

Selecting Location of Retail Stores Using Artificial Neural Networks and Google Places API

Mehmet Hakan Satman¹ & Mustafa Altunbey²

¹ Department of Econometrics, Istanbul University, Istanbul, Turkey

² PhD. Candidate, Dogus University, Istanbul, Turkey

Correspondence: Mehmet Hakan Satman, Department of Econometrics, Faculty of Economics, Istanbul University, Beyazit, Istanbul, Turkey. Tel: 90-212-440-0000 ext.10563. E-mail: mhsatman@istanbul.edu.tr

Received: December 23, 2013 Accepted: January 12, 2014 Online Published: January 21, 2014

doi:10.5539/ijsp.v3n1p67

URL: <http://dx.doi.org/10.5539/ijsp.v3n1p67>

The research is financed by NT Book Stores - <http://www.nt.com.tr>

Abstract

Selection of locations of retail stores (similarly, facilities) is a kind of decision problem in which the researcher targets maximum profit or minimum cost by using statistical or judgmental forecasts. In many cases, statistical forecasts are based on survey data or data collected for an other store which are not reliable as the survey may not measure the potential demand or the other store that is used for comparison may not be similar enough. Addition to this, estimating the potential demand for a non-existing store can be expensive. Google Places API is a software library which provides a nearby search utility that is able to report several kinds of institutions, establishments and other objects placed in a circumferential area for a given radius. In this paper, we suggest a method which is based on estimating the relation between environmental properties of existing retail stores and their financial rankings. We remark that feed forward neural networks are capable to reveal such a relation as they permit using both integer, real and binary variables in the model and non-linearity. We apply our method on the location selection problem of a retail stores chain in Turkey (Note 1) which has 144 clients in its organization. Expert opinions in the organization show us applicability of this method is high in similar situations.

Keywords: location selection, neural networks, Google Places API

1. Introduction

Analogously, selecting location for a facility or a retail store is like an effort of estimating an unknown parameter without any data in some cases. Since the retail store does not exist, there is no data for demand and sales at hand, the decision maker is obliged to use auxiliary data which is collected by surveys, and/or, the data collected by other stores similar and near to the candidate in order to perform any statistical forecast, and/or, subjective comparisons of attributes. Selecting location of a warehouse is more deterministic since the problem can be well modelled and solved using optimization techniques whereas retail stores are more profit-oriented and future demand should be forecasted before making any decision.

The problem of location selection can be separated into two branches. The first branch is a binary decision of building a facility in a predefined location. The decision maker is responsible to find the right answer which is either yes or no. The other branch can be considered as an optimization problem, that is, what must be the location to minimize some cost or maximize the profit. The former requires a single step decision process while the latter consists of iterations of steps given in the former (Church, 1999).

Applebaum (1966) reported the results of a survey study of store location research in the United States. Randhawa and West (1995) suggested an integrated approach which is based on analytical location models and a multi-criteria decision model for the problem of location selection. Yang and Lee (1997) used an analytic hierarchy process for facility location selection which is based on decision makers' subjective comparisons. Shen and Yu (2009) stated that the attributes used in multi-criteria decision models (such as political environment, proximity to markets and customers, supplier networks, expansion potential, availability of transportation systems and utility, quality-of-life

issues, culture issues, etc.) are evaluated by human perceptions and judgments which are not measured numerically. Erbiyik et al. (2012) also applied analytical hierarchy process in a real-life problem as an example of location selection of retail stores. Yong (2006) applied TOPSIS to determine location of plants. Some other areas of location selection problem are determining locations of international tourist hotels (Chou et al., 2008), restaurants (Tzeng et al., 2002), shopping malls (Cheng et al., 2005), shipyards (Guner et al., 2009) and large scale power systems (Nagata et al., 2001), among others.

Using GIS (Geographical Information Systems) is not trivial in the subject of location selection. Cheng et al. (2007) suggested a GIS approach for selection of shopping mall locations. Chang et al. (2008), combined GIS with fuzzy multi-criteria decision-making for location selection. Roig-Tierno et al. (2013) suggested a method based on analytic hierarchy process and GIS data for selecting retail store locations. In their study, Roig-Tierno et al. (2013) used parking, accessibility by car, distance of competition, growth in the area, etc, as criteria for the decision making procedure. In some perspective, our method is similar to the reported work as we use environmental data collected using Google Places API. An artificial neural network is performed to reveal relations between environmental data and financial performances of existing stores to forecast financial performances of non-existing stores. With these properties, our method can be considered as novel.

