

A Mixed Process Neural Network and its Application to Churn Prediction in Mobile Communications*

Guojie Song[†] Dongqing Yang[†] Ling Wu[†] Tengjiao Wang[†] Shiwei Tang[‡]

[†]Department of Computer Science, Peking University, Beijing, P.R. China

[‡]National Laboratory on Machine Perception, Peking University, Beijing, P.R. China

{gjsong, dqyang, wuling, tjwang, tsw}@pku.edu.cn

Abstract

Churn prediction is an increasingly pressing issue in today's ever-competitive commercial environments, especially in mobile communication arena. In this paper, a Mixed Process Neural Network (MPNN) based on fourier orthogonal base function has been proposed to support churn management, which can deal with both static value and time-varied continuous value simultaneously. To further improve its performance, an optimized network, c-MPNN, has been presented, which adopts fourier expansion based preprocessing and hidden layer combination techniques to optimize MPNN's structure. Most important of all, our method has been used in real applications in China Mobile. Experiments based on the real datasets also show that our proposed churn prediction method has good maneuverability and performance.

1. Introduction

Customer churn refers to the loss of subscribers switching from one service provider to another, which can lead to profits reduction and decrease shareholders value greatly. In the mobile communications industry, the annual churn rate reaches 25% in Europe and 30% in the United States [1]. It costs about five times as much to sign on a new subscriber as to retain an existing one [2]. However, given that most customers will only signal their intention to churn when they call to cancel their account, it is difficult using standard techniques to target for anti-churn marketing. Thus, an ideal solution to combating churn is to build a churn prediction model by using data mining techniques to predict customer churning behaviors, which are becoming more than necessary for the success of the enterprise, especially in the

highly competitive mobile communications industry.

In this paper, a mixed process neural network (MPNN) has been proposed, which can deal with both traditional static data and the time-varied continuous data simultaneously. Fourier orthogonal base function has been chosen as the base function of process neuron to expand each time varied continuous series efficiently. To avoid repeating computation of fourier base function expansion and time aggregation operation, an optimized MPNN, named c-MPNN, has been proposed, which use the preprocessing technique based on fourier transform to simplify the structure of MPNN. Some optimization techniques for improving maneuverability of the training process have also been proposed. The most important thing is that the proposed churn prediction model has been used in real applications in some provinces of China mobile communications. The effectiveness and efficiency of MPNN and c-MPNN have also been proved by our extensively experiments based on the real dataset.

Related works: Mozer et al. built models to predict the probability of a subscriber churning within a period of time [4]. Yan et al. built models to improve prediction of customer churn in non-stationary environments [10]. Lu provide examples of reasons to identify customer initiated churn [7]. Yan et [3] proposed a detail framework for churn prediction. Rosset and Neumann calculated customer value and identified potential churn among valuable customers [9]. However, all these methods accept static value only.

Since He and Liang proposed artificial process neuron model [5], it has been used in several applications, mainly in the area of automatic diagnoses [6] and petroleum geologic [8]. However, there is no research using PNN to churn prediction in mobile communications. This study is the first attempt as far as we known.

The remaining of the paper is organized as follows. In Section 2, some related background knowledge has been prepared. Section 3 introduce the MPNN model. Some op-

*This work is supported by the National Natural Science Foundation of China under Grant No. 60473051

timization techniques are discussed in Section 4. A performance study of the proposed methods is demonstrated in Section 5, and we conclude our study in Section 5.

2. MPNN: A MIXED PNN MODEL

Assume that the input of process neural network $X(t)$ is $(x_1(t), x_2(t), \dots, x_n(t))$, and the expected output is $y(t)$, where $x_i(t), y(t) \in C[0, T]$. $b_1(t), b_2(t), \dots, b_L(t), \dots$ are base functions in $C[0, T]$. According to weierstrass approximation theorem [?], given any $\varepsilon > 0$, there exists a polynomial $P(t)$ on $[0, T]$ such that $|x_i(t) - P(t)| < \varepsilon$ for all $t \in [0, T]$. In other words, any $x_i(t) \in C[0, T]$ can be uniformly approximated on $[0, T]$ by polynomials $P_l(t) (l = 1, 2, \dots)$ to any degree of accuracy. Suppose that the set of $P_l(t)$ is independent. There is a relationship between orthogonality and independence. It is possible to convert a set of independent functions $P_l(t)$ into the set of orthogonal functions $b_l(t)$ that spans the same space. Thus $y(t)$ can be expressed as formula 1.

$$y(t) = \sum_{l=1}^L c_l b_l(t) \quad (1)$$

2.1. Topological structure of MPNN

The MPNN proposed in this paper is composed of four layers: input layer, two hidden layers and one output layer.

