an additional previous cell index. This does not increase much storage but can improve the performance in a large margin. Event and sojourn time are not included in the cache entry. The reason is that in our system time are treated as discrete clicks so in every click the host behavior will be captured.

At each time click, the base station will make prediction for the mobile hosts under its control. The following function is used to select a cache entry used to make the prediction:

$$F_{pred} = \frac{dist(host.prev1, cache.prev1) + dist(host.prev2, cache.prev2) + dist(host.cur, cache.cur) + abs(host.speed - cache.speed)}{cache.age}$$

Here *dist()* is a function to calculate the distance between too base stations. *abs()* is used to calculate the absolute value. The cache entry which produces the smallest F_{pred} is used to make the prediction.

After a new mobile host joins a cell controlled by the base station, the cache is updated. The information of the new coming host is filled into the cache. If there is not any invalid, or empty, cache entries, cache replacement happens. In cache replacement, the entry with the smallest *age* is discarded and used to hold information of the new host.

3. Neural Network Prediction Solution

3.1 Introduction:

Another possible trend to solve motion prediction problem is to use neural network methods. Instead of modeling the data under traditional mathematical hypothesis, neural network methods loosen the requirement of the hypothesis by extracting feature directly from the data. Of course, it is necessary to assume that certain pattern exists in the data. Otherwise, we cannot make any prediction. However, the neural network method does not require an accurate model definition, which is usually very difficult to do in problem solving.

Not just neural networks [BEK97], we have considered using ANFIS (Adaptive Neural-Fuzzy Inference System). ANFIS integrated neural network and fuzzy inference system together, constructing data model with fuzzy inference techniques and choosing model parameter via adaptive neural network method. The adaptive capability of ANFIS makes it almost directly applicable to adaptive control and learning control [JANG93] such as the motion prediction problem in mobile communication systems..

Two important components of ANFIS are membership functions and fuzzy inference rules. Membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1 of the associated fuzzy sets. Fuzzy inference rules are some *if-then* rules to define output behavior from the inputs.

In the first stage, membership function must be constructed for the input data. Instead of choosing predetermined model structure based on characteristics of variables in the system, the membership function parameters of ANFIS are tuned (adjusted) via back-propagation algorithm or combined with least squares method, i.e. learning from the data. The parameters associated with the membership functions will change through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector, which provides a measure of how well the fuzzy inference system is modeling the input/output data for a given set of parameters [MW00].

In the next step the suitable fuzzy inference rules must be constructed. Three common rules in fuzzy inference system are available: Sugeno model, Mamdani model and Tsukamoto model. Mamdai model is one of the earliest fuzzy inference system proposed, it uses a set of linguistic control rules to generate the output. Sugeno model is proposed to develop a systematic approach to generating fuzzy model with the form:

If x is A and y is B then $z = f(x,y)$

where A and B are fuzzy sets in the antecedent and $f(x, y)$ is generally a polynomial in the input variables, although it can be any function that can appropriately describe the output model. In the Tsukamoto model, the consequent of each fuzzy *if-then* rule is represented by a fuzzy set with a monotonical MF, and the outputs of each rule is defined as the weighted average of each rule's output. In our system, we have chosen the first order Sugeno model type fuzzy inference system in the following form:

if x is A and y is B then $z = p*x + q*y + r$

Sugeno model is the simplest and very straightforward to be extended to other types of models. In addition, the extra complexity in structure and computation of Mamdani ANFIS does not necessary imply better learning capability or approximation power [JANG93]. By interpolating multiple linear models, Sugeno system can also be used to model nonlinear systems.

3.2 System model

In our project, we have checked different types of membership functions including "gbellmf", "pimf", "psigmf", "trimf" and "gaussmf". The experiment shows that "gaussmf" gives out the best result. So we choose "gaussmf" as our input membership function. In fact, Gauss distribution is identical to Normal distribution function and so it is a natural fit for the large sample set.