In this paper, we devise a new method for the location selection problem which is based on artificial neural networks and geographical data collected using Google Places API. In section 2, we give a brief literature overview on artificial neural networks and their use and refer the Google Places API and its nearby search utility. We present our method on a real-life example applied on a retail stores chain with 144 clients in Turkey. In section 3, we present the results. Finally, in section 4, we discuss the results and conclude.

2. Method

2.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are machine learning tools which mimic the brain and its working principals (Haykin, 1999). Since there are several ANNs reported in literature, we refer ANNs for the *supervised and feed forward neural networks*. Self-Organizing Maps (Kohonen, 1982) are other well-known neural networks which are aimed to be used in unsupervised clustering problems. ANNs are used in face detection (Rowley et al., 1998; Lawrence et al., 1997); automatic control systems of robots (Miller, 1989); automatic object detection (Ranganath & Kuntimad, 1999; LeCun et al., 1989); business management and finance (Azoff, 1994; White, 1988; Smith & Gupta, 2000; Altay & Satman, 2005); bank failure predictions (Tam & Kiang, 1992; Wilson & Sharda, 1994); classification and extracting rules (Towell & Shavlik, 1993; Khan et al., 2001), among others. Detailed technical information about neural networks can be found in Haykin (1999).

Suppose the *regression* model is

$$Y_i = f(\beta, X_i) + \epsilon_i \quad (1)$$

where β is a p -vector of unknown parameters, X is an $n \times p$ matrix of independent variables, Y is a vector of dependent variable, n is the number of observations, p is the number of unknown parameters, $i = 1, 2, \dots, n$, a_i is i th row of a , ϵ is the stochastic error term which is generally assumed to be normally distributed with zero mean and constant variance. If the model shown in Equation (1) is linear and has a single independent variable, the model turns into the *simple linear regression* and it is shown in Equation (2).

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \quad (2)$$

Since the ordinary least squares (OLS) is a distribution-free optimization algorithm, β_0 and β_1 can be estimated using OLS without a distributional assumption, whereas, the maximum likelihood estimator requires the distribution knowledge. Regardless the method used, the estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ are linear and unbiased estimators of the true values. If the dependent variable Y in Equation (1) is not only the function of X but a linear function of more than one independent variable, then the model is called *multiple regression* model and a k -variable multiple regression model is shown in (3).

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \dots + \beta_k X_{k,i} + \epsilon_i \quad (3)$$

If Y is not a vector but a matrix of more than one dependent variable, then the model turns into a *multivariate regression* model (Bedrick & Tsai, 1994). If the dependent variable Y in Equation (1) takes either 0 or 1 as value, the model turns into a *binary response* model. A binary response model is generally used to predict the probability that the dependent variable having 1 as a function of some observed explanatory variables (Davidson

& MacKinnon, 2004). *Linear Probability, logit and probit* are well-known *binary response* models in the statistics literature. A logit model can be written as

$$\frac{Y_i}{1 - Y_i} = e^{-f(\beta_k, X_i)} \quad (4)$$

and it is important in the basis of ANNs as the logistic function is frequently used as the *transfer* function. Although the logistic regression and ANNs have their roots in two different disciplines (statistics and computer science), they share similarities and many authors applied and compared these methods in their studies (Dreiseitl & Ohno-Machado, 2002).

If Y is a variable of non-negative integers or specifically count data and the link function in Equation (1) is *Poisson Distribution Function* then $Y \sim \text{Poisson}(\lambda)$ where λ is the mean and the variance of the distribution and the model is estimated using the *Poisson regression* (Saffari et al., 2013).

In short, regression estimators are developed for several measurement types or data structure as we highlight below:

- 1) The relation between dependent and independent variables may be either linear or non-linear,
- 2) Dependent variable(s) may have real, binary or integer values,
- 3) Data may have vector of responses rather than a single dependent variable.

Note that the regression estimators that are mentioned above cannot handle all of the situations, but ANNs are capable to estimate and make predictions using the data that have these properties.

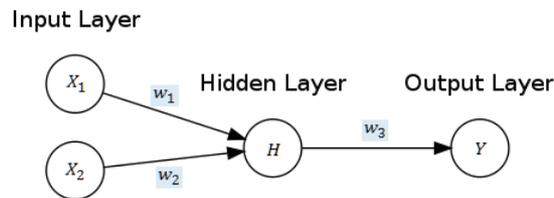


Figure 1. A simple feed-forward neural network diagram

In Figure 1, a diagram of a simple ANN is given. In this diagram, it is shown that the ANN has three layers. The input layer has two nodes and these nodes represent the independent variables or input of the network. The hidden layer has only one node and it is directly connected to the output layer. The mathematical form of this diagram can be written as

$$\hat{Y} = f(H \times w_3) \quad (5)$$

where

$$H = f(X_1 \times w_1 + X_2 \times w_2) \quad (6)$$

and the final equation can be written as

$$\hat{Y} = f(f(X_1 \times w_1 + X_2 \times w_2) \times w_3) \quad (7)$$

where $f(\cdot)$ is the *activation function* and it is generally selected as *sigmoid* as defined below

