Tourism Demand Forecasting Using Machine Learning Methods

Nesreen K. Ahmed  
Faculty of Computers and Information  
Cairo University, Giza, Egypt  
n.kamel@gmail.com

Amir F. Atiya  
Dept Computer Engineering  
Cairo University, Giza, Egypt  
amir@alumni.caltech.edu

Neamat El Gayar  
Faculty of Computers and Information  
Cairo University, Giza, Egypt  
hmg@link.net

Hisham El-Shishiny  
IBM Center for Advanced Studies in Cairo  
IBM Cairo Technology Development Center  
Giza, Egypt  
shishiny@eg.ibm.com

Abstract
Tourism demand forecasting has attracted the attention of researchers in the last decade. However, most of research focused on traditional quantitative forecasting techniques, such as ARIMA, exponential smoothing, etc. Although these traditional methods have achieved certain levels of success in the tourism research, it would be useful to study the performance of alternative models such as machine learning methods. This is the topic considered in this paper. The goal is to investigate how different machine learning models can be applied in the tourism prediction problem and to assess the performance of seven well known machine learning methods. Furthermore, we investigate the effect of including the time index as an input variable. Specifically, we consider the tourism demand time series for Hong Kong inbound travel.

Keywords: Tourism Forecasting, Machine Learning.

1 Introduction
Tourism as an industry is of particular importance to many countries, being both a very significant source of foreign exchange as well as a provider of direct and indirect employment. The need for accurate tourism forecasting is thus particularly important due to the industry’s significant contribution to the economy [1]. In the tourism industry planning is particularly important because of the rapid economic and political changes, the perishable nature of the tourism industry’s products and services (e.g. unoccupied airline seats, and unused hotel rooms cannot be stockpiled for future use), and the sensitive nature of the industry to natural and human made disasters [2]. To a large extent, planning relies heavily on accurate forecasts in order to reduce the risk in the decision making process. That is to reduce the chances a decision will fail to achieve the desired objectives. The need of accurate forecasting is therefore important to government and managers. For example, government bodies use demand forecasts to set marketing goals, and explore potential markets. Managers use demand forecasts to determine operational requirements such as staffing and capacity, and study project feasibility such as the financial viability to build a new hotel. There exists a significant literature highlighting tourism demand forecasting, but most of it concentrates on traditional time series methods. For example, Lim and McAleer [3] use Box-Jenkins’ Autoregressive Integrated Moving Average (ARIMA) model to forecast tourist arrivals to Australia from Hong Kong, Malaysia and Singapore. Goh and Law [4] present the use of Seasonal Autoregressive Integrated Moving Average (SARIMA) and Multivariate Autoregressive Integrated Moving Average (MARIMA) time series models with interventions in forecasting tourism demand using ten arrival series for Hong Kong. However, to our knowledge, there no attempts have been made to investigate the performance of machine learning models in forecasting tourism demand except for neural network models.
Neural networks were first applied to forecast tourism demand by Law and Au [5] to forecast Japanese demand for travel to Honk Kong. Experimental results show that neural network model forecasts outperform multiple regression, moving average, and exponential smoothing. Law and Au [6] extend the applicability of neural networks in tourism demand forecasting by applying the multilayer perceptron to tourism demand data. This paper investigates how seven different machine learning models can be applied to the tourism prediction problem. The models we consider are: multi-layer perceptron, radial basis functions, generalized regression neural networks (also called kernel regression), K-nearest neighbor regression, support vector regression, CART classification and regression trees, and Gaussian processes. The rest of the paper is organized as follows. Section (2) presents a brief description of the compared models. Section (3) describes the details of the experimental setup. Section (4) presents the analysis of the results, and finally concluding remarks are provided in Section (5).

2 Models

For each considered machine learning model, there are several variations proposed in the literature, hence it would be difficult to consider all such variations. Therefore, we focus our action only on the basic versions of the models. Below is a brief description of the models considered.

2.1 Multilayer Perceptron Neural Network (MLP)

The multilayer perceptron (often simply called neural network) is perhaps the most popular network architecture in use today both for classification and regression. The network has a simple interpretation as a form of input-output model, with the weights and biases being the free parameters of the model. Such network can model functions of arbitrary complexity, with the number of hidden nodes determining the network complexity. As is typically the case in practice, we consider a network with one hidden layer. Different training algorithms can be used to set the weights and biases so as to minimize the error of the neural network, the most well-known of which is the backpropagation algorithm. A second order optimization method called Levenberg Marquardt is generally known to be more efficient than the basic backpropagation algorithm, and this is the one we use in our implementation (we use the Matlab function trainlm).

