Application of Multi-Granularity Temporal Features in Churn Prediction: Postprint

Authors: Shi Hongbin, Yan Jianfeng, Bai Ruirui, Xu Caixu, Xu Guanggen

Date: 2018-05-02T00:00:00+00:00

Abstract

Telecommunications operators have developed diverse churn prediction models for different scenarios to identify potential churning customers. Existing churn prediction models typically select a single temporal granularity for feature extraction, and subsequently employ machine learning algorithms to model the extracted data. Such approaches focus solely on the model' s impact on classification performance, inadequately considering the role of data. To address these limitations, we propose a method that leverages multiple temporal granularities for feature extraction and attempts to fuse features of different granularities at different stages of model training. Experimental results demonstrate that models trained with multi-granularity feature extraction significantly outperform those trained with single-granularity features.

Full Text

Preamble

Application of Multi-Grain Temporal Features in Churn Prediction

Shi Hongbin, Yan Jianfeng, Bai Ruirui, Xu Caixu, Xu Guanggen
(School of Computer Science & Technology, Soochow University, Suzhou, Jiangsu 215006, China)

Abstract: Telecom operators have developed multiple churn prediction models to identify potential churners across different scenarios. Existing churn prediction models typically select a single time granularity for feature extraction, followed by modeling with machine learning algorithms. Such approaches focus solely on the model' s impact on classification performance without fully considering the role of data. To address this limitation, we propose a method that extracts features using multiple time granularities and attempts to fuse features

of different granularities at various stages of model training. Experimental results demonstrate that models trained with multi-grain features significantly outperform those trained with single-granularity features.

Key Words: churn prediction; time series data; multi-granularity

0 Introduction

In recent years, user churn has become a critical issue for telecom operators, attracting widespread attention from both industry and academia. Figure 1 [Figure 1: see original paper] shows the churn rate statistics for 2G/3G prepaid users of a Shanghai operator from September 2015 to August 2016. Active users are defined as those whose monthly total call duration, total internet traffic, or total number of sent text messages exceed certain thresholds, generating greater commercial profits for operators. Inactive users are those who fail to meet these thresholds across all three metrics, contributing less profit. As shown in Figure 2 [Figure 2: see original paper], active users account for a relatively high proportion of approximately 63%. Since churn among inactive users can be predicted relatively accurately through threshold-based methods and has minimal impact on operator profits, focusing on active user churn is of significant importance. Research indicates that the cost of acquiring a new customer is more than three times that of retaining an existing one, while the success rate of retention strategies exceeds that of new customer acquisition [1-3]. This paper focuses on the active user population, researching and implementing a churn prediction model based on multi-granularity temporal features to provide a scientific basis for targeted retention strategies, thereby maximizing user retention and profit growth.

From the operator' s perspective, users have two states: in-network and churned. A user is considered in-network if they generate revenue for the operator. When a user no longer generates any revenue, they are considered churned. Churn prediction involves analyzing historical behavioral data to identify patterns and predict future user behavior. Current churn prediction research primarily improves performance through three aspects: constructing useful features, building effective classifiers, and classifier ensemble [4,5]. In the feature construction stage, researchers typically extract features using a single time granularity, such as monthly [6,7], daily [8], or 2-hour intervals [9]. Monthly granularity involves partitioning time by month and statistics such as monthly bills, call volumes, and text message counts. Daily and hourly granularities follow similar principles. In the classifier construction stage, common algorithms include neural networks [6-9], logistic regression [10-12], support vector machines [13-15], decision trees [14-17], and K-nearest neighbors [18]. For instance, reference [6] constructs churn prediction models using neural networks with three architectures: a two-layer fully connected network, a three-layer fully connected network, and a network with one convolutional-pooling layer and two fully connected lay-

ers. Ensemble learning is recognized as an effective method for improving model performance, with applications in churn prediction including support vector machine ensembles [19-21] and decision tree ensembles [22]. Reference [19] trains multiple different support vector machines and fuses them, while reference [22] trains multiple different decision trees and combines them. In summary, most existing work focuses on classifier design and model fusion, while typically selecting only a single time granularity for feature extraction, rarely considering the simultaneous use of multiple time granularities [23].