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-ax}} \quad (8)$$

where a is a custom real. Given the residuals $\hat{e} = Y - \hat{Y}$ and total network error $E = \frac{1}{2} \sum \hat{e}^2$, solution of the equations $\frac{\partial E}{\partial w_i} = 0$ for $i = 1, 2, \dots, p$ minimizes the total network error. Since derivatives of the *sigmoid* and *tanh* functions can be easily handled, they are frequently used. Note that any activation function can be used instead, including $y = x$, but derivatives are required if any gradient based algorithm performed for the optimization. Evolutionary algorithms, such as genetic algorithms (Goldberg & Holland, 1988) and differential evolution (Storn & Price, 1997) can also be applied to estimate unknown parameters w_i (Kalderstam et al., 2013; Ilonen et al., 2003).

In most of cases, the mathematical model can also be defined as

$$\hat{Y} = f(f(X_1 \times w_1 + X_2 \times w_2 + \text{bias}) \times w_3) \quad (9)$$

where *bias* is an input node which has the value of 1. This parameter plays a role similar to *intercept* parameter β_0 in model (3).

2.2 Google Places API

Google Place API (Note 2) (Application Programming Interface) provides location based geographical data via http (Note 3) over the Internet. The API, in deep, allows developers to build applications with capabilities of Place Search, Place Details, Place Actions, Place Photos, Place Autocomplete and Query Autocomplete.

Some of the reported areas of Google Places API are generating cartographic visualisation (Gibin et al., 2008), determining geographically disconnected areas after a disaster (Chen et al., 2011), recommendation of touristic sights (Kabassi, 2010; Yu & Chang, 2009), and location based services in mobile devices (Singhal & Shukla, 2012), among others.

Nearby search functionality in the current version returns a list of items with supported types given in Table 1. Using this API, it is possible to fetch the number of items in a circumferential area for a given radius. The format of results is either JSON (Note 4) or XML (Note 5) which is easy to parse.

The free version of API limits the number of results per request to 60. These results are fetched in groups of 20, so whole the results must be handled in iterations. Since the maximum number of objects for each items is restricted, a small area should be selected to prevent the effects of censored data.

Table 1. Supported data types by Google Places API

Items		
accounting	embassy	museum ✓
airport	establishment ✓	night_club
amusement_park	finance ✓	painter
aquarium	fire_station	park
art_gallery	florist ✓	parking ✓
atm ✓	food ✓	pet_store
bakery ✓	funeral_home	pharmacy ✓
bank	furniture_store	physiotherapist
bar ✓	gas_station	place_of_worship
beauty_salon	general_contractor	plumber
bicycle_store	grocery_or_supermarket ✓	police ✓
book_store ✓	gym	post_office ✓
bowling_alley	hair_care	real_estate_agency ✓
bus_station ✓	hardware_store	restaurant ✓
cafe ✓	health ✓	roofing_contractor
campground	hindu_temple	rv_park
car_dealer	home_goods_store ✓	school ✓
car_rental	hospital ✓	shoe_store
car_repair	insurance_agency	shopping_mall ✓
car_wash	jewelry_store ✓	spa
casino	laundry	stadium
cemetery	lawyer ✓	storage
church ✓	library ✓	store ✓
city_hall ✓	liquor_store	subway_station ✓
clothing_store ✓	local_government_office ✓	synagogue ✓
convenience_store	locksmith	taxi_stand ✓
courthouse	lodging	train_station ✓
dentist	meal_delivery	travel_agency ✓
department_store	meal_takeaway	university ✓
doctor ✓	mosque ✓	veterinary_care
electrician	movie_rental ✓	zoo
electronics_store ✓	movie_theater ✓	
accounting	moving_company	

† https://developers.google.com/places/documentation/supported_types

2.3 Proposed Method and an Example

The number of *objects* fetched using Google Places API serves many information about the structure of potential customers for a given area. For instance, the number of schools and universities is a proxy variable for educational structure of population, whereas, the numbers of bus stations, gas stations, subway stations and taxi stands are proxies for transportation structure of the given area. Some variables, such as, the number of casinos and the number of night clubs and pubs give an idea of customer population in certain time intervals. The number of worship places also reflects the male/female ratio and other properties of potential population in special dates and predefined days. The total number of banks, atms, financial institutions is a symptom of high daytime population of adults, whereas, the number of cafes, shopping malls and restaurants reminds young people at weekdays and families at weekends. Finally, the environmental structure of a certain area directly influences the distribution of population. If the relation between the environmental structure and demand of existing retail stores in same sector is estimated then demand of the new store under consideration could be predicted.

