International Journal of Research
Available at https://edupediapublications.org/journals

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 10
June 2016

Dynamic Pricing in Ecommerce using Neural Network approach

Kavyashri
Student, Final semester M.tech in
Network and Internet Engineering
Department of Electronics and Communication
Sri Jayachamarajendra College of Engineering
kavya.sparkles@gmail.com

Dr. M.N.Jayaram
Associate Professor
Department of Electronics and communication
Sri Jayachamarajendra College of engineering
Mysore, India
melkotejayaram@yahoo.com

Mr. Jeevitesh M.S.
Tech Lead, Unilog
Content Solution,
Mysore, India
jeevitesh.ms@gmail.com

Abstract— Price optimization uses mathematical approach to determine how the customers will react to various prices for the same product through various medium. Determining retail price of the product is the Challenging effort for the retailer to sustain in the market.

Every seller intends to fix the retail price of the product .So that their revenue is maximized. The approach adjust the retail price of the product dynamically using neural network in order to increase the revenue .In this paper an Artificial Neural Network used to determine optimum price for the product .The model uses back propagation algorithm to get desired output with high accuracy.

Keywords—price optimization; back propagation algorithm; dynamic pricing algorithm.

I. INTRODUCTION

Predicting the optimal price of the product is difficult task in market because price keeps on fluctuating. So every seller wants to set the selling price of their product so that their revenue is increased by dynamic pricing the product where, product were suppose to be sold within a given time horizon. Usually a customer before buying a product selects store/sells for the purchase. The selection can be done over multiple attribute such as competitive price, product stock, discount, rating, delivery time.

Price optimization not only delivers improved profitability but also provides the business can play vital role in determining customers' purchase decision would include product stock, delivery time, discount, rating, and competitor price.

In our model we consider these mentioned five attributes in determining a competitive price for a product P taking consideration of three companies like Savoy, Amazon and EBay. The approach use neural network to determine a competitive price (selling price) for the products in order to maximize sellers' revenue. In our simulation we demonstrate that once the sellers set an initial price of the product, our model adjusts the price of the product automatically with the help of neural network in order to maximize profits.

II. RELATED WORK

Over the past few years there can be observed a noticeable rise in interest of dynamic pricing in commercial and research communities. Several analytical models have been developed for dynamic pricing in online. Some of the research made an assumption that there is only one seller in the market [7]. There exists intelligent agent called pricebots with enable online sellers to dynamically calculate a competitive price for a product. According to Dasgupta et al. [8],"these intelligent agents provide a convenient mechanism for implementing automated dynamic pricing algorithms for the sellers in an online economy". Kephart et al. [1], for their work, considered a picture where a monopolist seller willing to maximize his/her revenue, provided buyers demand curve is random and unpredictable. Li et al. [9] studied the enterprises' dynamic decision problem on price strategies (dynamic pricing decision) in duopolistic retailing market under uncertain market state. Chinthalapati et al. [10] used machine learning based approach to study price dynamics in an electronic retail market. Alexandre et. al [11] discussed on the problems of dynamic pricing in finite time horizon. They considered a retailer who has to set the price of a good to optimize the total expected revenues over a period of time T. Their model is dependent on demand curve of the products. Kong [12], in his paper, examined seller strategies for dynamic pricing in a market for which a seller has finite time horizon to sell its inventory. Tesauro and Kephart [3], in their experiment they assumed that there are only two competing sellers participating in the economy who alternatively take turns in adjusting their prices at each time step. Dey [et.al [2] discussed methodical model based on Analytical hierarchy Process (AHP) and Artificial Neural Network (ANN) for estimation of player price in IPL. They used back propagation algorithm to predict players bidding price in IPL. Yatim [5], in his paper describes the global issue of sustainability, data collection and potential applications of an analysis using artificial neural network in predicting service life for an ongoing research on affordable quality housing at University Technology Malaysia.

The back-propagation training method used within neural network model was able to predict the service life of timber as material to form building components for coastal area in Malaysia. There is a need to improve the neural network performance on data distribution and learning capability. Ghose and Tran [6] used an approach of dynamic pricing where buyers purchase decision is dependent on many parameters to determine product price dynamically by using feed-forward neural network to maximize the revenue.

Architecture of Neural Network: Neural network architecture consists of three layers: input layer, hidden layer and a output layer. Each layer consists of one or more nodes. This is shown in the figure1. The line indicates the flow of information from one node to another node or more nodes.

Nodes at the input layer are passive i.e. they do not modify the data. They receive single value as input and duplicates to multiple outputs.