To address these limitations, this paper proposes a churn prediction method based on multi-granularity temporal feature fusion. During the model training stage, features are categorized as either invariant or varying based on whether their values change over time, with varying features further distinguished by monthly and daily granularities. Different combinations form distinct datasets, which are trained using Random Forest and GBDT to obtain six models. In the model fusion stage, both averaging and Stacking methods are employed to fuse different models, ultimately improving the Top 25000 precision to 0.5696.

1.1 Base Models

1.1.1 Random Forest Random Forest, proposed by Breiman in 2001 [24], is a variant of Bagging. Bagging constructs datasets by sampling with replacement from the original dataset, repeating this process to obtain multiple datasets, training a base classifier on each, and finally combining them through voting. Random Forest introduces random attribute selection on top of Bagging. Assuming a sample has attributes, each base decision tree in the Random Forest selects attributes for splitting leaf nodes, typically choosing . By simultaneously introducing sample perturbation and attribute perturbation, Random Forest significantly improves the generalization capability of base learners.

1.1.2 GBDT GBDT, like Random Forest, trains multiple decision trees. However, unlike Random Forest where trees can be trained simultaneously without dependencies, GBDT trains each tree sequentially, with each tree depending on the output of previous trees. Let denote the predicted value for the -th sample in GBDT, which can be expressed as:

where is the number of trees and represents the output of the -th tree. For GBDT, the objective function is:

where is the loss function and is the regularization term. Expanding the loss function using Taylor series at yields:

where and are defined as:

From the Taylor expansion, it is evident that training the -th tree depends on the previous trees.

1.2 Model Fusion Methods

Model fusion combines multiple single models to further improve performance. Let denote the output of the -th model and represent the number of models. This paper employs two fusion methods:

1.2.1 Averaging The final output of the model is:

That is, after obtaining the prediction probabilities from each model, their average is taken as the final output.

1.2.2 Stacking The Stacking procedure is as follows:

Algorithm: Stacking
Input: Training set , base learning algorithm , secondary learning algorithm
Process:
1. Split training set into and
2. Train model using and
3. Use and to train model
4. Train model using and

This paper uses logistic regression as the secondary learning algorithm. During prediction, base learners generate predictions on the dataset, which serve as input to the secondary learner whose predictions become the final output.

2 Churn Prediction Model Based on Multi-Granularity Temporal Features

The proposed churn prediction model framework consists of four components: data preparation, feature extraction, model training, and model fusion, as illustrated in Figure 3 [Figure 3: see original paper]. To validate the framework' s generality, different methods are experimented with during the model training and fusion stages.

2.1 Data Preparation

The telecom operator' s data platform generates approximately 2.3 TB of data daily, including BSS (Business Support System) and OSS (Operation Support System) data. BSS data comprises user basic information, user behavior, billing information, voice data, SMS data, and call details, generating about 24 GB per day. Currently, BSS supports 2G/3G/4G prepaid and postpaid users. This study focuses on 2G/3G prepaid active users, selecting 10 tables from over thirty available tables as data sources, including daily user basic information table, monthly user basic information table, daily user behavior table, package table, monthly user balance table, user voice call detail table, user SMS detail table, monthly bill table, terminal table, and recharge table. The training set contains 1,415,429 records, while the test set contains 1,380,154 records.

Datasets are constructed using a sliding window approach, with time windows for each feature shown in Table 3 . The training set uses monthly-granularity features from January 2015 to December 2015 (12 months) and daily-granularity features from October 1, 2015 to December 31, 2015 (92 days), with labels from January 2016 indicating user churn or retention. The test set uses monthly-granularity features from February 2015 to January 2016 (12 months) and daily-granularity features from November 1, 2015 to January 31, 2016 (92 days), with labels from February 2016. Daily-granularity features provide finer granularity and can more precisely describe recent user behavior, but using too many days leads to excessive features and training difficulties. Monthly-granularity features are coarser with fewer features, better suited for capturing long-term user behavior trends. Using 12 months of monthly features and 92 days of daily features simultaneously leverages both advantages without introducing excessive features.