Suppose that we have the frequency data of Table 1 for N retail stores in a chain and rankings of each stores assigned by a clustering algorithm or personal judgments as shown in Table 2. *Rankings* can be numeric or ordered as well as non-numeric. In our example, we apply *k-Medoids* clustering algorithm (Kaufman & Rousseeuw, 2009) on a data in which the columns are number of sales slips, total sales etc. and rows are retail stores. An other clustering algorithm such as *k-Means* can be used instead but *k-Medoids* is superior as it is more robust and efficient (Reynolds et al., 2006).

Table 2. Environmental data

Stores	#Bus Stations	#Restaurants	#Banks	...	Ranking
1	3	12	7	...	A
2	2	1	13	...	B
3	0	30	11	...	E
⋮	⋮	⋮	⋮	⋮	⋮
N	5	2	41	...	C

Configuration of our ANN is given in Table 3.

Table 3. ANN configuration

Parameter	Value	Description
Inputs	42	Selected by best performance and expert judgments
Hidden Layer	1	120 Neurons with Sigmoid transfer function
Outputs	5	Bit vectors for Rankings as in Table 4

Our proposed method is based on constructing an ANN that takes *environmental variables* as input (Note 6) and *Rankings* as output. If the relation between the input and the output variables is exposed, we can feed the network with input values of new store and evaluate the generated output.

Since the input variables have the integers between 0 and 60, no encoding and decoding operations are required. If any variables in different ranges are used, data should be rescaled in order to map all of the variables into the same space. If the output variable is non-numeric, the output should be translated to binary vectors. Gilbert and Troitzsch (1999) stated that the variables are best coded in binary with units assigned to each binary position.

In our example, the output variable holds the segmentation labels A, B, C, D, E as *Rankings* which are encoded in binary vectors as shown in Table 4.

Table 4. Binary encoding of output variable

Ranking	Y_1	Y_2	Y_3	Y_4	Y_5
A	1	0	0	0	0
B	0	1	0	0	0
C	0	0	1	0	0
D	0	0	0	1	0
E	0	0	0	0	1

We train the ANN using the 123 retail stores and left the 18 cases for test to take a decision of opening 3 new stores. Zhang et al. (1999) stated that the large training sets are more capable to reveal clusters of data, however, if the objective is forecasting or clustering unseen cases, size of the test data must be enlarged. In many cases, the ratio of the test cases is ranging from 10% to 30% and the researcher would select the network which performs best on the testing set (Kaastra & Boyd, 1996). In our network, we set the ratio of testing set to 13% which includes the stores that are suggested by firm experts. In the whole research process, the *rankings* of stores in testing set are unknown for an objective and blind check.

We select 42 input variables out of 96 by taking expert advises and considering performance and they are shown in Table 1 with ✓ symbol. Note that a principal component analysis could be applied on the data for dimensional reduction. However, the process of interpreting, calculating and extracting linear combinations is not straightforward. There are many suggested rules of thumb for determining the number of hidden nodes, such as $n/2$, n , $n + 1$ and $2n + 1$, where n is the number of input nodes (Zhang et al., 1999). In our case, non of these suggestions performed well and we choose a single hidden layer with 120 neurons as the network topology.