The first layer is input layer, which is composed of $n + \sum_{i=1}^m A_i$ units. In churn prediction model, the input includes both continuous time-varied data, such as billing data, and discrete data, such as customer gender etc. Thus, the input of MPNN include n input node dealing with continuous time series data and $\sum_{i=1}^m A_i$ nodes for discrete static input data, where each input node d_{i,A_i} corresponds to one value of property A_i , with 1 if $value(A_i) = d_{i,A_i}$ else with 0.

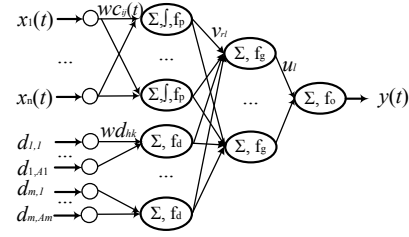
The second layer is the first hidden layer, which is composed of n process neurons and m traditional neurons. Process neurons deal with n time continuous input data and traditional neuron only accept $\sum_{i=1}^m A_i$ discrete static data from the input layer.

The third hidden layer is composed of p traditional neurons and the last layer is output layer. To reduce the complexity, we will only consider the case of one output unit. The topological structure of MPNN is depicted in Figure 1.

2.2. Relationship between input and output

Input: The inputs of the first layer can be expressed as two parts: $X(t)$ for time varied continuous time series and D for discrete static input, denoted as

$$X(t) = (x_1(t), x_2(t), \dots, x_n(t))$$



Topological structure of MPNN

$$D = (d_{1,1}, \dots, d_{1,A_1}, \dots, d_{m,1}, \dots, d_{m,A_m})$$

The outputs of the first hidden layer: We first consider the computation of j -th process neuron connected with continuous input $X(t)$. $y_j^{(c,1)} = f_p(\sum_{i=1}^n \int_0^T w_{c_{ij}}(t) x_i(t) dt)$ where $w_{c_{ij}}(t)$ is the link weight function between j -th process neuron in the first hidden layer and the i -th unit of $X(t)$ in the input layer ($i, j \in [1, n]$).

According to formula 1, if we take fourier base function $s_i(t), i \in [1, L]$, as process base function to expand process weight function and input $x_i(t)$, we have

$$y_j^{(c,1)} = f_p\left(\sum_{i=1}^n \sum_{q=1, z=1}^{q=L, z=L} c_{iz}^{(z)} w_{c_{iq}}^{(q)} \int_0^T s_q(t) s_z(t) dt\right)$$

Thus, the outputs of the process neurons in the first hidden layer can be expressed as $y_j^{(c,1)} = \sum_{i=1}^n \sum_{q=1, z=1}^{q=L, z=L} c_{iz}^{(z)} w_{c_{iq}}^{(q)}$ where f_p is the activation function of the process neurons in the first hidden layer.

For traditional neurons in the first layer, its outputs can be expressed as $y_k^{(d,1)} = f_d(\sum_{h=1}^{A_k} d_{kh} w_{d_{hk}})$ where $w_{d_{hk}}$ is the link weight between k -th neuron in the first hidden layer and the h -th value of property d_k in the discrete input layer ($k \in [1, m], h \in [1, A_k]$).

Thus, the outputs of the first hidden layer is

$$out_r^{(1)} = y_j^{(c,1)} + y_h^{(d,1)} \quad (2)$$

where $r \in [1, m + n]$.

The outputs of the second hidden layer: Based on the outputs of the first hidden layer, the outputs of the second hidden layer can be expressed as

$$y_l^{(2)} = f_g\left(\sum_{r=1}^{m+n} out_r^{(1)} v_{rl}\right) \quad (3)$$

where v_{rl} is the link weight between the l -th neuron in the second hidden layer and the r -th neuron in the first hidden layer. f_g is the activation function of the neuron in the second hidden layer.

Output: The output function of the MPNN can be expressed as

$$y(t) = f_o\left(\sum_{l=1}^p y_l^{(2)} u_l\right) \quad (4)$$

where u_l is the link weight between the l -th neuron in the second hidden layer and the output node. f_o is the activation function of the output node.