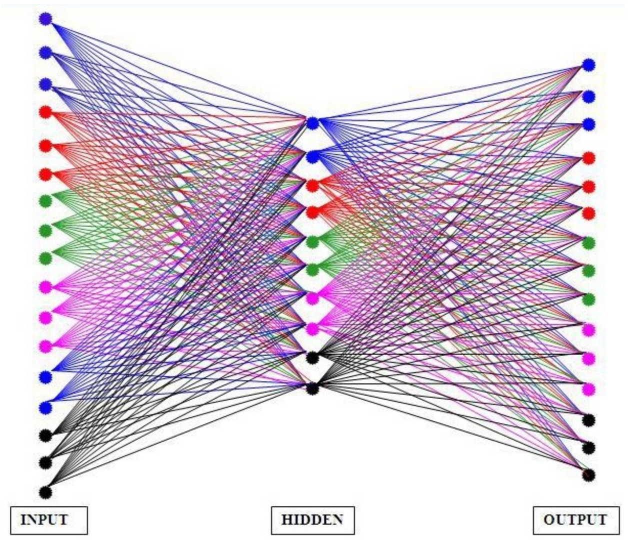


Figure1: Neural Network Architecture

The values entering hidden layer are multiplied by weights, a set of predetermined number stored in program. The weighted inputs are then added to produce a single number. Before leaving the node this input is passed through a non linear mathematical function.

Back Propagation Algorithm:

The back-propagation algorithm is the most widely used method for calculating the error derivative of the weights, denoted by ΔE. ΔE can be computed as follows [4]:

$$\textbf{Error, } E = \tfrac{1}{2}\,\mathrm{Err}^2 \approx \tfrac{1}{2}(X - Y)^2$$

where X = desired output, Y = actual output.

The squared error can be reduced by calculating partial derivative of error E with respect to every weight $W_{j,i}$ as follows:

$$\Delta E = \frac{\partial E}{\partial W_{j,i}} = \frac{\partial}{\partial W_{j,i}}\left(\frac{1}{2}Err^2\right) = Err \times \frac{\partial Err}{\partial W_{j,i}}$$

$$= Err \times \frac{\partial}{\partial W_{j,i}}(X - Y)$$

The process of training the three-layered network using back-propagation algorithm to reduce the errors at each layer by updating the weights is as follows:

i. Compute the errors ΔE* at the output units which is simply the difference between desired output (X) and actual output (Y).

ii. Update the weights between the hidden layer and the output layer by using the errors ΔE*.

iii. Propagate the errors ΔE* back to the hidden layer to find the errors ΔE, - at the hidden layer units.

iv. Update the weights between the input layer and the hidden layer by using the errors ΔE, determined in step (iii).

The activation function/controls the amplitude of the output of the units and we used tanh function as activation function.

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

The algorithm uses neuralnet R package to build the code. The following function describes the optimization price.

neuralnet (formula, data, hidden = 1, threshold = 0.01, stepmax = 1e+05, rep = 1, start weights = NULL, learning rate. Limit = NULL, learning rate. Factor = list (minus = 0.5, plus = 1.2), learning rate=NULL, life sign = "none", lifesign.step = 1000, algorithm = "rprop+", err.fct = "sse", act.fct = "logistic", linear. Output = TRUE, exclude = NULL, constant. Weights = NULL, likelihood = FALSE) [6]

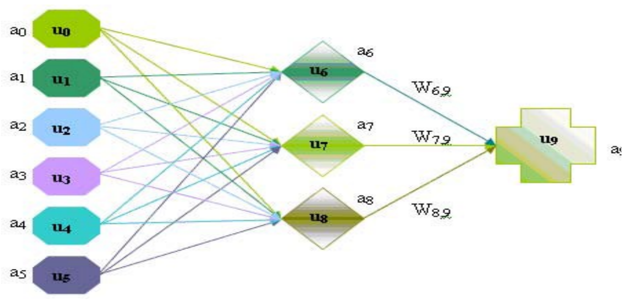
With many trails and error method the following arguments for the neuralnet function is used.

Formula	a symbolic description of the model to be Fitted in formula.
Hidden	a vector of integers specifying the number of hidden neuron (vertices) in each layer
Threshold	a numeric value specifying the threshold for the partial derivatives of the error function as stopping criteria.
Rep	the number of repetitions for the neural network's training
Start Weights	a vector containing starting values for the weights. The weights will not be randomly initialized
Learning rate	numeric value specifying the learning rate used by traditional back propagation
Algorithm	a string containing the algorithm type to calculate the neural network.The following types are possible: 'backprop', 'rprop+','rprop-', 'sag', or 'slr'. 'backprop' refers to back propagation, 'rprop+' and 'rprop-' refer to the resilient back propagation with and without weight backtracking, while 'sag' and 'slr' induce the usage of the modified globally convergent algorithm (grprop)
Err.fct	a differentiable function that is used for the calculation of the error. Alternatively, the strings 'sse' and 'ce' which stand for the sum of squared errors and the cross-entropy can be used.