The final ANFIS system is shown in Figure 3.1. Here the inputs represent the known parameters of the mobile host, such as present position, present cell number, and speed, etc. Output defines the prediction value, that is, the predicted cell number. In the off-site training process, historical information is used as inputs to help the system acquire enough rules. In real application, on-site information is used to make the prediction.

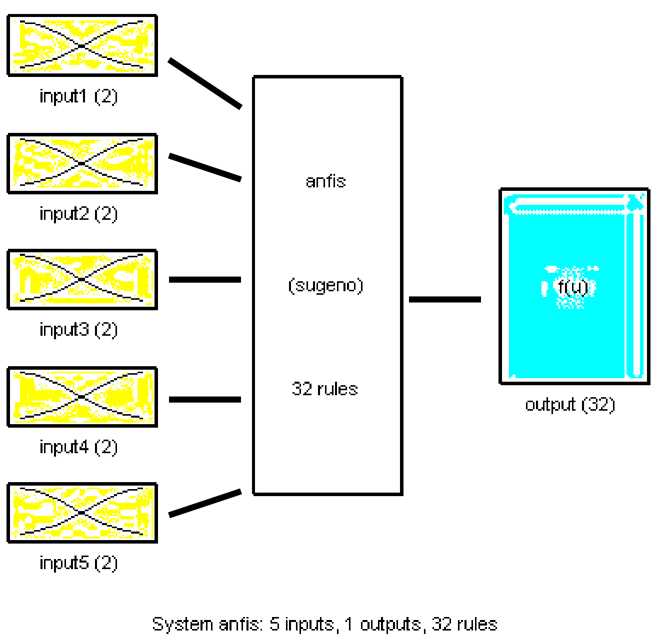

Figure 3.1, ANFIS system

4. Data Generating

4.1 Introduction

To simulate the correctness and efficiency of the prediction method, we need some way to generate the simulation data. Due to the difficulty to get commercial trace, we developed our own simulation data generation method. We also tried to use probabilistic models to generate the input data. However, the probabilistic methods can never reflect the geographical information, which is vital in the prediction method.

4.2 Assumptions and rationality

When we are trying to produce data, we have to build a motion model first. Models are an abstraction of the real world. In models, we simplify the factors, leave out the unimportant factors and keep the affecting ones. But we should be careful to keep enough information for the real world. Because the aim of the model is to simulate the world, if the factors are improperly simplified, we will get an unreasonable model and producing non-rational data.

From another viewpoint, when building a model, we are making assumptions of the non-bias world, which means, the data are originally produced stochastically. And we add some assumptions to make it biased, and the bias is the pattern we expected in the data.

There are two ways of making models. One is deducing the factors from a set, and the other is adding the factors from nought. But they reach the same result. One important issue is that we need to leave enough space for practical implementation, that is, we are not going to betray the aim of theoretical research.

The assumptions we make are:

1. Hosts are moving in the traffic system.

This is the basic assumption we make for this model. In real life, hosts are surely not following this rule absolutely. For example, in the desert or grassland, there is not a paved way for vehicles. They can move to any directions as they like. Also for mobile phones, people don’t always move along the streets when they are walking. They can get into a building or a square and move without following a predictable pattern. But for hosts on vehicles, we can say reasonably assume they move along the high way and roads. And as we have declared in the introduction, the misprediction in hand-off mostly happens for hosts on vehicles, so we omit the cases of hosts in human hands.

So we need a map in the experiments of the project. Without losing the generality, we choose the map of East Lansing. We got this map from Yahoo.com. The map covers an area of about 64 square miles. And there are three interstate high ways and several roads in the map. We selected three highways and several normal roads in the map without losing generality.

2. The speed of motion

Vehicles can move in variant speeds along the roads. However, we simplified this limit by setting two speed levels for the vehicles when they move along different roads. The Speed along the high way is set to be 2 pixels, which means the host can change their positions as far as two pixels from their original position, and the speed along normal ways to be 1 pixel per time unit. In addition, we can change sampling intervals to generate speed which is close to the real world.