2.2 Radial Basis Function Neural Network (RBF)

The radial basis function network is similar in architecture to the multilayer network except that the nodes have a localized activation function (Powell [7], and Moody and Darken [8]). Most commonly, node functions are chosen as Gaussian functions, with the width of the Gaussian function controlling the smoothness of the fitted function. The outputs of the nodes are combined linearly to give the final network output. Because of the localized nature of the node functions, other simpler algorithms have been developed for training radial basis networks. The algorithm we use is the Matlab function newrb. It is based on starting with a blank network, and sequentially adding nodes until an acceptable error in the training set is achieved. Specifically, we add a node centered around the training pattern giving the maximum error. Then we recompute all the output layer weights using the least squares formula. We continue this way until the error limit is reached or the number of nodes reaches a maximum predetermined value. We select the maximum number to be 25 percent of the training set.

2.3 Generalized Regression Neural Network (GRNN)

GRNN is commonly called the Nadaraya-Watson estimator or the kernel regression estimator (Nadaraya [9] and Watson [10]). In the machine learning community the name generalized regression neural network is typically used. The GRNN model is a nonparametric model where the prediction for a given data point \( x \) is given by the average of the target outputs of the training data points in the vicinity of the given point \( x \) [11]. The local average is constructed by weighting the points according to their distance from \( x \), using some kernel function (we have used a Gaussian kernel). The estimation is just the weighted sum of the observed responses (or target outputs) given by

\[
\hat{y} = \frac{\sum_{m=1}^{M} \mathcal{K}(\frac{x-x_m}{h}) y_m}{\sum_{m=1}^{M} \mathcal{K}(\frac{x-x_m}{h})},
\]

where \( y_m \) is the target output for training data point \( x_m \), and \( \mathcal{K} \) is the kernel function. The parameter \( h \), called the bandwidth, is an important parameter as it determines the degree of smoothness of the fit. The algorithm we use is the Matlab function newgrnn.

2.4 K Nearest Neighbor Regression (KNN)

The \( K \) nearest neighbor regression (KNN) method is a nonparametric method that bases its prediction on the target outputs of the \( K \) nearest neighbors of the given query point. Specifically, given a data point we compute the Euclidean distance between that point and all points in the training set. We then pick the closest \( K \) training data points and set the prediction as the average of the target output values for these \( K \) points. Naturally \( K \) is a key parameter in this method, and has to be selected with care. A large \( K \) will lead to a smoother fit, and therefore a lower variance, of course at the expense of a higher bias, and vice versa for a small \( K \). We built our own KNN procedure using MATLAB.
2.5 Classification and Regression Trees (CART)

CART is a model that is based on a hierarchical tree-like partition of the input space [12]. Specifically, the input space is divided into local regions identified in a sequence of recursive splits. The tree consists of internal decision nodes and terminal leaves. Given a test data point, a sequence of tests along the decision nodes starting from the root node will determine the path along the tree until reaching a leaf node. At the leaf node a prediction is made according to the local model associated with that node. To implement CART, we used the Matlab function treefit.

2.6 Support Vector Regression (SVR)

Support vector regression (Scholkopf and Smola [13], [14]) is a successful method based on using a high-dimensional feature space (formed by transforming original variables), and penalizing the ensuing complexity using a penalty term added to the error function. Consider first, for illustration a linear model. Then, the prediction is given by

\[ f(x) = w^T x + b, \]  

(2)

where \( w \) is the weight vector, \( b \) is the bias and \( x \) is the input vector. Let \( x_m \) and \( y_m \) denote respectively the \( m \)th training input vector and target output, \( m = 1, \ldots, M \).

The error function is given by

\[ J = \frac{1}{2} \|w\|^2 + C \sum_{m=1}^{M} |y_m - f(x_m)|_\epsilon. \]  

(3)

The first term in the error function is a term that penalizes the model complexity. The second term is the \( \epsilon \)-insensitive loss function, defined as

\[ |y_m - f(x_m)|_\epsilon = \max\{0, |y_m - f(x_m)| - \epsilon\} \]  

(4)

It does not penalize errors below \( \epsilon \), allowing it some wiggle room for the parameters to move to reduce model complexity. It can be shown that the solution that minimizes the error function is given by

\[ f(x) = \sum_{m=1}^{M} (\alpha_m^* - \alpha_m) x_m^T x + b, \]  

(5)

where \( \alpha_m \) and \( \alpha_m^* \) are Lagrange multipliers. The training vectors giving non-zero Lagrange multipliers are called support vectors, and this is a key concept in SVR theory. Non-support vectors do not contribute directly to the solution. This model is extended to the nonlinear case through the concept of kernel \( K \), giving a solution:

\[ f(x) = \sum_{m=1}^{M} (\alpha_m^* - \alpha_m) K(x_m^T x) + b. \]  

(6)

A common kernel is the Gaussian kernel. Assume its width is \( \sigma_K \) (the standard deviation of the Gaussian function). In our simulations we used the toolbox provided by Canu et al. [15].