2.2 Feature Extraction

Features are categorized into three types: invariant features, monthly features, and daily features. Invariant features are those whose values do not change over time or change linearly, such as gender and age extracted from user basic information. Monthly features are extracted on a monthly basis, such as monthly bills, call duration, and internet traffic. Daily features are extracted daily, such as daily bills, call duration, and internet traffic. When extracting monthly features, for data already at monthly granularity, values are directly used; for daily-granularity data, the monthly average is computed as the monthly feature. This study extracts 3 invariant features, 52 monthly features per month, and 34 daily features per day. Table 2 shows sample extracted features.

2.3 Model Training

Tree-based models are widely used in industry, with references [14-17] demonstrating their effectiveness in churn prediction tasks. Additionally, tree models can perform feature selection to some extent, reducing redundancy risks when fusing features of different time granularities. Among tree models, Random Forest and GBDT are particularly effective, so this paper employs both for training.

Six models are trained using these two algorithms on the three datasets. To ensure fairness, identical parameters are used for all three Random Forest models and all three GBDT models. For Random Forest, 200 trees are used. For GBDT, the learning rate is set to 0.1, maximum tree depth to 8, and L2 regularization to 50, with other parameters using toolkit defaults.

2.4 Model Fusion

Stacking is a general and effective fusion method, used in reference [25] for churn prediction tasks. Averaging is another simple yet effective method, employed

here for comparison. Let denote Averaging fusion and denote Stacking fusion. For the six models trained, 14 fusion approaches are experimented with from two perspectives: 1) fusing monthly and daily granularity features, and 2) fusing classifiers. and attempt direct concatenation of the two granularity features to train a single classifier. Classifier fusion includes fusing identical classifiers and fusing different classifiers. When using Stacking, 90% of the training data is used for base learners and 10% for the secondary learner.

The specific fusion approaches are:

1. **Monthly and daily granularity feature fusion**
   - Use prediction results as final output ( )
   - Use prediction results as final output ( )
2. **Averaging identical classifiers**
   - Average predictions of ( )
   - Average predictions of ( )
   - Average predictions of ( )
   - Average predictions of ( )
3. **Stacking identical classifiers**
   - Stack predictions of ( )
   - Stack predictions of ( )
   - Stack predictions of ( )
   - Stack predictions of ( )
4. **Averaging different classifiers**
   - Average predictions of and ( )
   - Average predictions of , , and ( )
   - Average predictions of and ( )
   - Average predictions of , , and ( )
5. **Stacking different classifiers**
   - Stack predictions of and ( )
   - Stack predictions of , , and ( )
   - Stack predictions of and ( )
   - Stack predictions of , , and ( )

3 Experiments

Experiments use precision, recall, and AUC as evaluation metrics. Since correct prediction is more important than complete prediction in churn prediction, precision is the primary focus, with recall and AUC as references. The confusion matrix is shown in Table 4 .

Precision (P) and recall (R) are defined as:

AUC is the area under the ROC curve, where the vertical axis is the True Positive Rate (TPR) and the horizontal axis is the False Positive Rate (FPR), defined as:

In experiments, precision and recall for Top 25000 are calculated by sorting

predicted probabilities in descending order, labeling the top 25,000 samples as positive and the rest as negative, then computing the metrics.

3.1 Baseline Performance

Random Forest and GBDT are trained on and datasets, yielding four models: , , , and . Models trained on the same time granularity are fused using averaging or Stacking as baselines for comparison with the 14 fusion methods. Tables 5 through 10 show the experimental results.

Table 5 shows baseline performance. At Top 25000, Random Forest achieves better precision and recall than GBDT, while GBDT has superior AUC. Random Forest performs better on monthly-granularity features, whereas GBDT excels on daily-granularity features. When fusing single-granularity models, monthly-granularity features outperform daily-granularity features.