3. The turning at intersections

We assign a probability of turning to different roads at the intersections. The pattern of turning can be obtained from processing real world data. Currently, we just simulate real world behavior by assigning the probabilities according the following rules:

1) At intersections, the probability of turning to high ways doubles that of normal roads.

2) We require the sum of the probabilities to be 1 in total.

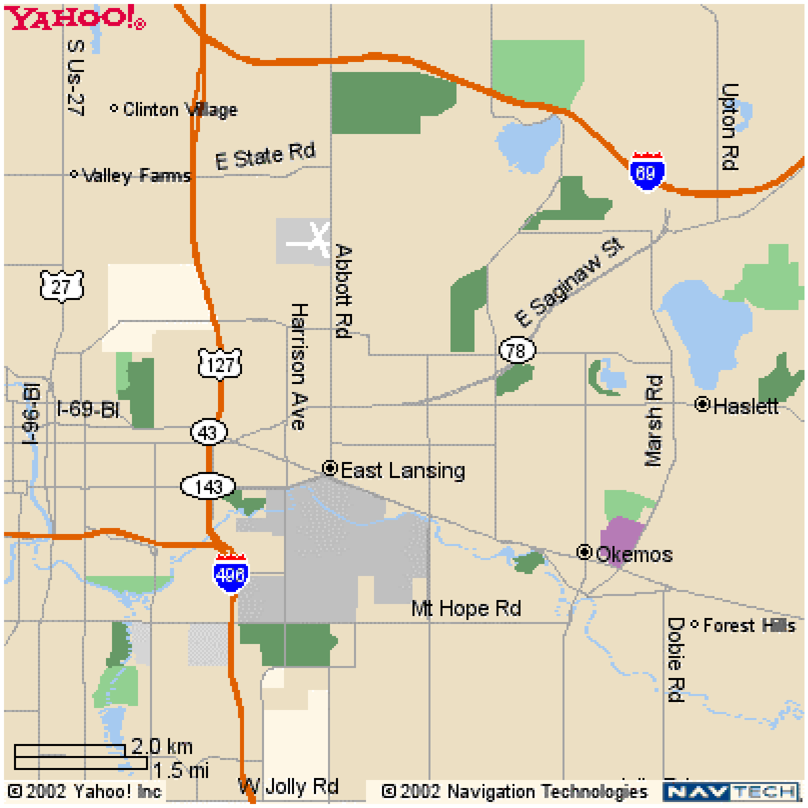

Figure 2.1 Map of East Lansing

4.3 Implementation issues.

1. Data representation

We use a 2-dimensional matrix to hold the information of every spot of the map. The unit of the map is pixel. Every cell of the matrix holds a structure as shown here.

```
struct{
      speed;
      number of branches;
      direction of each branch from the center;
      probabilities of each branch;}
```

The matrix is good enough to hold the information of the map. And it is obviously a sparse matrix.

2. Data production

Data file is obtained according to following routines.

1) Initialization: We put the vehicles at the border of the map. Of course the vehicles should be in a spot where a road runs through. Under this condition, the spot is given randomly.

2) Running: Each vehicle analyzes the information of the spot where it is located, and selects randomly the next branch it will go to. Of course at a spot of non-intersections, there is only one way to go.

3) Ending: Step 2 runs for a certain number of loops. During the course, when a vehicle reaches the border for a second time, we look on it as going out of the map. If in a certain loop, all the vehicles go out of the map, the program ends.

5. Simulation

We designed our simulator based on GloMoSim ([GLOMOSIM]) to simulate the prediction algorithms. GloMoSim is a scalable simulation environment for wireless network systems. It is designed using the parallel discrete-event simulation capability provided by a parallel extension of C, PARSEC ([PARSEC]). Some probabilistic motion models are also built in GloMoSim, one of which is random waypoint ([CN02]).