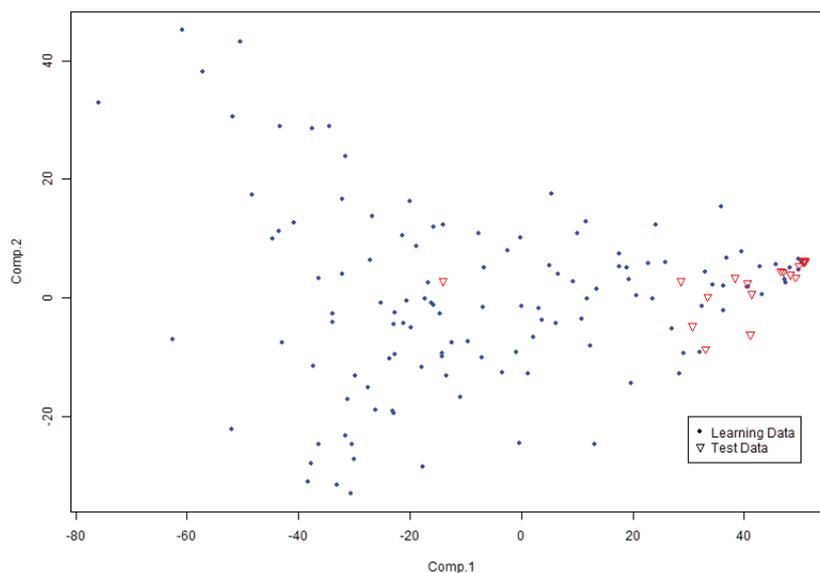


Figure 2. Two principal components of input data

In Figure 2, principal components of input data are shown. Learning and test sets are plotted using • and ▽ symbols, respectively. Figure 2 shows an approximation of similarities between old and new stores by means of input variables rather than the relation between inputs and *Rankings*. Note that other dimension reduction tools such as *multidimensional scaling* can be used instead. Without a dimensional reduction technique, a distance matrix can be calculated and k nearest stores can be identified. Median of *Rankings* of selected nearest stores is the estimated *Ranking* of the new store. However, these methods only increase the human perception on the given input variables but the relation between geographical data and the financial rankings is not investigated.

3. Results

Success of the proposed method for retail store location selection is evaluated in two stages. Firstly, the selected network is acceptable as it has small sum of squares of errors over all learning dataset and forecasting accuracy on the out of sample data.

Environmental data are fetched using a *JavaScript* program which uses the Google Places API as a library. The minimum, maximum, $Q1$, median, and $Q3$ statistics of 42 variables are shown in Figure 3 (a-d) using boxplots. The radius of circumferential areas is set to 500 meters for all stores. This limitation is convenient for most cases except for the number of *restaurants* because 43 stores have at least 60 *restaurants* and it is impossible to be sure whether they have 60 or more *restaurants* using the free edition of the API. In our study, we left the radius as 500 meters by considering the descriptive statistics of other variables.

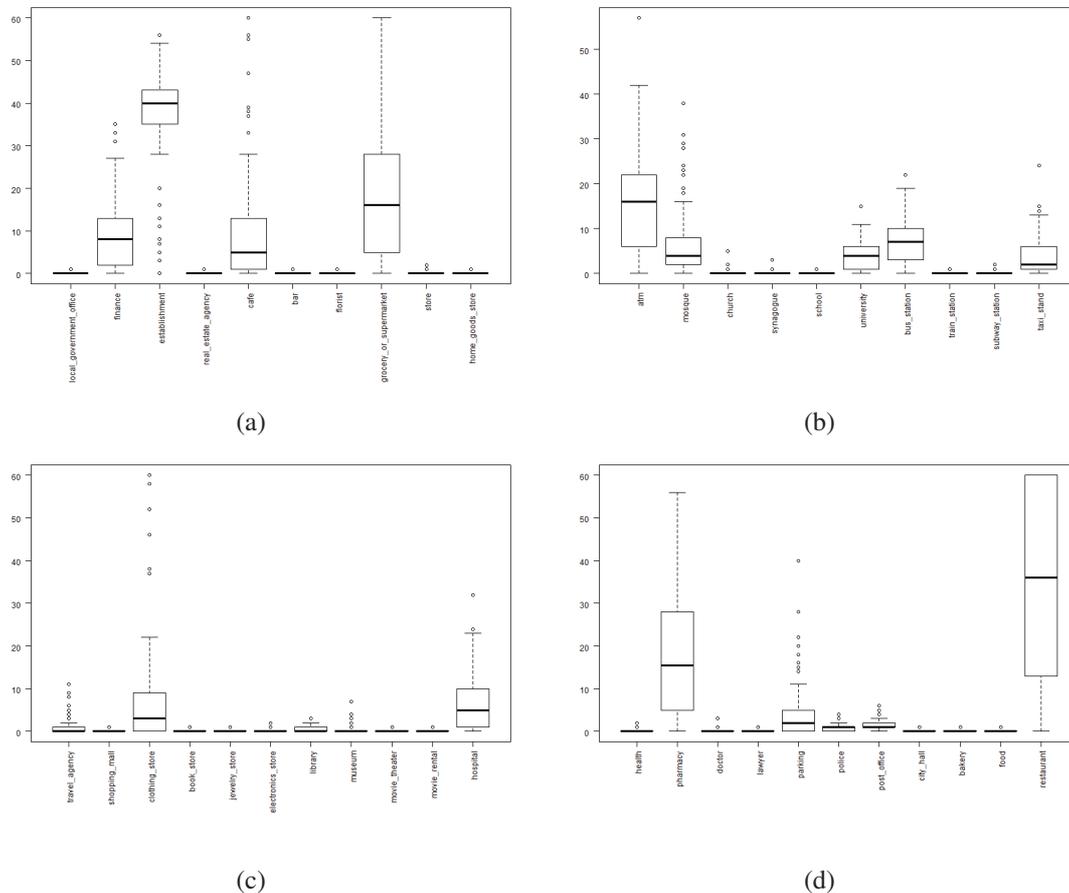


Figure 3. Boxplots of Google places data

The network has been trained in a PC with Intel i5 CPU, 4GBs memory and Linux installed. The statistical software *R* (R Core Team, 2013) and the R package *neuralnet* (Fritsch et al., 2012) were used for analysis. Network has been trained many times and the best one has been reported by the package. The final network has been reported in 0.74 seconds with network error 0.00009.