.

Design of the Proposed Model:

In our model, for determining the price used a back propagation neural network which contains three layers: input layer, hidden layer and output layer. The network we designed consists of five units in the input layer across three companies. The input layer also consists of one extra unit *u*0 as the bias unit. We set the value of *a*0 to the production cost of the product. Usually, sellers are not willing to sell their products below the production cost of the corresponding products. Hence, we considered the production cost of the product as the output of the bias unit. Initially, all the values *a*1, *a*2, *a*3, *a*4, *a*5, a6, a7, a8, a9, a10, a11, a12, a13 of the input units are set by the sellers. In setting the initial price of a product, we assume that sellers use their prior knowledge about the prices of the product offered by other competing sellers in the market.



Competator Price	discount	Rating	Delivery cost	MRP	Product stock
Input 1	Input 1	Input 1	Input 1	Input 1	Input 1
Input 2	Input 2	Input 2	Input 2	Input 2	Input 2
Input 3	Input 3	Input 3	Input 3	Input 3	Input 3

Figure 2: A three-layered back propagation Neural Network for Price Determination

Our model can accept more attributes. One additional unit in the input layer needs to be added for each new attribute. On the other hand, in order to remove an attribute from the network the corresponding unit from the input layer, along with all the links that are connected to the unit, has to be eliminated. This implies that our model will work for any number of attributes.

Train Network

Assumption is made that sellers use their historical data as the training patterns to the network. A training pattern consists of a set of inputs with desired output. We began our simulation by training the network of our model with 10 sets of training patterns so that errors can be minimized as much as possible by using back propagation algorithm. We trained our network for nine different numbers of epochs or iterations: 10, 50, 100, 500, 1000, 5000, 10000, 50000 and 100000. As the training continues, after each iteration or epoch, the network calculates amount of error. The calculated error is then used to update the weights of the links by using back propagation algorithm so that error is minimized in the next iteration. Practically the value of error never becomes zero, but approaches to zero. We let our network to tolerate an error of amount 0.01 and 0.001. We run our network with five different learning rates: 0.01, 0.005, 0.001, 0.0005 and 0.0001. Analysis of the training process in the following sub-section indicates that the model performs better if we use 50000 epochs with 0.005 learning rate during training the network.

Dataset is taken from three companies like Savoy, EBay and Amazon website for many product based on research parameters as shown in below Table 1.

Table 1: Training Dataset

Dynamic Pricing Algorithm

The price of the product determined by the network (Fig. 1) can be found by using final output *a*9. The value of $a9$ can be calculated with the aid of equation (1) as follows:

Finaloutput, $a9 = f(W6,9a6 + W7,9a7 + W8,9a8)$. (2)

where, $a6 = f(W0,6a0 + W1,6a1 + W2,6a2 + W3,6a3 + W4,6a4 + W5,6a5)$

$a7 = f(W0,7a0 + W1,7a1 + W2,7a2 + W3,7a3 + W4,7a4 + W5,7a5)$

$a8 = f(W0,8a0 + W1,8a1 + W2,8a2 + W3,8a3 + W4,8a4 + W5,8a5)$

The phase is sub-dividing the process of dynamic pricing by our model of neural network into two phases: training phase and price determination phase. In the training phase we train our network with a set of training pattern. A training pattern consists of a set of inputs with desired output. A typical set of training pattern for our model each row represents a training pattern which contains a set of inputs with corresponding desired output. Initially assumptions that the buyers have equal preference on all the five attributes across three companies that are considered. Therefore, link between input units and hidden units with equal weights. The purpose of the training process is to adjust the weights between the links such that the errors are minimized. To obtain this goal we feed units of the input layer of our network with the corresponding input values ($Input_{ij}$) from each training pattern. We then determine the output from our network and compare it with the corresponding desired output (*Output*) of the training pattern to calculate error. Finally, then update weights between the links depending on the calculated errors. In our model, during the process of training, the errors between the links are minimized by using back-propagation technique.