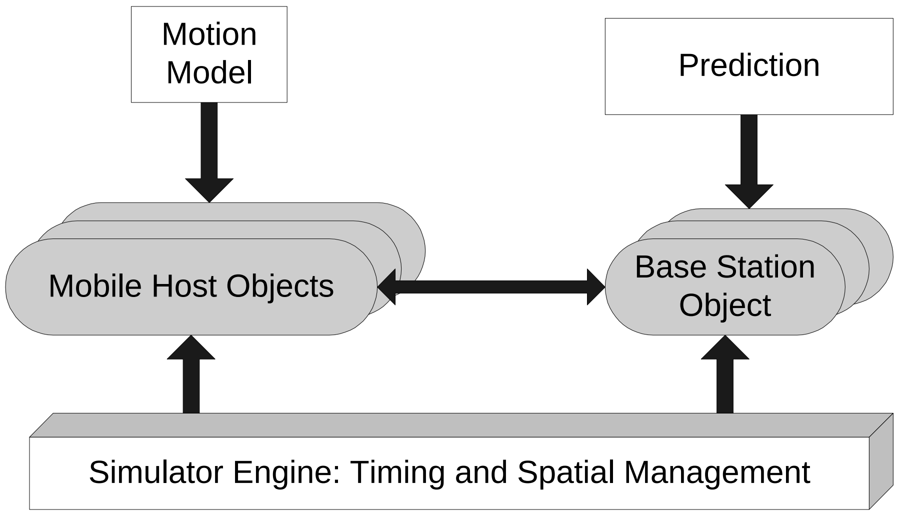

Figure 5.1 Simulator Architecture

The simulator architecture is shown in Figure 5.1. The simulation terrain is divided into equal area rectangles. Each rectangle represents a service cell. In each cell there is one and only one base station. The mobile hosts move according to the motion model. During simulation we changed the motion models to achieve more accuracy. Each base station object will execute the motion prediction algorithm and output the prediction result.

5.1 Simulation for the age based statistical solution

Two host motion models are used in the simulation. One is random waypoint model. The other is the geographical model. Details of the second model are dwelt on in the data generating part.

The parameters and results of the random waypoint mobility model are displayed in Table 5.1. Prediction successful rate is calculated by dividing the number of successful predictions by the number of predictions made totally.

Attributes	*Data*
Simulation Terrain Dimension	400×400
Number of Base Stations	100
Number of Nodes	4
Number of Motion Samples	2004
Overall Prediction Successful Rate	0.7189

Table 5.1 Random Waypoint Simulation Parameters and Results (Statistical Solution)

Figure 5.2 shows the accumulative and aggregated prediction successful rate for random waypoint model. For the accumulative prediction accuracy plot, the X axis denotes the accumulation sequence, which can be regarded as time flow. The Y axis denotes the accumulated predication successful rate, that is:

$$AccumulatedSuccessfulRate(x) = \frac{NumberOfSuccessful\,\mathrm{Pr}\,edictionsUntilx}{Numberof\ \mathrm{Pr}\,edictionsUntilx}$$

The trend line shows a steady increase of the prediction successful rate. The initial low prediction rate is due to the cache setup process. At last the trend line stables around 70%. This is the overall prediction successful rate.

For the aggregated prediction accuracy plot, the X axis denotes the aggregation sequence, which can be regarded as time flow. The Y axis denotes the aggregated predication successful rate, that is:

$$AggregatedSuccessfulRate(x) = \frac{NumberOfSuccessful\,\mathrm{Pr}\,edictionsDuring(x, x+1)}{Numberof\ \mathrm{Pr}\,edictionsDuring(x, x+1)}$$

The aggregated prediction successful rate has a big fluctuation. The initial aggregated prediction successful rate is also low due to the cache setup process.

