Prediction on Time Series Data of DMV Office

Ye Zhang

***Abstract*—In California Department of Motor Vehicles(DMV), people sometimes have to waiting a long time to get services. It is necessary to anticipate future waiting time based on history so that people can choose a better time and a better nearby DMV office. This paper evaluates future waiting time prediction methods: hidden Markov model, dynamic Bayesian network, Markov chain model, n-gram language model, decision tree and support vector machine(SVM). We calculate the average prediction error in hours to compare these different models.**

***Keywords*-component; formatting; style; styling;**

I. Introduction

Can future waiting time be predicted based on previous waiting time in DMV office? Normally, waiting time in an office follows some regular patterns. For example, if there is long queue at 11:00 am in a certain DMV, there might be also a long queue at 11:10 am in this office. Also, we observe that people prefer to go to DMV late in the morning or early in the afternoon rather than early in the morning or late in the afternoon. So if a person asks when he should go to DMV best, we might suggest that he should choose early in the morning or late in the afternoon. However, sometimes, people are not available in these time buckets and they have to go to DMV during peak hours. In this case, we have to think of some methods to predict waiting time at a certain time in the future when people want to go to DMV to better satisfy their needs.

In this paper, we predict future waiting time based on previous waiting time and corresponding time buckets. We use ”DMV office waiting time” dataset. In this dataset, we have waiting time in 176 DMV offices in recent 3 months. The waiting time for each office is sampled one time every 10 minutes, so we use 10 minutes as a time bucket. For example, between 8:00 am and 9:00 am, we have 8:00–8:10, 8:10–8:20, 8:20–8:30, 8:30–8:40, 8:40–8:50, 8:50–9:00 these time buckets, and other time is similar. Several prediction techniques are proposed in this paper including hidden Markov model(HMM), dynamic Bayesian model, Markov chain model, n-gram language model, decision tree and SVM. We split the whole dataset into two parts with equal size. One is used for training and the other is used for testing. Continuous waiting time in training dataset is discretized into 20 discrete values which we use to denote different states. This is obtained by drawing equal height histogram for all waiting time values for the office. In this histogram, we have 20 intervals and in each bucket, we have the same number of values of waiting time. Then we use the median value of each interval to denote the discrete value for all the waiting time values that in this interval.

The main criterion for comparison is the average prediction error. After we use different models to train the dataset, we predict a discrete value for a given future time bucket in the testing data and then convert this discrete value into corresponding continuous waiting time and compare this prediction value with the true waiting time. Finally, we compute average error over all testing data.

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

II. Related Work

Some previous work has involved time series prediction. has used HMM for modeling time series data.

III. Prediction Models

We developed and evaluated several statistical models to predict future waiting time without appointment at a certain office. Since we sampled data each 10 minutes, we use 10 minutes as a time bucket. We split the whole dataset into two parts with equal size and use one for training and the other for testing. In order to better visualize waiting time distribution, we choose three offices randomly among all the offices and fit a Gaussian distribution for each time bucket for each of these three offices. We can see the distribution from Figure 1, Figure 2 and Figure 3. Then we take into account waiting time values of all offices for each time bucket and again fit a Gaussian distribution for each time bucket.

A. Time-based Waiting Time Prediction

We also draw figures for another 5 offices for visualization purposes and we don't show them here. Observing the results, we can claim that the waiting time distribution of different office is similar and we decide to randomly select a office for research purpose. The office we choose is 632.

B. Time-based Waiting Time Prediction

first model used in our work is the time-based waiting time prediction model. This model uses only the time feature to predict future waiting time. and fit a Gaussian distribution for each time bucket. The following is the error bar distribution for DMV office 632 whole complete day. We use the The average error over testing data is 0.5235 hour.