3.2 Experimental Results

3.2.1 Feature Granularity Fusion Table 6 shows results of directly concatenating monthly and daily granularity features. Both and show significant improvements in precision and recall, with modest AUC improvements. achieves higher precision than four baseline multi-model fusion methods and two other fusion approaches without multi-model fusion, demonstrating that direct concatenation of monthly and daily features substantially improves model performance.

3.2.2 Classifier Fusion Table 7 compares averaging-based fusion. Averaging and improves performance over individual models, raising precision from 0.5174 for and 0.5072 for to 0.5507. Adding further improves performance to 0.5545, surpassing two baseline averaging methods (0.5286 and 0.5142). For GBDT, the two averaging fusion methods perform comparably to baselines with less improvement than Random Forest, indicating that averaging diverse base classifiers reduces generalization error and improves performance.

Table 8 shows Stacking-based fusion of identical classifiers, similar to Table 7 but replacing averaging with Stacking. Stacking improves Random Forest performance but yields slightly worse results for GBDT. For the same base classifiers, ensemble approaches (both averaging and Stacking) improve generalization over individual classifiers.

Tables 9 and 10 present fusion results for different classifier types. Averaging and raises precision from 0.5545 for and 0.5449 for to 0.5696. Stacking improves precision from 0.5585 for and 0.5246 for to 0.5624, with improvements also observed in recall and AUC. This demonstrates that fusing different types of classifiers yields better performance, with averaging outperforming Stacking.

3.2.3 Impact of Classifier Quantity Results in Tables 5-10 show that final model performance improves with increasing classifier quantity. Using GBDT

with simple averaging as an example (Figure 5 [Figure 5: see original paper]), Top 25000 precision, recall, and AUC for improve over by 0.0594, 0.0294, and 0.0135 respectively. Adding improves these metrics by 0.0821, 0.0407, and 0.018 over alone, while improves over by 0.0994, 0.0493, and 0.0148. Using improves over by 0.1068, 0.0575, and 0.016.

3.2.4 Impact of Different Classifiers Tables 5-8 show that with fixed input and fusion methods, Random Forest achieves better Top 25000 precision and recall than GBDT, while GBDT has superior AUC. Since precision is the primary focus, Random Forest performs slightly better. Regardless of classifier choice, using both monthly and daily granularity features simultaneously outperforms using a single granularity.

3.2.5 Impact of Different Fusion Methods Tables 7-10 demonstrate that averaging achieves better performance than Stacking on this dataset. Regardless of fusion method, using both monthly and daily granularity features simultaneously outperforms single-granularity training.

4 Conclusion

Traditional churn prediction models select single time granularity features for training, focusing only on model impact without considering data perspective. This paper proposes fusing features of different time granularities to form three distinct datasets, training Random Forest and GBDT models, and obtaining final results through model fusion. Experiments demonstrate that this approach improves precision from 0.4628-0.5147 to 0.5696. Multi-granularity temporal features essentially perform feature engineering through aggregation at different time granularities—a simple yet effective approach. This paper validates the effectiveness of multi-granularity temporal features for churn prediction and proposes a corresponding framework.

Currently, only monthly and daily granularities are considered. Future research will incorporate more time granularities and explore more complex fusion methods, such as constructing multi-level fusion models, to further improve performance.

References

[1] Coussement K, Van den Poel D. Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques [J]. Expert Systems with Applications, 2008, 34(1): 313-327.

[2] Verbeke W, Dejaeger K, Martens D, et al. New insights into churn prediction in the telecommunication sector: A profit driven data mining approach [J]. European Journal of Operational Research, 2012, 218(1): 211-229.

[3] Reinartz W J, Kumar V. The impact of customer relationship characteristics on profitable lifetime duration [J]. Journal of marketing, 2003, 67(1): 77-99.

[4] Guyon I, Lemaire V, Boullé M, et al. Design and analysis of the KDD cup 2009: fast scoring on a large orange customer database [J]. ACM SIGKDD Explorations Newsletter, 2010, 11(2): 68-76.

[5] Yu H F, Lo H Y, Hsieh H P, et al. Feature Engineering and classifier ensemble for KDD Cup 2010 [C]// Proc of JMLR: Workshop and Conference Proceedings. 2010: 1-16.