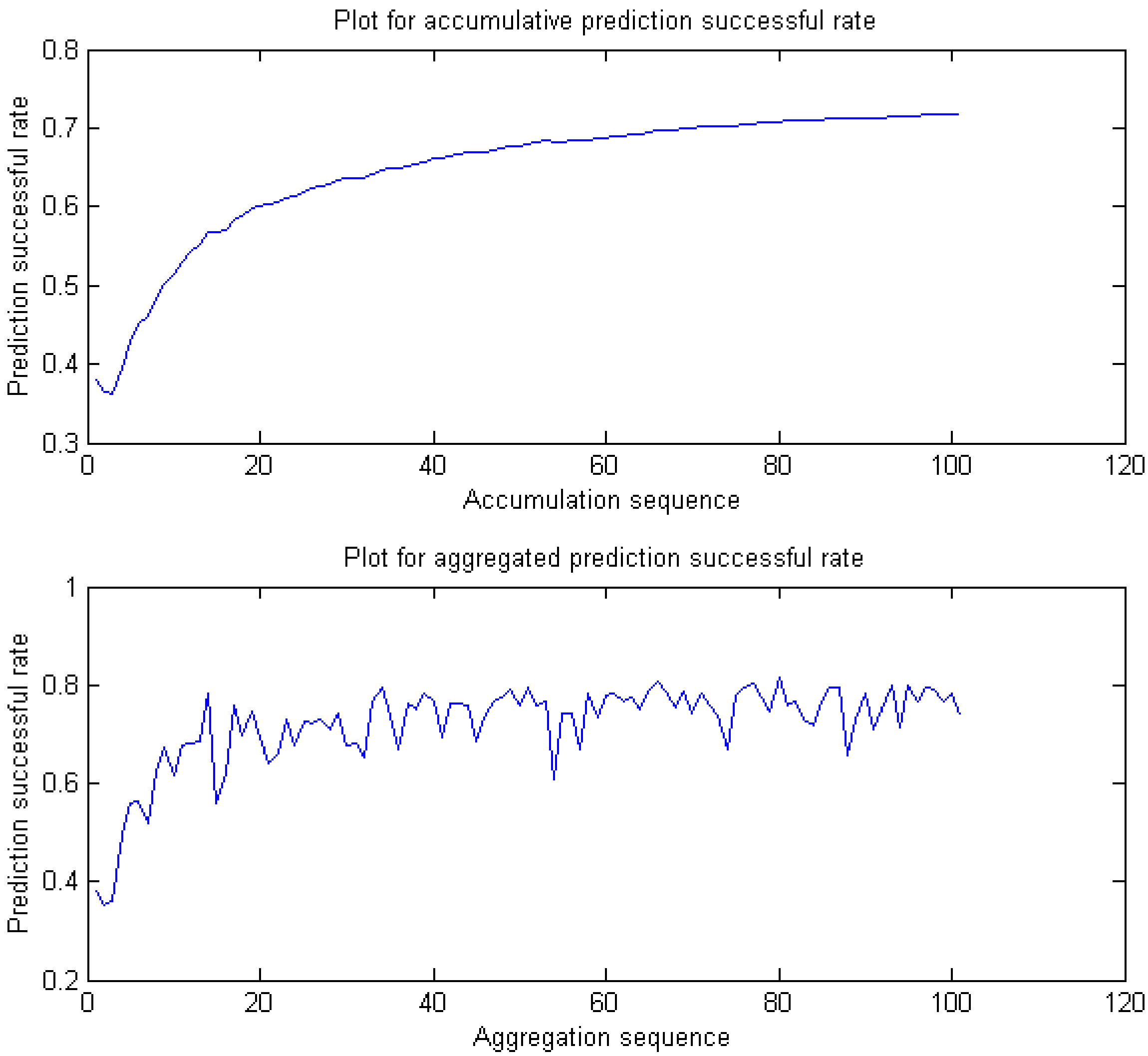

Figure 5.2 Accumulative and aggregated prediction successful rate for random waypoint model (Statistical Solution)