The small network error shows that the network fits the training data well but forecasting quality of the network should be investigated. MAD (Mean Absolute Deviations), MSE (Mean Square Error), and MEAD (Median Absolute Deviations) statistics are generally calculated for this purpose and these statistics are shown in Table 5.

Table 5. Forecast performance of estimated network

	MAD	MSE	MEAD
Calculated	0.388888889	0.388888889	0
Maximum	4	16	4

In Table 5, it is shown that the calculated MAD, MSE, and MEAD are quite small when they are compared to maximum possible values in the case of worst forecasting performance.

Secondly, method is simulated on the data collected for random locations and expert reviews are considered for testing issues. Reviews of experts show us our method produces reasonable estimates of *Rankings* of candidate stores. Our method also reveals similarities between the candidate and currently available stores.

Finally, the firm has applied the algorithm and forecasted the *Rankings* of three candidate retail stores for several places. After the starting up process, they reported that the *Rankings* are realized as forecasted by the method.

4. Discussion

In this paper, we suggest a method for estimating the relation between the environmental structures and the financial

Rankings of retail stores using ANNs and Google Places API. The method was applied in a retail stores chain (NT Book Stores) with 144 clients which is previously used to make decisions by marketing researches. The firm reported that neither multi-criteria decision methods nor survey data worked well in their cases.

In statistics, properties of an estimator should be investigated either theoretically or computationally. Although the neural networks has an increasing trend in statistics literature, the properties of the estimator are not unveiled because of their *black-box* and non-linear nature. However, partitioning data into training and testing sets generally reflects the forecasting performance.

We are aware of testing usability of such a method is not possible using a single example, however, we have some evidence that the method work well on estimating financial *Rankings* of candidate retail stores by only introducing their environmental structures to an ANN trained before as we stated in Section 3.

The proposed method is inexpensive as it is replacing the data collecting, survey or marketing research part by downloading and parsing environmental data which could be considered as proxy variables. Note that reliability of the method increases as the number of existing retail stores increases. If the training set is not large enough to reveal the true relation, method may fail.

In our example, since the financial *Rankings* are constructed after performing a clustering algorithm using electronically collected data, they do not contain measurement errors. However, the data collected by Google Places API is not that reliable in all cases. The researcher should examine the data as it contains *universities* in type of *schools*, *bars* in type of *restaurants*, etc., if the data itself is used as inputs rather than principal components.

Our method forecasts the financial *Rankings* rather than the demand. The estimated demand is simply the arithmetic mean of demands of existing retail stores that fall into the same *Ranking*. Forecasting the demand directly would be the subject of a future study.

The suggested method can be run in a loop for several circumferential areas and the area with the highest *Ranking* can be reported. This modification turns our method into an optimization problem which has an objective of profit maximization. Investigating properties and measuring performance of such a *window sliding* algorithm would be also the subject of a future work.

References

- Altay, E., & Satman, M. H. (2005). Stock market forecasting: artificial neural network and linear regression comparison in an emerging market. *Journal of financial management and analysis*, 18.
- Applebaum, W. (1966). Guidelines for a store-location strategy study. *The Journal of Marketing*, 42-45. <http://dx.doi.org/10.2307/1249498>
- Azoff, E. M. (1994). *Neural network time series forecasting of financial markets*. John Wiley & Sons, Inc.
- Bedrick, E. J., & Tsai, C. L. (1994). Model selection for multivariate regression in small samples. *Biometrics*, 226-231. <http://dx.doi.org/10.2307/2533213>
- Chang, N. B., Parvathinathan, G., & Breeden, J. B. (2008). Combining GIS with fuzzy multicriteria decision-making for landfill siting in a fast-growing urban region. *Journal of environmental management*, 87(1), 139-153. <http://dx.doi.org/10.1016/j.jenvman.2007.01.011>
- Chen, L. J., Li, C. W., Huang, Y. T., & Shih, C. S. (2011, November). A rapid method for detecting geographically disconnected areas after disasters. In *Technologies for Homeland Security (HST), 2011 IEEE International Conference on*, pp. 501-506. <http://dx.doi.org/10.1109/THS.2011.6107919>
- Cheng, E. W., Li, H., & Yu, L. (2005). The analytic network process (ANP) approach to location selection: a shopping mall illustration. *Construction Innovation: Information, Process, Management*, 5(2), 83-97. <http://dx.doi.org/10.1108/14714170510815195>
- Cheng, E. W., Li, H., & Yu, L. (2007). A GIS approach to shopping mall location selection. *Building and Environment*, 42(2), 884-892. <http://dx.doi.org/10.1016/j.buildenv.2005.10.010>
- Chou, T. Y., Hsu, C. L., & Chen, M. C. (2008). A fuzzy multi-criteria decision model for international tourist hotels location selection. *International Journal of Hospitality Management*, 27(2), 293-301. <http://dx.doi.org/10.1016/j.ijhm.2007.07.029>
- Church, R. L. (1999). Location modelling and GIS. *Geographical information systems*, 1, 293-303.

