Wait Time Prediction: How to Avoid Waiting in Lines?

Ye Zhang  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA 15213, USA  
yezhang@andrew.cmu.edu

Le T. Nguyen  
Carnegie Mellon University  
Moffet Field, CA, USA  
ze.nguyen@sv.cmu.edu

Joy Zhang  
Carnegie Mellon University  
Moffet Field, CA, USA  
joy.zhang@sv.cmu.edu

Abstract
One of the challenges of organizations providing services to the public is the effective resource allocation. Many service providers such as hospitals, city halls or department of motor vehicles suffer from a service demand, which is unevenly distributed over the day. In this work, we evaluate techniques for predicting the service demand. We use the wait time dataset collected from the websites of California Department of Motor Vehicles (DMV). We extract patterns of the service demand in form of wait time during each hour of a day and each day of a week. This information is used to train multiple machine learning models in order to predict the future wait time at DMV offices.

Author Keywords  
Time Series Prediction, Machine Learning

ACM Classification Keywords  
I.2.6 [Artificial Intelligence]: Learning.

General Terms  
Time Series Prediction, Machine Learning

Introduction  
One of the challenges of organizations providing services to the public is the effective resource allocation. Many service providers such as hospitals, city halls or department of motor vehicles suffer from a service demand, which is unevenly distributed over the day. In this work, we evaluate techniques for predicting the service demand. We use the wait time dataset collected from the websites of California Department of Motor Vehicles (DMV). We extract patterns of the service demand in form of wait time during each hour of a day and each day of a week. This information is used to train multiple machine learning models in order to predict the future wait time at DMV offices.
allocation. Many service providers such as hospitals, city halls or department of motor vehicles suffer from a service demand, which is unevenly distributed over time. For example in city halls, people have to wait hours in a line when coming at the peak hours even though there are time periods during a day when the clerks have no one to serve. Due to the resource constraints (e.g. space availability in the offices) the issue cannot be easily addressed by only adjusting the number of workers for a specific time of a day.

In the recent years, many organizations aimed to smoothen the peak in demands by making information about the current wait time available online [4, 5, 3]. The goal is to make people aware of the peak hours, so that they can avoid them. However, the current wait time information is rather obsolete as people rather need to know the expected wait time at a certain time in the future (e.g. in 3 hours or in the next days).

In this work, we evaluate approaches for predicting the expected wait time given a historical data. We use the real-world data set collected from the website of the California Department of Motor Vehicle (DMV). DMV provide many services related to obtaining and maintaining a driver license and many vehicle related services such vehicle registration etc. California with a population of 37.5 million people has 24 million registered drivers owning in total 31 million vehicles [2]. This results in a large amount of people visiting DMV everyday especially at the peak hours. Based on our analysis a wait time duration at certain offices might differ as much as 3 hours, i.e., an “unlucky” customer might have to wait 3 hours more than an informed customer.

In this work, we frame the wait time prediction as a time series prediction problem, where we aim to predict a future data point given the data in the past. Time series prediction approaches have been widely used in various applications such as stock market prediction [6], traffic prediction [8] or user activity and location prediction[7]. In this work, we use these approaches to answer basic questions such as 1) What is the best time in the week to go to DMV, 2) When should one go to DMV today given the wait time observed today or 3) Given a location of the user and the current time, which DMV should the user go to? We evaluate multiple prediction algorithms and report the results of the prediction accuracy.

**Dataset**

There are 176 DMV offices in California spread over the whole state (as shown in Figure 1). The wait time dataset (http://mlt.sv.cmu.edu/joy/publications/dmvq.zip) was collected by scraping the DMV websites every 10 minutes during the office hours on weekdays. We assume that the raw wait time information was estimated based on the number of outstanding tickets and an average duration of one ticket being processed at a counter and the number of open counters. We collected data for 3 months resulting in total 0.7 million data records.
Figures 2, 3 and 4 show the average and the standard deviation of wait time for three DMV offices. As we can observe from the figures, there are some patterns common for all three offices. For example the wait time starts in the morning from 0, increases over time and peaks at around 11am. However, we can also see that there are some characteristics specific to certain offices. For example for the Santa Clara office we can see two peaks, one at 11am and the second one around 4pm as people leave the work early to go to DMV. One the other hand, for the San Jose office we can see that the wait time does not significantly vary during 11am and 4pm.
Figure 4: Average and standard deviation of wait time for the Los Gatos DMV office

Figure 5 shows the location of the three DMV offices described above. These offices are about 10 miles away from each other. Thus, based on this information and the current location of a user, we can make a suggestion which office a user should go to at a given time. For example, around 11am the San Jose office has a shortest line, but at 2am we would recommend a customer to visit the Los Gatos office.

Figure 6 shows the cumulative distribution function (CDF) of wait time in the Santa Clara office. As we can see around 50% of the time a customer will have to wait 1 hour or more. In 10% of the cases a customer would have to wait even 2 to 3 hours in line.

Figure 6: CDF of wait time in Santa Clara office.