Attributes	*Data*
Simulation Terrain Dimension	400×400
Number of Base Stations	100
Number of Nodes	4
Number of Motion Samples	2000
Prediction Successful Rate	0.9308

Table 5.2 Geographical Model Simulation Parameters and Results (Statistical Solution)

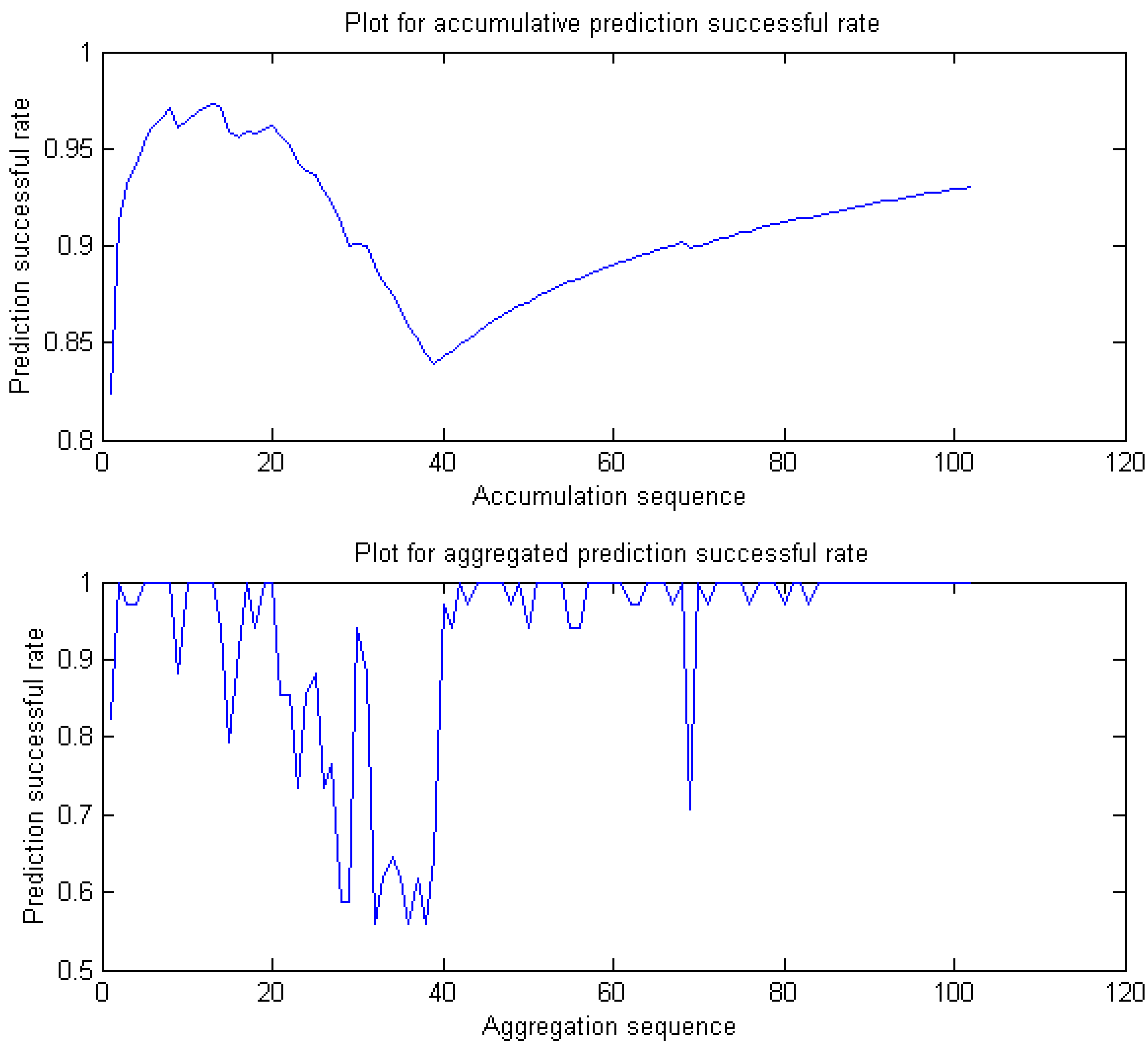

Figure 5.3 Accumulative and aggregated prediction successful rate for the geographical model (Statistical Solution)