2.7 Gaussian Processes (GP)

Gaussian process regression is based on modeling the observed responses of the different training data points (function values) as a multivariate Gaussian random variable. For these function values an a priori distribution is assumed that guarantees smoothness properties of the function. Specifically, the correlation between two function values is high if the corresponding input vectors are close (in the Euclidean distance sense) and decays as they go farther from each other. The posterior distribution of a to-be-predicted function value can then be obtained using the assumed prior distribution using simple probability manipulations. Let \( V(X, X) \) denote the covariance matrix between the function values, where \( X \) is the matrix of input vectors of the training set.

Let the \( (i, j)^{th} \) element of \( V(X, X) \) be \( V(x_i, x_j) \), where \( x_i \) denotes the \( i^{th} \) training input vector. A typical covariance matrix is the following:

\[ V(x_i, x_j) = \sigma_f^2 e^{-\frac{\|x_i - x_j\|^2}{2\sigma_e^2}}. \]  

(7)

where \( \sigma_f \) is the data standard deviation and \( \alpha \) is the length scale of the covariance function. In addition, some independent zero-mean noise having standard deviation \( \sigma_n \) is assumed to be added to the function values to produce the observed responses (target values). Then, for a given input vector \( x_\star \), the prediction \( \hat{y}_\star \) is derived as

\[ \hat{y}_\star = E(y_\star | X, y, x_\star) = V(x_\star, X)^T [V(X, X) + \sigma_n^2 I]^{-1} y, \]  

(8)

where \( y \) is the vector of target outputs (response values) for the training set. In our simulations we used the toolbox provided by Rasmussen at al [16].

3 Methodology

3.1 Data and Variables

In this study we use data published in the study made by Law and Pine to forecast inbound travel demand for Hong Kong [2]. In their study, Law and Pine aimed to model yearly demand for travel to Hong Kong in the time span from 1970 to 1999, from the following key tourist originating countries: China, Japan, United States, United Kingdom, and Taiwan. Law and Pine assert that demand for travel to Hong Kong from a particular country can be represented by the following function

\[ Q_{it} = f(t, \text{INC}_i, \text{RP}_t, \text{HR}, \text{FER}_t, \text{POP}_i, \text{MKT}), \]  

(9)

where \( Q_{it} \), the dependent variable, represents origin \( i \)'s demand for travel to Hong Kong at time \( t \), \( \text{INC}_i \) represents income of origin \( i \) at time \( t \), \( \text{RP}_t \) represents prices in Hong Kong relative to origin \( i \) at time \( t \), \( \text{HR} \) represents average hotel rate in Hong Kong at time \( t \), \( \text{FER}_t \) represents foreign exchange rate measured as units of Hong Kong's currency per unit of origin \( i \)'s currency at time \( t \), \( \text{POP}_i \) represents population in origin \( i \) at time \( t \), and \( \text{MKT} \) represents...
marketing expenditures to promote Hong Kong tourism industry at time $t$. Following the mainstream of tourism forecasting, demand for travel is measured by the total number of visitor arrivals to Hong Kong. Income of an origin country is represented by the country’s per capita income. Relative price is defined as the ratio of CPI of Hong Kong to that of the origin country. In addition, average hotel rate is used as a proxy variable for the tourists’ costs of living in Hong Kong. For marketing expenses the budget of the Hong Kong Tourist Association is used.

### 3.2 Data Preprocessing

Data preprocessing can have a significant impact on the subsequent forecasting performance. Data have a variety of features such as a trend (linear or nonlinear) that can affect modeling of the data. Based on some experiments, we selected the following preprocessing in the following order:

1. Detrending
2. Scaling

The step for detrending is made only for the number of tourist arrivals by simply fitting a linear model and then subtract it from the original data to remove the trend. Scaling is widely known as an essential step to have the data in a suitable range. Particularly for the neural network models to prevent node saturation by high input values. We have used linear scaling to have every input and the output between $-1$ and $1$. Once we calculated the forecasts, we do the reverse transformation.