- Davidson, R., & MacKinnon, J. G. (2004). *Econometric theory and methods*. New York: Oxford University Press.
- Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5), 352-359. [http://dx.doi.org/10.1016/S1532-0464\(03\)00034-0](http://dx.doi.org/10.1016/S1532-0464(03)00034-0)
- Erbiyik, H., Ozcan, S., & Karaboga, K. (2012). Retail Store Location Selection Problem with Multiple Analytical Hierarchy Process of Decision Making an Application in Turkey. *Procedia-Social and Behavioral Sciences*, 58, 1405-1414.
- Fritsch, S., Guenther, F., & Suling, M. (2012). neuralnet: Training of neural networks. R package version 1.32. Retrieved from <http://CRAN.R-project.org/package=neuralnet>
- Gibin, M., Singleton, A., Milton, R., Mateos, P., & Longley, P. (2008). An exploratory cartographic visualisation of London through the Google Maps API. *Applied Spatial Analysis and Policy*, 1(2), 85-97. <http://dx.doi.org/10.1007/s12061-008-9005-5>
- Gilbert, N., & Troitzsch, K. G. (1999). *Simulation for the Social Scientist*. Open University Press.
- Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, 3(2), 95-99.
- Guneri, A. F., Cengiz, M., & Seker, S. (2009). A fuzzy ANP approach to shipyard location selection. *Expert Systems with Applications*, 36(4), 7992-7999. <http://dx.doi.org/10.1016/j.eswa.2008.10.059>
- Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.
- Ilonen, J., Kamarainen, J. K., & Lampinen, J. (2003). Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters*, 17(1), 93-105.
- Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10(3), 215-236. [http://dx.doi.org/10.1016/0925-2312\(95\)00039-9](http://dx.doi.org/10.1016/0925-2312(95)00039-9)
- Kabassi, K. (2010). Personalizing recommendations for tourists. *Telematics and Informatics*, 27(1), 51-66. <http://dx.doi.org/10.1016/j.tele.2009.05.003>
- Kalderstam, J., Edén, P., Bendahl, P. O., Strand, C., Fernö, M., & Ohlsson, M. (2013). Training artificial neural networks directly on the concordance index for censored data using genetic algorithms. *Artificial intelligence in medicine*. <http://dx.doi.org/10.1016/j.artmed.2013.03.001>
- Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. Wiley Series in Probability and Statistics.
- Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., & Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6), 673-679.
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1), 59-69. <http://dx.doi.org/10.1007/BF00337288>
- Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1), 98-113. <http://dx.doi.org/10.1109/72.554195>
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 541-551. <http://dx.doi.org/10.1162/neco.1989.1.4.541>
- Miller, W. T. (1989). Real-time application of neural networks for sensor-based control of robots with vision. *IEEE Transactions on Systems, Man and Cybernetics*, 19(4), 825-831. <http://dx.doi.org/10.1109/21.35345>
- Nagata, M., Tanaka, K., & Taniguchi, H. (2001). FCL location selection in large scale power system. *IEEE Transactions on Applied Superconductivity*, 11(1), 2489-2494. <http://dx.doi.org/10.1109/77.920370>
- R Core Team. (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from <http://www.R-project.org/>
- Randhawa, S. U., & West, T. M. (1995). An integrated approach to facility location problems. *Computers &*