Table 5.2 shows the geographical model simulation parameters and results, and Figure 5.3 shows the accumulative and aggregated prediction successful rate for the geographical model. Because geographical information is introduced into the mobility data, a higher prediction successful rate is achieved. However fluctuation still exists. The big dropdown in the accumulative prediction successful rate around 40 might indicate that the mobile host is having a rough turn from its original route.

5.2 Simulation for the neural network prediction solution

We simulated the neural network prediction solution by a smaller data size (as shown in Table 4.3). Due to the limitation of time and the computing facility we have, the data set has a very small size here. A larger data size will need more training time, while in real system this training can be done before hand other than in the actual prediction process. Here only the geographical model is used. The reason is that the geographical model reflects the reality better and neural network could possibly extract more parameters from this model.

Attributes	*Data*
Simulation Terrain Dimension	400×400
Number of Base Stations	100
Number of Nodes	4
Number of Motion Samples	400
Prediction Successful Rate	0.9167

Table 5.3 Geographical Model Simulation Parameters and Results (Neural Network Solution)

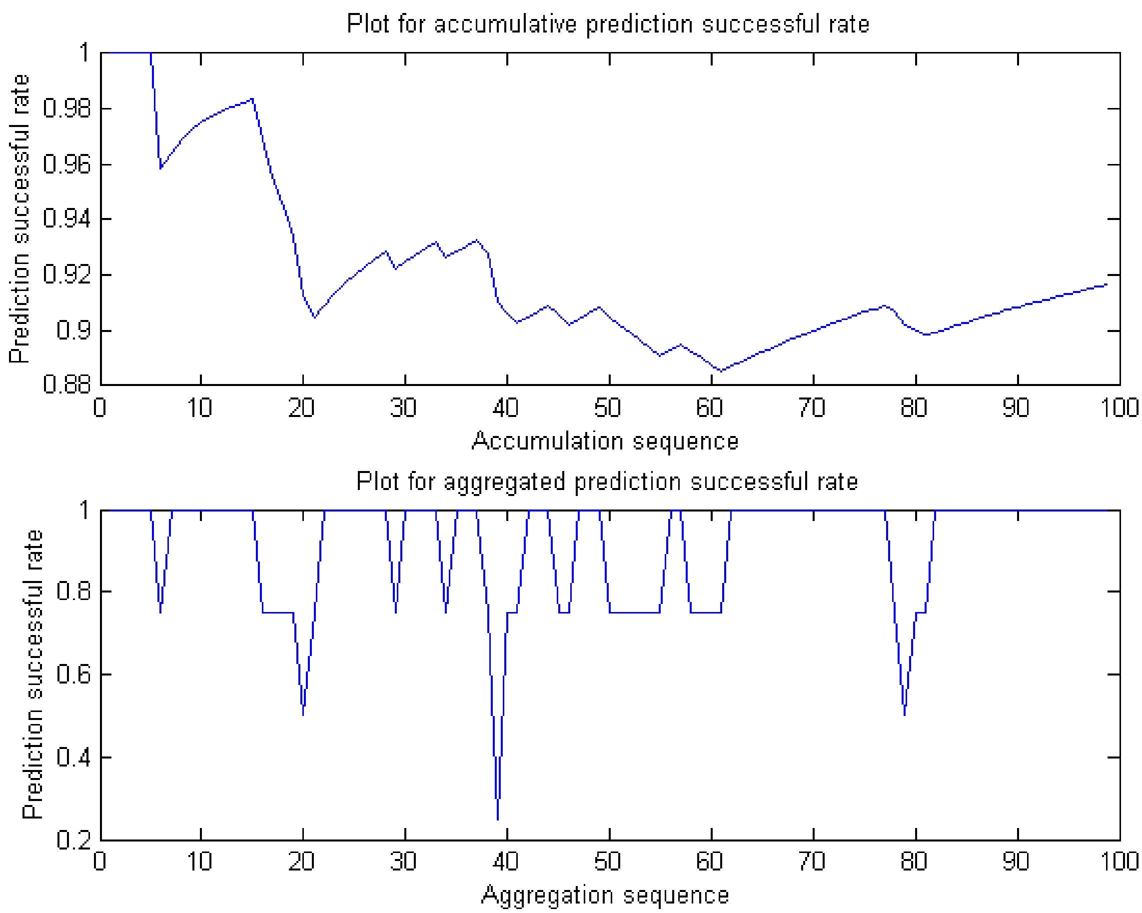

Figure 5.4 Accumulative and aggregated prediction successful rate for the geographical model (Neural Network Solution)

Figure 5.4 shows the accumulative and aggregated prediction successful rate for the geographical model. The results suggest that the whole prediction process of ANFIS achieves a rather higher successful rate. Some fluctuation exists in successful rate, which might be caused by noise data.

5.3 Comparison of the two methods

From the above results, we can find statistical method has a better average prediction successful rate (more than 93%) than ANFIS method (about 92%). However, the result in ANFIS method is smoother than that of the statistical method. For the statistical method, the accumulative successful rate in the worst case is about 82%, while for the ANFIS method it is over 88%. The cache method is more vulnerable to changes in short time, such as a big turn on the road. This is because of the limit