### 3.3 Parameter Determination

For every selected method, there are a number of key parameters that determine the complexity of the model. Those parameters need to be determined carefully. For example, the number of hidden nodes for the MLP, the widths of radial bases for RBF, the number of neighbors for KNN, the widths of the kernels for GRNN, and other parameters for SVR and GP that are explained below. We have used the K-Fold-Validation method for parameter determination. In this method, training set is divided into $K$ equal parts (called folds), the model is trained using the $K - 1$ folds and validated using the $K^{th}$ fold. Then we rotate the validation fold and repeat the whole process for $K$ times. We compute the sum of validation errors for the $K$ times. Then we select the parameter value that gives the minimum validation error. Due to the small number of training points we have, we use 5-fold-validation. For every model, we consider a suitable range for the parameter as follows. For the number of hidden nodes in the MLP model the range is $[0, 3, 5, 7, 9, 11, 13, 15]$, where 0 nodes means essentially a linear model. For the RBF, the width of the radial bases is selected from the values $[2.5, 5, 7.5, 10, 12.5, 15, 17.5, 20]$. For the GRNN model the kernel width is selected in the range $[0.02, 0.05, 0.09, 0.1, 0.3, 0.5, 0.7, 0.9, 1.5]$. For the number of neighbors in the KNN model the candidate values are $[2, 4, 6, 8]$. For the case of SVM the key parameters that control the complexity of the model are $\epsilon, C$, and $\sigma_K$. We have fixed $C$ as $y_{max}$ as typically done in practice, and we allowed $\epsilon$ and $\sigma_K$ to be set using 5-fold validation. We consider a two-dimensional grid of $\epsilon$ and $\sigma_K$ using the values $\sigma_{y_0}[0.05, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5] \times \sigma_0[0.05, 0.1, 0.25, 0.5, 0.75, 1, 2, 4]$. The term $\sigma_y$ is the estimated noise level in the time series, and it is estimated as the standard deviation of the output. The term $\sigma_0^2$ gives a measure of the spread of the input variables, and is measured as the sum of the variances of the individual inputs. For Gaussian processes there are three key parameters: $\sigma_f$, $\sigma_n$, and $\alpha$. We opted for the model selection algorithm proposed by Rasmussen and Williams [17]. It is an algorithm that maximizes the marginal likelihood function. Concerning the other less key parameters and model details, we selected them as follows. For MLP, we have used the sigmoidal activation functions for the hidden layer, and a linear output layer. Training is performed for 1000 epochs, using a momentum term 0.2, an adaptive learning rate with initial value 0.2, an increase step of 1.05 and a decrease step of 0.7. For RBF’s the maximum number of radial bases is selected as 25% of the size of the training set. For GRNN and SVR, we used the commonly used Gaussian kernel.

### 4 Analysis and Results

This study builds seven different machine learning techniques to model the recent demand for the key tourist generating countries/regions to Hong Kong. It extends the comparison conducted by Law and Pine to include different machine learning models other than backpropagation neural networks. We include also the time index as an input variable in order to investigate the effect of the time. Among the available data points for each time series, the first two-thirds (11 for China, and 20 for other origins) were selected for model establishment (calibration or training) and the remaining entries were used for model testing (out of sample set). It should be noted that there was a drastic decrease in total tourist arrivals in Hong Kong since 1997 due to the Asian financial crisis. This portion of the time series is part of the test set, so we have a stringent test for each model whether it can hold up in this period of drastic structural change. The forecasting quality was measured in terms of mean absolute percentage error (MAPE) measured by the following equation

$$MAPE = \frac{1}{M} \sum_{m=1}^{M} \frac{|y_m - \hat{y}_m|}{y_m},$$

(10)

where $y_m$ is the target output and $\hat{y}_m$ is the prediction. In order to even out the fluctuations due to the random initial weights for the MLP and the parameter determination using the K-fold-validation approach, we repeated the whole process 20 times. In each run, we shuffle the training set randomly, then estimate the parameter values, train the
models, and finally make the out of sample prediction. For the MLP, we select the best of five trained models based on the training error.