The training process can be portrayed by the following steps:

i Input values from a training pattern to units of the input layer of the network.

ii If the current training pattern is the first training pattern of the training set, then associate the links between input units and hidden units with equal weight. Also, associate the links between hidden units and output unit equally.

iii Determine the value from the output layer.

iv Compute the error, i.e., the difference between desired output of the training pattern and the value obtained in step *iii*.

v If the error is more than zero then go to step *viii*.

vi If the error is approximately zero and there is more training pattern left, then take the next training pattern and go to step *i*.

vii If the error is approximately zero and there is no more training pattern left, then terminate the training process.

viii Update the weights of the links using back-propagation technique to minimize the error.

ix Go to step *iii*.

The training phase updates weights between the links of the network as needed so that it can provide better output. Once the training process is complete, our model of network is ready to determine a competitive price for a product, P, from the price determination phase as follows:

i Set the production cost of the product, P as the input to bias unit of the input layer and set the weights of the links associated with bias unit to 1.

ii Set the values (a_i) of the input units for the corresponding purchase attributes of product by using prior knowledge about the prices of product offered by other competing sellers.

iii Run the network and derive the price from the output layer.

iv Set the price from the output layer as the product price.

Results

An ANN with back-propagation algorithm was trained and the training epoch (cycles) set for each network is 10,000. The purpose of the training is to minimize the mean squared error (*MSE*) and RMSD represents the sample standard deviation of the differences between predicted values and observed values. The RMS of the pairwise differences of the two data sets can serve as a measure how far on average the error is from 0 as shown in below table 2.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Where,
n = number of observed values,
y_j = input values
y_j^ = predicted output values

Table 2: Output Result

Savoy_selling_Price	predicted1	predicted 2
244	244.7052629	232.485742
226	212.2127205	258.841885
544	575.3172563	514.475422
398	405.9309495	406.421231
398	373.6915769	389.190495
398	367.0089771	390.176706
106.2	118.6508992	101.248141
117	132.0265618	117.955834
310	340.6640461	321.737381

We got RMSE as **0.15** which is **99.85%** accurate.

Conclusions and Future Work

The proposed approach described here considered multiple purchase attributes to determine product price dynamically by using back propagation neural network. We simulated an e-commerce market place with 200 buyers, three sellers where all the sellers trying to sell a product P. The experimental results showed that the seller employing our model earned higher revenue than that of earned by other two sellers who followed simple pricing algorithm and derivative-following strategies.. Our model made an assumption that sellers have limited prior knowledge about market parameters in setting the initial price of the products. We would like to eliminate the assumption from our model by employing a web crawler tool in our application in order to learn the information on prices set by other competitive sellers in the market. Using this information we may set the initial price of the products such that the sellers no need to initialize the product prices while using our model.

REFERENCES

[1] Kephart, J., Brooks, C., Das, R.: Pricing information bundles in a dynamic environment. In: ACM Conference on Electronic Commerce 2001, pp. 180–190 (2001)

[2] Pabitra Kr. Dey, Abhijit Banerjee, Dipendra Nath Ghosh, Abhoy Chand Mondal : AHP-Neural Network Based Player Price Estimation in IPL(2007)

[3] Tesauro, G., Kephart, J.: Foresight-based pricing algorithms in agent economies.Decision Support Systems 28(1-2), 49–60 (2000)

[4] Russell, J. Stuart, Peter Norvig: Artificial Intelligence: A modern Approach, Second Edition, Prentice Hall (2005)

[5] J.M. Yatim, S.H. Tapir and F. Usman: Evaluation of Building Performance Using Artificial Neural Network: Study on Service Life Planning in Achieving Sustainability(2006)

[6] Tapu Kumar Ghose and Thomas T. Tran: A Dynamic Pricing Approach in E-Commerce Based on Multiple Purchase Attributes(2010)

[7] Gallego, G., Ryzin, G.: Optimal dynamic pricing of inventories with stochastic demand over finite horizons. Manage. Sci. 40(8), 999–1020 (1994)

[8] Dasgupta, P., Hashimoto, Y.: Multi-attribute dynamic pricing for online marketsusing intelligent agents. In: AAMAS (2004)

[9] Li, C., Wang, H., Zhang, Y.: Dynamic pricing decision in a duopolistic retailing market. In: Proceedings of the 6th World Congress on Intelligent Control andAutomation, Dalian, China (June 2006)

[10] Chinthalapati, V., Yadati, N., Karumanchi, R.: Learning Dynamic Prices in MultiSeller Electronic Retail Markets With Price Sensitive Customers, Stochastic Demands, and Inventory Replenishments. IEEE, Los Alamitos (2006)

[11] Carvalho, A., Puterman, M.: Dynamic pricing and reinforcement learning. IEEE, Los Alamitos (2003)

[12] Kong, D.: One Dynamic Pricing Strategy in Agent Economy Using Neural Network Based on Online Learning. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (2004)