The wait time does also significantly differ between different days in the week. Figures 7 and 8 show the difference of wait time between busy Mondays and less-busy Wednesdays, which confirms the observations made in [1]. From the results we can observe that at the peak time the difference of wait time could be 30 minutes or more.

Figure 7: Wait time on Monday for the Santa Clara DMV office
Prediction
The goal in this work is to predict a future wait time at DMV offices. We evaluate a large number of prediction algorithms including Hidden Markov model (HMM), Dynamic Bayesian model, Markov chain model, n-gram language model, decision tree etc.

We split the dataset into two parts with equal size. One is used for training and the other is used for testing. Since we use multiple algorithms that can process only categorical data, we discretize the continuous wait time resulting into 20 values. This is obtained by creating an equal-height histogram for all wait time values of one office. In this histogram, we have 20 intervals and in each histogram interval we have the same number of wait time elements. Then we use the median value of each interval to represent the discrete value for all the wait time values that in this interval.

Time-based Prediction
The first model used in this work is the time-based prediction model. For each time of the week we compute the average wait time and use this value for the prediction. For example, in the training data set we find all wait time values on Monday 8:00am and average these values. In the test dataset we find all occurrences of Monday 8:00am, use the average from the training dataset for the prediction and compute the prediction error. With this approach we achieved an average prediction error of 0.5235 hours.

HMM Model
In the second approach we use HMM to model time series. The observation value is the time of day, and the related hidden value is the corresponding wait time. In HMM we want to predict the sequence of wait time values given a sequence of observed time of the day. First we use MLE on training data to learn the state, transition and emission probabilities. Then we use Viterbi algorithm to do inference on wait time sequence given a time sequence. And we use this inferred wait time sequence as prediction and calculate the error. The average prediction error for this approach is 0.2981 hour.

Dynamic Bayesian Network
Dynamic Bayesian Network (DBN) can be used to for the time series prediction. For this approach we use the information of day of the week and time of day as two separate variables. Thus, we have three variables: wait time, time of day, and day of the week. The observations of DBN are time of day and the weekday, and the related hidden node is the corresponding wait time. We use MLE to learn the parameters from the training data set. Then we use Viterbi algorithm to do inference on hidden sequence given an observed time sequence in testing data. The average error we obtained is 0.7340 hour.

Figure 8: Wait time on Wednesday for the Santa Clara DMV office

Figure 8: Wait time on Wednesday for the Santa Clara DMV office
One Order Markov Chain Model
Actually, when we want to know wait time at a given
time bucket $t$, we always know wait time at $t-1$ plus prior
knowledge about waiting time at time bucket $t$.
Specifically, based on these historical knowledge, we
want to calculate:

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

where $W_t$ denotes the wait time at time $t$, 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:

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

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 $W_{t-1}$. This value is used as the prediction. The
average error of this approach is 0.1629 hours.

4-Gram Language Model
In practice, when we want to predict wait time at a
certain time $t$, we know not only one previous bucket's
wait time, but also wait time of several previous time
buckets. Motivated by this, 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 (wait time) at a location given previous three
words (three time buckets’ wait time). Then we use
back-off model to do prediction on testing data.
Specifically, we want to calculate:

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

In order to calculate $W_t$, we have 2 options discussed
in the following.

Log-Linear Probability Model
The first approach is called log-linear probability model:

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

We use $\lambda$ to denote the importance of previous wait
time. The larger is $\lambda$ the more is the current wait time
dependent on historical wait time. The average error
for different $\lambda$ are as the following:

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>Error (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>0.2032</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1674</td>
</tr>
<tr>
<td>0.8</td>
<td>0.1651</td>
</tr>
</tbody>
</table>

We can see from the results that the previous wait time
plays an important role in the estimation of the current
wait time.

Reversed 3-order Markov Model
In the second approach we reverse the wait time
sequence to $W_n, W_{n-1}, \ldots W_3, W_2, W_1$ and train an
$n$-gram:
\[ W_t = \arg \max_{W_t} P(W_t|W_{t-1}, W_{t-2}, W_{t-3}, H_t) \]
\[ = \arg \max_{W_t} P(W_t, W_{t-1}, W_{t-2}, W_{t-3}) \]
\[ = \arg \max_{W_t} P(W_t|H_t) \times P(W_{t-1}|W_t) \times P(W_{t-2}|W_t, W_{t-1}) \times P(W_{t-3}|W_{t-2}, W_{t-1}, W_t) \]

The error of this model is 0.1698 hour. From the above two results, we conclude that prediction is less accurate if we consider longer historical time sequences compared to the situation when we only know one previous wait time.

**Distance N-Gram Model**

In real life, we often want to predict a wait time in 30 minutes or in even 1 hour. This means that we might not necessarily have the information about the wait time between the current time and a future time, which is to be predicted. In this case, we use MLE to train the transition probability \( P(W_t|W_{t-gap}) \) by using one of the following three different methods.