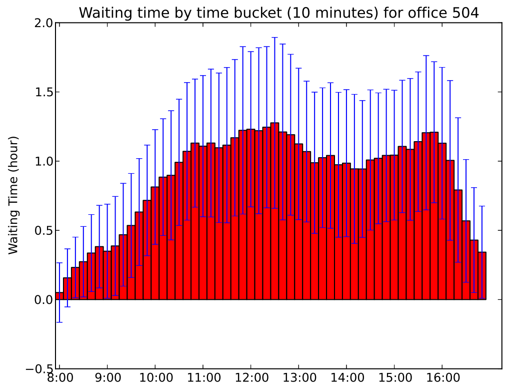

Figure 1. Waiting Time Distribution for Office 504

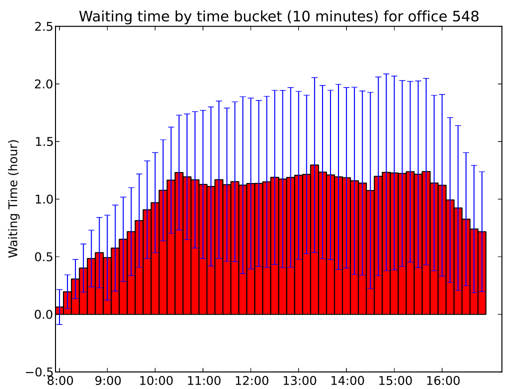

Figure 2. Waiting Time Distribution for Office 548

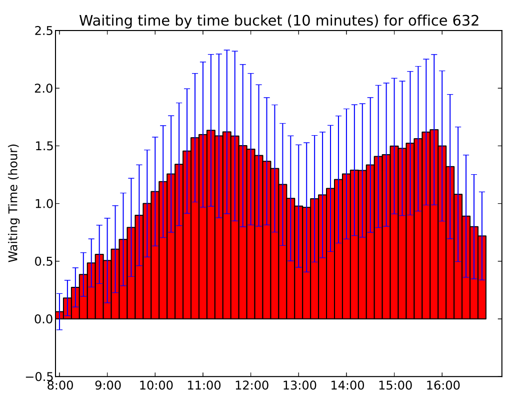

Figure 3. Waiting Time Distribution for Office 632

C. HMM Model

We use HMM to model time series. First we use MLE on training data to obtain transition probability, emission probability and initial probability. Then we use Viterbi path algorithm to do inference on waiting time sequence given us a time sequence. And we use this inferred waiting time sequence as prediction to calculate the error. The error we obtained is 0.2981 hour.

D. Danymic Bayesian Network

We also use Danymic Bayesian Network to model the time series. The whole procedure is similar as in HMM model. But this time, we use weekday plus time of day these two as two variables. So in total, there are three variables: waiting time, time of day, and weekday. We use MLE to calculate necessary parameters including transition probability, emission probability and initial probability. Then we use Viterbi algorithm to do inference on hidden sequence given a time sequence in testing data. The average error we obtained is 0.7340 hour. So we can claim that after treating weekday as a feature, the prediction error will increase compared to HMM.

E. One Order Markov Chain Model

Actually, when we want to know waiting time at a given time bucket t, we always know waiting time at t-1plus prior knowledge about waiting time at time bucket t. Specifically, based on these historical knowledge, we want to calculate

$$W_t = \arg\max_{W_t} P(W_t|W_{t-1}, H_t) \tag{1}$$

where W_t denotes the waiting time at time time bucket t, and H_t denotes the time bucket t. We first use Bayes rule to convert this formula to the following form assuming that H_t is independent from transition probability : We use Bayes rule to convert this formula into the following:

$$W_t = \arg\max_{W_t} P(W_{t-1}|W_t) * P(W_t|H_t) \tag{2}$$

We use MLE to calculate probability $P(W_{t-1}|W_t)$ and $P(W_t|H_t)$, and find the maximum W_t for each given H_t and W_{t-1} in the testing data and use this maximum as prediction. The average error we obtain is 0.1629 hour.

F. 4-Gram Language Model

In practice, when we want to predict next waiting time, we know not only previous moment's waiting time, but also previous waiting time of several time buckets. So we try 4-Gram Language Model where we model each complete time sequence as a sentence in 4-Gram model. The task is to predict a word (waiting time) at a location given previous three words (three time buckets' waiting time). Then we use back-off model to do prediction on testing data. Specifically, we want to calculate:

$$W_t = \arg\max_{W_t} P(W_t|W_{t-1}, W_{t-2}, W_{t-3}, H_t) \tag{3}$$

In order to calculate W_t, we decide to use two different methods.