- Industrial Engineering*, 29(1), 261-265. [http://dx.doi.org/10.1016/0360-8352\(95\)00082-C](http://dx.doi.org/10.1016/0360-8352(95)00082-C)
- Ranganath, H. S., & Kuntimad, G. (1999). Object detection using pulse coupled neural networks. *IEEE Transactions on Neural Networks*, 10(3), 615-620. <http://dx.doi.org/10.1109/72.761720>
- Reynolds, A. P., Richards, G., de la Iglesia, B., & Rayward-Smith, V. J. (2006). Clustering rules: a comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms*, 5(4), 475-504. <http://dx.doi.org/10.1007/s10852-005-9022-1>
- Roig-Tierno, N., Baviera-Puig, A., Buitrago-Vera, J., & Mas-Verdu, F. (2013). The retail site location decision process using GIS and the analytical hierarchy process. *Applied Geography*, 40, 191-198. <http://dx.doi.org/10.1016/j.apgeog.2013.03.005>
- Rowley, H. A., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1), 23-38. <http://dx.doi.org/10.1109/34.655647>
- Saffari, S. E., Adnan, R., & Greene, W. (2013). Investigating the impact of excess zeros on hurdle-generalized Poisson regression model with right censored count data. *Statistica Neerlandica*, 67(1), 67-80. <http://dx.doi.org/10.1111/j.1467-9574.2012.00532.x>
- Shen, C. Y., & Yu, K. T. (2009). A generalized fuzzy approach for strategic problems: The empirical study on facility location selection of authorss' management consultation client as an example. *Expert Systems with Applications*, 36(3), 4709-4716. <http://dx.doi.org/10.1016/j.eswa.2008.06.035>
- Singhal, M., & Shukla, A. (2012). Implementation of Location based Services in Android using GPS and Web Services. *IJCSI International Journal of Computer Science Issues*, 9(1).
- Smith, K. A., & Gupta, J. N. (2000). Neural networks in business: techniques and applications for the operations researcher. *Computers & Operations Research*, 27(11), 1023-1044. [http://dx.doi.org/10.1016/S0305-0548\(99\)00141-0](http://dx.doi.org/10.1016/S0305-0548(99)00141-0)
- Storn, R., & Price, K. (1997). Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359. <http://dx.doi.org/10.1023/A:1008202821328>
- Tam, K. Y., & Kiang, M. Y. (1992). Managerial applications of neural networks: the case of bank failure predictions. *Management science*, 38(7), 926-947. <http://dx.doi.org/10.1287/mnsc.38.7.926>
- Towell, G. G., & Shavlik, J. W. (1993). Extracting refined rules from knowledge-based neural networks. *Machine learning*, 13(1), 71-101. <http://dx.doi.org/10.1007/BF00993103>
- Tzeng, G. H., Teng, M. H., Chen, J. J., & Opricovic, S. (2002). Multicriteria selection for a restaurant location in Taipei. *International Journal of Hospitality Management*, 21(2), 171-187. [http://dx.doi.org/10.1016/S0278-4319\(02\)00005-1](http://dx.doi.org/10.1016/S0278-4319(02)00005-1)
- White, H. (1988). Economic prediction using neural networks: The case of IBM daily stock returns. *IEEE International Conference on Neural Networks*, 451-458. <http://dx.doi.org/10.1109/ICNN.1988.23959>
- Wilson, R. L., & Sharda, R. (1994). Bankruptcy prediction using neural networks. *Decision support systems*, 11(5), 545-557. [http://dx.doi.org/10.1016/0167-9236\(94\)90024-8](http://dx.doi.org/10.1016/0167-9236(94)90024-8)
- Yang, J., & Lee, H. (1997). An AHP decision model for facility location selection. *Facilities*, 15(9/10), 241-254. <http://dx.doi.org/10.1108/02632779710178785>
- Yong, D. (2006). Plant location selection based on fuzzy TOPSIS. *The International Journal of Advanced Manufacturing Technology*, 28(7-8), 839-844. <http://dx.doi.org/10.1007/s00170-004-2436-5>
- Yu, C. C., & Chang, H. P. (2009). Personalized location-based recommendation services for tour planning in mobile tourism applications. *E-Commerce and Web Technologies*, 38-49. http://dx.doi.org/10.1007/978-3-642-03964-5_5
- Zhang, G., Hu, M. Y., Patuwo, B. E., & Indro, D. C. (1999). Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis. *European Journal of Operational Research*, 116(1), 16-32. [http://dx.doi.org/10.1016/S0377-2217\(98\)00051-4](http://dx.doi.org/10.1016/S0377-2217(98)00051-4)

Notes

Note 1. NT Book Stores, <http://www.nt.com.tr>

Note 2. <https://developers.google.com/places/documentation/>

Note 3. Hypertext Transfer Protocol - <http://www.ietf.org/rfc/rfc2616.txt>

Note 4. JavaScript Object Notation

Note 5. eXtensible Markup Language

Note 6. Selected data types are shown in Table 1 with ✓ symbol

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).