2.3. Learning algorithm

Assume that we have K learning sample functions:

$$\begin{bmatrix} x_{11}(t), & x_{12}(t), & \dots, & x_{1n}(t), & y_1(t) \\ x_{21}(t), & x_{22}(t), & \dots, & x_{2n}(t), & y_2(t) \\ \vdots & \vdots & & \vdots & \vdots \\ x_{K1}(t), & x_{K2}(t), & \dots, & x_{Kn}(t), & y_K(t) \end{bmatrix}$$

where the first suffix i in $x_{ij}(t)$ denotes the serial number of learning sample, and the second suffix j denotes the serial number of component in input function vector. $y_k(t)$ is the expected output function for input $x_{k1}(t), x_{k2}(t), \dots, x_{kn}(t), (k \in [1, K])$.

Suppose that $\tilde{y}(t)$ is the desired output function, and $y(t)$ is the corresponding actual output function of the MPNN, then the mean square error of the MPNN output can be written as

$$\begin{aligned} E &= \frac{1}{2} \sum_{b=1}^K (y_b(t) - \tilde{y}_b(t))^2 \\ &= \frac{1}{2} \sum_{b=1}^K \left[f_o \left(\sum_{l=1}^p y_{lb}^{(2)} u_l \right) - \tilde{y}_b(t) \right]^2 \\ &= \frac{1}{2} \sum_{b=1}^K \left\{ f_o \left[\sum_{l=1}^p f_g \left(\sum_{r=1}^{m+n} (f_p \left(\sum_{i=1}^n \sum_{q=1, z=1}^{q=L, z=L} c_{izb}^{(z)} w_{iq}^{(q)} \right) \right. \right. \right. \\ &\quad \left. \left. \left. + f_d \left(\sum_{h=1}^{A_k} d_{khh} w_{dhk} \right) v_{rl} \right) u_l \right] - \tilde{y}_b(t) \right\}^2 \end{aligned} \quad (5)$$

For analysis convenience, Z_b, Q_b, P_b and H_b are defined respectively as

$$\begin{aligned} H_b &= \left(\sum_{h=1}^{A_k} d_{khh} w_{dhk} \right), & P_b &= \sum_{i=1}^n \sum_{q=1, z=1}^{q=L, z=L} (c_{izb}^{(z)} w_{iq}^{(q)}) \\ Q_b &= \sum_{r=1}^{m+n} (f_d(H_b) + f_p(P_b)) v_{rl}, & Z_b &= \sum_{l=1}^p Z_b u_l \end{aligned}$$

According to the gradient descent method, the learning rules are defined as follows

$$\begin{aligned} w_{iq}^{(q)} &= w_{iq}^{(q)} + \alpha \Delta w_{iq}^{(q)} \\ w_{dhk} &= w_{dhk} + \beta \Delta w_{dhk} \\ v_{rl} &= v_{rl} + \gamma \Delta v_{rl} \\ u_l &= u_l + \eta \Delta u_l \end{aligned} \quad (6)$$

where $\alpha, \beta, \gamma, \eta$ are the learning rate, and $i \in [1, n], k \in [1, m], h \in [1, A_k], l \in [1, p], q, r \in [1, L]$.

$\Delta w_{ik}^{(k)}, \Delta w_{dhk}, \Delta v_{ik}$, and Δu_k can be calculated as follows

$$\begin{aligned} \Delta w_{iq}^{(q)} &= -\frac{\partial E}{\partial w_{iq}^{(q)}} = -\Omega u_l f'_g(Q_b) v_{rl} f'_p(P_b) c_{izb}^{(z)} \\ \Delta w_{dhk} &= -\frac{\partial E}{\partial w_{dhk}} = -\Omega u_l f'_g(Q_b) v_{rl} f'_d(H_b) d_{khh}^{(z)} \\ \Delta v_{rl} &= -\frac{\partial E}{\partial v_{rl}} = -\Omega u_l f'_g(Q_b) \\ \Delta u_l &= -\frac{\partial E}{\partial u_l} = -\Omega \end{aligned} \quad (7)$$

where $\Omega = \sum_{b=1}^K (f_o(Z_b) - \tilde{y}_b(t)) f'_o(Z_b)$.

In this paper, all activation functions have been substituted by Sigmoid function, i.e. $f_o(u) = f_g(u) = f_p(u) = f_d(u) = \frac{1}{1+e^u}$, with $f'_*(u) = f_*(u)(1 - f_*(u))$.