Table I
ERROR FOR DIFFERENT λ FOR LOG LINEAR 4-GRAM LANGUAGE MODEL

λ	0.3	0.5	0.8
Error(hour)	0.2032	0.1674	0.1651

1) Log Linear Probability Model: The first one is called log linear probability model.

$$W_t = \arg\max_{W_t} \lambda * \log P(W_t|W_{t-1}, W_{t-2}, W_{t-3}) + (1-\lambda) * \log P \quad (4)$$

We use λ to denote the importance of previous waiting time. That is, the larger λ is, the more a certain time waiting time depends on historical waiting time. The average errors for different λ are as the following:

We can see from the result that the previous waiting time plays a more important role in deciding current waiting time.

2) Reversed 3-order Markov Model:

$$t-3, H_t)$$

$$= \arg\max_{W_t} P(W_t, W_{t-1}, W_{t-2}, W_{t-3}) = \arg\max_{W_t} P(W_t|H_t) * P(W \quad t-1, W_t) \quad (5)$$

We reverse the waiting time sequence to W_n, W_{n-1}, $\ldots W_3$, W_2, W_1 and train an n-gram The error of this model is 0.1698 hour. From the above two methods, we can conclude that prediction is less accurate if we consider longer previous time sequences compared to the situation when we only know one previous bucket's waiting time.

G. Distance N-Gram Model

In real life, we often want to know the waiting time after 30 minutes or even 1 hour which means that we lack an interval of waiting time between the current time bucket and a future time bucket which we are concerned. In this case, we also use MLE to training transition probability $P(W_t|W_{t-gap})$ and we try three different methods to predict future waiting time.

1) Using transition probability only: First, we only use transition probability without considering prior knowledge on future time. So future waiting time only depends on previous waiting time.

$$W_t = \arg\max_{W_t} P(W_t|W_{t-gap}) \quad (6)$$

We draw the plot for the prediction error versus time gap which is shown in 4. The range of the time gap is 20 minutes to 180 minutes (3 hours). From the figure, we can see that if the gap is more than 1 hour, the prediction is no better than direct prediction with prior knowledge.

2) Using transition probability plus prior knowledge and using Bayes rule: The second method we use add prior

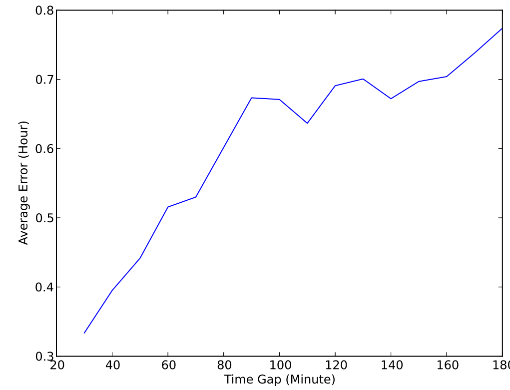

Figure 4. Prediction Error of Distance N-Gram Model Using Transition Probability Only

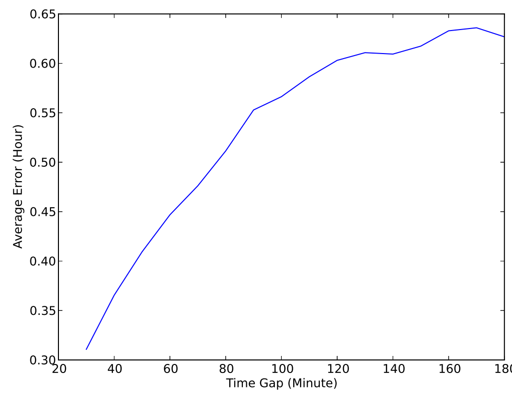

Figure 5. Prediction Error of Distance N-Gram Model Using Transition Probability Plus Prior Knowledge

knowledge which is similar to the situation without gap.

$$W_t = \arg\max_{W_t} P(W_t|W_{t-gap}, H_t = \arg\max_{W_t} P(W_{t-gap}|W_t) * P(W \quad (7)$$

The result is shown in 5 The prediction is better than pure prediction without prior knowledge.