of the cache size, and hence the cache needs a certain period of time to accumulate historical information. ANFIS can take advantage of off-site training, so that the characteristics of the mobile system can be grasped more accurately. Training is both a time and computing resource consuming process. That is the reason why a much smaller data set is used in the simulation process for the neural method in this paper. In real systems, however, off-site training is not a curse as long as the gaining of prediction accuracy outweighs its price.

Compared with other methods based on probabilistic methods ([CS98], [LAN95]), both of the two methods are easier to implement. And since there is not any probabilistic assumptions in the two models, they can be more adaptable to changes in the real environment.

6. Conclusion

In this study we explored the motion prediction techniques of mobile communication system. This study aims to improve the quality of service for mobile computing environments. An age based statistical method and a neural network method are presented and their performance is evaluated through simulation. Neural network solution is a promising direction. Further research might involve selecting better neural network system or improving the training efficiency.

References

[BEK97] Jens Biesterfeld, Elyes Ennigrou, Klaus Jobmann 1997
Neural Networks for Location Prediction in Mobile Networks

[CN02] Tommy Chu, Ioanis Nikolaidis: On the Artifacts of Random Waypoint Simulations. International Conference on Internet Computing 2002: 69-76

[CS98] S. Choi and K.G. Shin, “Predictive and Adaptive Bandwidth Reservation for Hand-Offs in QoS-Sensitive Cellular Networks”, Proceedings of ACM SIGCOMM ’98, Vancuver, BC, Oct. 1998, p.155-166.

[GLOMOSIM] http://pcl.cs.ucla.edu/projects/glomosim/

[Jang93] J.-S. R. Jang , ANFIS: Adaptive Neuro-Fuzzy Inference System, 1993

[LAN95] D.A. Levine, I.F. Akyildiz and M. Naghshineh, “The Shadow Cluster Concept for Resource Allocation and Call Admission in ATM-Based Wireless Networks”, Proceeding of ACM/IEEE MOBICOM’95, Berkeley, CA, Nov. 1995, p142-150.

[MANET] http://www.ietf.org/html.charters/manet-charter.html

[MW00] MathWorks Inc. ,”Fuzzy Logic Tool Box User’s Guide” 2000

[NS] http://www.isi.edu/nsnam/ns/

[PARSEC] http://pcl.cs.ucla.edu/projects/parsec/