[6] Umayaparvathi V, Iyakutti K. Automated feature selection and churn prediction using deep learning models [J]. International Research Journal of Engineering and Technology, 2017, 4(3): 1846-1854.

[7] Castanedo F, Valverde G, Zaratiegui J, et al. Using deep learning to predict customer churn in a mobile telecommunication network [DB/OL]. http://www.wiseathena.com/pdf/wa_dl.pdf.

[8] Wangperawong A, Brun C, Laudy O, et al. Churn analysis using deep convolutional neural networks and autoencoders [J]. arXiv preprint arXiv:1604.05377, 2016.

[9] Zaratiegui J, Montoro A, Castanedo F. Performing highly accurate predictions through convolutional networks for actual telecommunication challenges [J]. arXiv preprint arXiv:1511.04906, 2015.

[10] Stripling E, Van den Broucke S, Antonio K, et al. Profit maximizing logistic regression modeling for customer churn prediction [C]// Proc of IEEE International Conference on Data Science and Advanced Analytics. 2015: 1-10.

[11] Lu Ning, Lin Hua, Lu Jie, et al. A customer churn prediction model in telecom industry using boosting [J]. IEEE Trans on Industrial Informatics, 2014, 10(2): 1659-1665.

[12] Owczarczuk M. Churn models for prepaid customers in the cellular telecommunication industry using large data marts [J]. Expert Systems with Applications, 2010, 37(6): 4710-4712.

[13] Zhao Xi, Shi Yong, Lee Jongwon, et al. Customer churn prediction based on feature clustering and nonparallel support vector machine [J]. International Journal of Information Technology & Decision Making, 2014, 13(05): 1019-1035.

[14] Shaaban E, Helmy Y, Khedr A, et al. A proposed churn prediction model [J]. International Journal of Engineering Research and Applications, 2012, 2(4): 1206-1212.

[15] Abbasimehr H, Setak M, Tarokh M J. A comparative assessment of the performance of ensemble learning in customer churn prediction [J]. Internal Arab Journal Information Technology, 2014, 11(6): 599-606.

[16] Binti Oseman, K, Haris N A, bin Abu Bakar F. Data mining in churn analysis model for telecommunication industry [J]. Journal of Statistical Modeling and Analytics Vol, 2010, 1(19-27).

[17] Kirui C, Hong Li, Cheruiyot W, et al. Predicting customer churn in mobile telephony industry using probabilistic classifiers in data mining [J]. International Journal of Computer Science Issues, 2013, 10(2): 1694-0784.

[18] Idris A, Khan A. Ensemble based efficient churn prediction model for Telecom [C]// Proc of the 12th International Conference on Frontiers of Information Technology. 2014: 238-244.

[19] Coussement K, Van den Poel D. Churn prediction in subscription services: an application of support vector machines while comparing two parameter-selection techniques [J]. Expert Systems with Applications, 2008, 34(1): 313-327.

[20] Verbeke W, Martens D, Mues C, et al. Building comprehensible customer churn prediction models with advanced rule induction techniques [J]. Expert Systems with Applications, 2011, 38(3): 2354-2364.

[21] Kim N, Jung K H, Kim Y S, et al. Uniformly subsampled ensemble (USE) for churn management: Theory and implementation [J]. Expert Systems with Applications, 2012, 39(15): 11839-11845.

[22] Wei C P, Chiu I T. Turning telecommunications call details to churn prediction: a data mining approach [J]. Expert Systems with Applications, 2002, 23(2): 103-112.

[23] Zhang Junbo, Zheng Yu, Qi Dekang. Deep spatio-temporal residual networks for citywide crowd flows prediction [C]// Proc of the 31st AAAI Conference on Artificial Intelligence. 2017.

[24] Breiman L. Random forests [J]. Machine Learning, 2001, 45(1): 5-32.

[25] De Groot D. S. Churn prediction in telecommunication [D]. Delft: Technische University Delft, 2017.

Note: Figure translations are in progress. See original paper for figures.