3) Using transition probability plus prior knowledge and using log linear probability model: We also try log linear probability model to adjust different weights on prior knowledge and previous waiting time.

$$W_t = \arg\max_{W_t} P(W_t|W_{t-gap}, H_t = \arg\max_{W_t} (1-\lambda) * \log P(W_t|W \quad (8)$$

Here, we also try different λ to give different weights to prior knowledge. We use three methods to set λ. Firstly, we just let $\lambda = 0.5$ giving equal weight on prior knowledge and previous knowledge. The result is shown in Figure 6. The result is similar to the

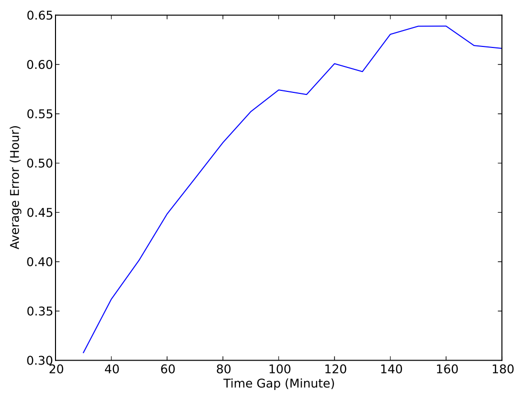

Figure 6. Log Linear Model with λ being 0.5

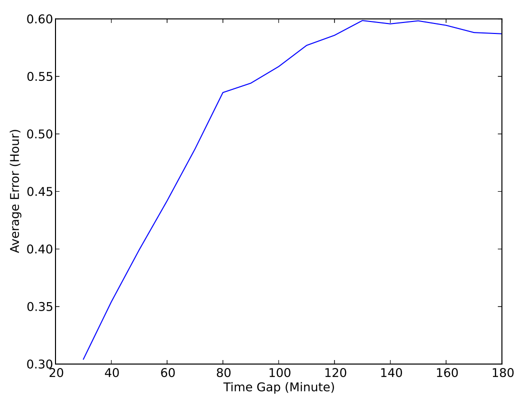

Figure 7. Log Linear Model with λ Linear to Time Gap

Bayes rule. Second, we let $\lambda = 0.5 + gap * 0.5/18$ so that the larger the time gap is, the more weight we put on the prior knowledge about time bucket t. The result is shown in FIgure 7. We obtain a much better result setting λ linear to time gap compared to constant equal λ. We can conclude that as the gap becomes larger, prior knowledge plays a much more important role in prediction. Motivated by this, we make a step further by letting λ exponential to time gap. We let $\lambda = 1 - 0.5^{gap}$ to put even more weight on prior knowledge when the gap becomes larger. And the corresponding prediction error is shown in Figure 8. As we can see from this figure, when the gap is small, we'll get high error rate if we put overly much weight on prior knowledge which accords with our common sense. But as the gap become larger, the effect is very similar to the case when λ is linear to the gap.

H. Classification Model

We can also model this time prediction problem as classification problem. As above, we discretize the continuous

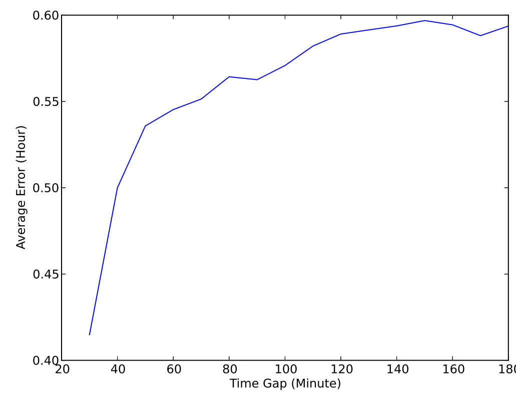

Figure 8. Log Linear Model with λ Exponential to Time Gap

waiting time into 20 values and we treat these 20 values as 20 different classes. Then we use Decision Tree and Support Vector Machine(SVM) to solve the problem with time bucket and previous waiting time as features. This time, we compute the percentage of correctly classified data points in testing data. The results are:

Acknowledgment

Acknowledgements to be added to camera ready version upon acceptance.