Table (1) shows the average MAPE for the 20 runs for every considered method and for every originating country. Table (1) represents the results when time index is included as an input variable. The last two columns in the table represent the weighted average error MAPE and the average standard deviation for every considered method on all the datasets (for all originating countries). We use the weighted average instead of average because the size of China test set is smaller than those of the other datasets. From these results, it is clear that machine learning methods differ significantly in performance. There is no dominant method which performed the best for all the datasets. Results are quite acceptable considering the small size of the datasets. Most machine learning models require large datasets for model training. Another issue that has to be considered is the Asian crisis that happened in mid 1997. It is known that such catastrophic events perturb the tourism industry with respect to historical trends. This can of course complicate the forecasting problem and affect the performance. Including the time index as an input can be used to account for the trend existing in the data. Although we make detrending as a preprocessing step, time index can help for the nonlinear trend which can not be removed by simply subtracting the linear trend. On average, GRNN performed the best for the five datasets, followed by KNN, GP. Then comes RBF and SVR, which were very close to each other and finally MLP and CART. The relatively bad performance of the MLP compared to other methods is perhaps because of the small size of the training set relative to the higher number of inputs. The highly nonlinear nature of the data requires more training data points for the MLP to be able to model such nonlinearity. However, the MLP results are numerically consistent with the findings in Law and Pine’s study. Of course, because of the small size of the data, the obtained ranking should not indicate general ranking of the different methods for general forecasting problems, and implies only ranking of the methods w.r.t. the considered data only. The standard deviations for MLP, RBF and CART represent the highest standard deviations among all the methods. This is because those methods are highly diverse models in which small changes obtained by shuffling the data at each run, will produce different trained models. However, GRNN, KNN, SVR, and GP are considered more stable methods, so their standard deviations across the 20 runs are very low. Frechting considered forecasts with MAPE values of less than 10% as highly accurate forecasting, between 10% and 20% as good forecasting, between 20% and 50% as reasonable forecasting, and larger than 50% as inaccurate forecasting [1]. From these results, GRNN was the only method that achieved three highly accurate forecasting, one good forecasting, and only one reasonable forecasting. KNN, MLP, GP, and RBF did not attain any inaccurate forecasting. However, SVR and CART got only one inaccurate forecasting regarding Taiwan dataset.

5 Conclusion

We investigated how different machine learning models can be applied in the tourism prediction problem to make predictions for tourism demand to Hong Kong. We focused on seven commonly used machine learning techniques. Additionally, we investigated the effect of the time index when it has been included as an input variable. Although this study is limited both in the number of originating countries, time span, and sample sizes, the findings should be of use to tourism practioners and researchers who are interested in forecasting using machine learning methods. Machine learning models were not widely used in tourism demand forecasting except for multi-layer perceptron models. Our findings reflect that machine learning models other than MLP models can give a more satisfactory performance in particular when the available data are very small. There is no method that can be considered the best one for all the datasets.

Acknowledgement

This work is part of the Data Mining for Improving Tourism Revenue in Egypt research project within the Egyptian Data Mining and Computer Modeling Center of Excellence.

References


<table>
<thead>
<tr>
<th>Method</th>
<th>Japan</th>
<th>Taiwan</th>
<th>US</th>
<th>UK</th>
<th>China</th>
<th>Weighted Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>0.23</td>
<td>0.47</td>
<td>0.21</td>
<td>0.16</td>
<td>0.13</td>
<td>0.252</td>
<td>0.029</td>
</tr>
<tr>
<td>RBF</td>
<td>0.19</td>
<td>0.25</td>
<td>0.27</td>
<td>0.16</td>
<td>0.33</td>
<td>0.228</td>
<td>0.034</td>
</tr>
<tr>
<td>GRNN</td>
<td>0.17</td>
<td>0.26</td>
<td>0.086</td>
<td>0.084</td>
<td>0.079</td>
<td>0.142</td>
<td>0.003</td>
</tr>
<tr>
<td>KNN</td>
<td>0.16</td>
<td>0.42</td>
<td>0.20</td>
<td>0.083</td>
<td>0.11</td>
<td>0.202</td>
<td>0.006</td>
</tr>
<tr>
<td>CART</td>
<td>0.16</td>
<td>0.50</td>
<td>0.28</td>
<td>0.09</td>
<td>0.14</td>
<td>0.246</td>
<td>0.01</td>
</tr>
<tr>
<td>SVR</td>
<td>0.22</td>
<td>0.52</td>
<td>0.18</td>
<td>0.086</td>
<td>0.14</td>
<td>0.238</td>
<td>0.003</td>
</tr>
<tr>
<td>GP</td>
<td>0.18</td>
<td>0.49</td>
<td>0.13</td>
<td>0.082</td>
<td>0.14</td>
<td>0.213</td>
<td>3.746x10^{-19}</td>
</tr>
</tbody>
</table>