3. The OPTIMIZATION TECHNIQUES TO CHURN PREDICTION

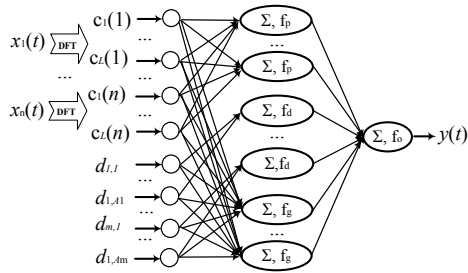
In this section, we present two novel optimization techniques for improving the performance of churn prediction based on MPNN model, including topological structure optimization and training process optimization.

3.1. Topological Structure Optimization

The proposed optimization techniques of topological structure includes two aspects: *preprocess optimization* and *network layer optimization*.

Preprocess optimization: Based on formula 2.2, if we take fourier base function as the base function of process neuron in MPNN, each input time varies function $x_i(t)$ can be expanded with a set of fourier coefficient c_i and corresponding weight w_i . If we take $x_i(t)$ as input of process neuron in MPNN during each time iteration of training process, it should be expanded one time by using DFT accompanied with one time time aggregation operation by using formula $\int_0^T s_q(t) s_z(t) dt$. In fact, such costs of the fourier expansion can be avoided if each input time series $x_i(t), i \in [1, n]$, has been preprocessed before it is input into MPNN. Because c_i is a constant and has no relationship with the training process, so taking fourier coefficient c_i as input will not influence the final results. By using such preprocessing strategy, the process neurons in MPNN have become a traditional neurons by losing its time aggregation ability.

Network layer optimization: It is well known that the efficiency of neural network is inverse proportion to the num-



Topological structure of c -MPNN

ber of neurons. To reduce the number of neurons, the meaning of each neurons in MPNN should be analyzed. MPNN includes four kinds of neuron, denoted as f_p , f_d , f_g and f_o respectively. f_p is connected with the properties of customer's consuming history, expressed as time varied function; f_d accept the inputs of customer's basic information, expressed as enumerated values; The outputs of f_p and f_d will be input into f_g , and f_o can predict whether customer will churn or not. f_p and f_d provide the partial customer's information, but f_g can provide the complete information for f_o . If we input the outputs of f_p and f_d into f_o directly, and move the neurons of f_g into the first hidden layer but with less numbers of neurons q ($q < p$), neurons f_o not only can accept partial information and complete information together, but also can reduce the number of neurons in MPNN. The efficiency can be improved greatly with the reduction of neurons in MPNN can keep the accuracy of prediction results simultaneously, which has been proved by our experiments.

By using above two optimization techniques, MPNN model has been condensed into another new model, named c -MPNN, which includes three layers: input layer, hidden layer and output layer. The topological structure is illustrated as Figure 2.

4. DATA And EXPERIMENTAL RESULTS

4.1. Characteristics of Input Data

Ultimately, churn occurs because subscribers are dissatisfied with the price or quality of service, usually as compared to the offerings of competing carriers, which can be embodied by the information of their usage. For the purpose of this study, we prepare the real data from the China Mobile Communication Company. We sampled the data from Jan. 2004 to April 2004. The data set covers demographic variables and the variables about telephone usage. After filtering the data with missing values, we select 220 thousands samples. 2000 samples have been selected randomly for training data set and 10000 samples for testing data set. The description of the variables for this research is presented as

follows.

Time varied continuous data: It includes three kinds of usage series data: *call time*, *the number of messages* and *the number of different telephones communicated with him(her)*. Each data element in series is accumulated in term of day, and 91 element spanning three months have been generated in each series.

Traditional static discrete data: Two discrete variables have been selected, which are customer's *gender* (male:0, female:1) and *age* (being discretized into five segments, (0-20), (20-30), (30, 40), (40, 60) and (60-100) in advance).

Mark of churn: According to the definition of churn management strategies, different in each province, the mark of churn for each customer can be defined with whether he is churn in (churn:1, nonchurn:0), that is in June 2004.

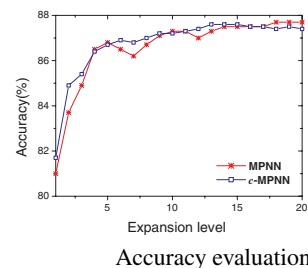
4.2. Experimental Results

All experiments were performed on Pentium IV 1.4G with 512MB main memory, running Windows XP. The MPNN code was written in C++.

We tested the performance of MPNN and its optimized version c -MPNN about predicting accuracy, training and testing time. The model predicts correctly with churn if the predicted user's churn probability is bigger than (or equal to) 0.5 or the user doesn't churn with the probability less than 0.5. Otherwise the Model predicts wrong. We demonstrate the accuracy by the percentage of correctly predicting users.