1. **Using transition probability only:** First, we only use transition probability without considering prior knowledge on future time. Thus, future wait time only depends on the previous wait time.

   \[ W_t = \arg \max_{W_t} P(W_t|W_{t-gap}) \]

   Figure 9 shows the prediction error in relation to the time gap. The range of the time gap is 20 minutes to 180 minutes (3 hours). From the figure we can observe that if the gap is longer than 1 hour, the prediction is no better than direct prediction with the prior knowledge.

2. **Using transition probability plus prior knowledge and using Bayes rule:** The second method we use adds prior knowledge, which is similar to the situation without gap.

   \[ W_t = \arg \max_{W_t} P(W_t|W_{t-gap}, H_t) \]

   \[ = \arg \max_{W_t} P(W_{t-gap}|W_t) \times P(W_t|H_t) \]

   The result of this approach is shown in Figure 10. The prediction is better than pure prediction without prior knowledge.
3. Using transition probability plus prior knowledge and using log linear probability model: The last option is to apply the log linear probability model to adjust different weights on prior knowledge and previous wait time.

\[ W_t = \arg \max_{W_t} P(W_t|W_{t-gap}, H_t) \]
\[ = \arg \max_{W_t} (1 - \lambda) \log P(W_t|W_{t-gap}) + \lambda \log P(W_t|H_t) \]

We use three methods to set \( \lambda \). First, we set \( \lambda = 0.5 \) giving equal weight on the prior knowledge and the previous knowledge. The result is similar to the Bayes rule (as shown in Figure 11).

Second, we set \( \lambda = 0.5 + \text{gap} \times 0.5/18 \) so that the larger the gap, the more weight we put on the prior knowledge. The result is shown in Figure 12.

We obtain a better result by setting \( \lambda \) linear to time gap compared to a constant \( \lambda \). We can conclude that as the gap becomes larger, prior knowledge plays a more important role in the prediction.
prediction. Motivated by this, we make a step further by setting \( \lambda \) exponential to time gap. We set \( \lambda = 1 - 0.5^{\text{gap/20}} \) to put more weight on prior knowledge when the gap becomes larger. And the corresponding prediction error is shown in Figure 13.

![Figure 13: Log Linear Model with \( \lambda \) Exponential to Time Gap](image)

As we can see from the results, when the gap is small, we get high prediction error if we put a high weight on prior knowledge. However, as the gap become larger, this effect is very similar to the case when \( \lambda \) is linear to the gap.

**Classification Model**

The time series prediction can be framed as a classification problem. As above, we discretize the continuous wait time into 20 values and we treat these 20 values as 20 different classes. Then we use Decision Tree and Support Vector Machine (SVM) predict future wait time given the features corresponding to previous wait time, current time of the day and day of the week. The average prediction error of Decision tree is 0.2592 hour and of SVM is 0.1636 hour.

**Linear Regression**

Linear regression can be used to solve the time series prediction problem. Assuming, we know all the previous wait time and want to predict the next wait time. We assume that the wait time at time \( t \) is linear with respect to all the previous wait time value. Thus, we focus on the linear relationship:

\[
W_t = a_{t-1} \times W_{t-1} + a_{t-2} \times W_{t-2} + \cdots + a_1 \times W_{1}
\]

where \( W_t \) is the wait time at time \( t \) that we want to predict, and \( W_{t-1}, W_{t-2}, \ldots W_{1} \) are all the previous wait time. \( a_{t-1}, a_{t-2}, \ldots a_1 \) are the coefficients learned from the wait time history. The average prediction error achieved with this approach is 0.1582 hour.

When we consider the case with time gap, we use the following equation:

\[
W_t = a_{t-gap} \times W_{t-gap} + a_{t-gap-1} \times W_{t-gap-1} + \cdots + a_1 \times W_{1}
\]

The average prediction error with the gap is shown in Figure 14.

![Figure 14: Linear Regression with Time Gap](image)
**Prediction errors**

Table 1 summarizes the results achieved by all the algorithms evaluated in this work. From the results we can observe that the simple linear regression model seems to be performing the best. Thus, by considering only a small number of the past wait times we can achieve the best prediction.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Prediction error</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBN</td>
<td>0.7340</td>
</tr>
<tr>
<td>Time-based</td>
<td>0.5235</td>
</tr>
<tr>
<td>HMM</td>
<td>0.2981</td>
</tr>
<tr>
<td>Decision tree</td>
<td>0.2592</td>
</tr>
<tr>
<td>4-Gram Language (Reversed Markov)</td>
<td>0.1698</td>
</tr>
<tr>
<td>4-Gram Language (Log Linear)</td>
<td>0.1651</td>
</tr>
<tr>
<td>SVM</td>
<td>0.1636</td>
</tr>
<tr>
<td>One Order Markov</td>
<td>0.1629</td>
</tr>
<tr>
<td>Linear Regression</td>
<td>0.1582</td>
</tr>
</tbody>
</table>

**Table 1:** Prediction error of each algorithm (in hours).

**Acknowledgments**

This work is supported in part by NSF project CNS1228813: Capturing People’s Expectations of Privacy with Mobile Apps by Combining Automated Scanning and Crowdsourcing Techniques.

**References**