4.3. Accuracy evaluation

We increase the level of fourier expansion for continuous serial attributes from 1 to 20, and evaluate the accuracy of MPNN and c -MPNN at different expansion level. The results are the average value of 10 independent experiments. Figure 3 shows the accuracy of the two models.

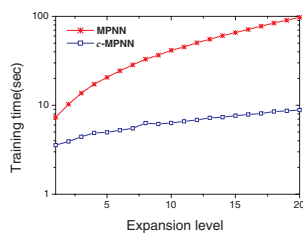


As Figure 3 shows, the accuracy of c -MPNN is almost equals to that of MPNN or even sometimes better than it. Also c -MPNN is more stable in that the curve is smoother. So c -MPNN is less influenced by random error than MPNN.

4.4. Efficiency evaluation

4.4.1 Training time evaluation

In practice of mobile churn prediction, the characteristic of user activity changes frequently, thus the predicting model should be updated correspondingly. Therefore the time consumed in training process is an important factor of time efficiency. Here we fix the iteration times with 1000. Notice that the training time is expressed using logarithm.

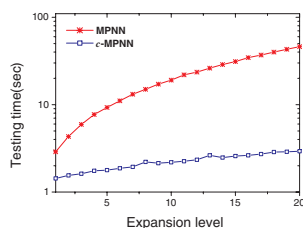


Training time evaluation

As shown in Figure 4, the training time of MPNN increases rapidly according to the number of neurons and up to one magnitude bigger than *c*-MPNN soon. For example, as fourier expansion level is 5, the training time of MPNN is 20.63 second, while *c*-MPNN is only its 1/4 , 4.97 second.

4.4.2 Testing time evaluation

The amount of data used by churn prediction model in mobile communication is very huge, can up to 5,000,000 records (about 12Gbytes) in each province. So the testing time of Churn model is also an important factor. The evaluation result has been shown as follows. Also the testing time is expressed using logarithm, too. The test time of



Testing time evaluation

MPNN is much longer than *c*-MPNN. With the increase of Fourier expansion level, the difference grows to a magnitude. For example, as the Fourier expansion level up to 5, the test time of MPNN is 9.29 seconds, while *c*-MPNN is only about 1.78 seconds.

4.4.3 Conclusions

MPNN is a mixed process neural network, which accepts the inputs consisted of traditional static discrete data and

time varied continuous series together. To simplify the learning algorithm, fourier orthogonal basis function has been selected to expand the input functions. The preprocessing process has been adopted to optimized the network structure with a new process neural network *c*-MPNN. Most important thing is that our system has been applied to serve for mobile communication, and its performance has been proved in real scenario.

References

- [1] Anderson Consulting, Battling Churn to Increase Shareholder Value: Wireless Challenge for the Future, Anderson Consulting Research Report, 2000.
- [2] Ding-An Chiang, Yi-Fan Wang, Shao-Lun Lee, Cheng-Jung Lin, Goal-oriented sequential pattern for network banking churn, Expert Systems with Applications 25 (2003) 293-302.
- [3] Yan, L., Wolniewicz, R.H., Dodier, R. Predicting Customer Behavior in Telecommunications. IEEE Intelligent Systems, Vol. 19, 2 (2004) 50-58.
- [4] Mozer, M.C., Wolniewicz, R., Grimes, D.B.: Predicting Subscriber's Dissatisfaction and Improving Retention in the Wireless Telecommunications Industry. IEEE Transactions on Neural Networks, Vol. 11, 3 (2000) 690-696.
- [5] He X.G., Liang J.Z. Some Theoretical Issues on Process Neural Networks. Engineering Science, 2 (2000) 40-44.
- [6] HE X.G. , XU S.H. A feedback process neuron network model and its learning algorithm[J] . Acta Automatica Sinica , 2004 , 30 (6) :801 - 806.
- [7] Lu, J., 2002. Predicting customer churn in the telecommunications industry—An application of survival analysis modeling using SAS® SAS Group International 27th Annual Conference, paper 114.e
- [8] XU S.H , HE X.G. Research and applications of self organization process neural networks[J] . Journal of Computer and Development , 2003 , 40 (11) : 1612 - 1615.
- [9] Rosset, S. and Neumann, E., 2003. Integrating Customer Value Considerations into Predictive Modeling. Third IEEE International Conference on Data Mining.
- [10] Yan, L. Miller, D.J. Mozer, M.C. and Wolniewicz, R. Improving prediction of customer behavior in nonstationary environments. Proc. of the International Joint Conference on Neural Networks. 3(2001), 2258-2263.