ABSTRACT

Churn prediction is a popular research area where many methods have been proposed and applied. However, few of the churn prediction methods consider protecting the privacy of customers. We apply data distortion algorithms to the original data to protect customers’ privacy. At the same time the distorted data keeps the main property of the original data so that the accuracy of churn prediction will not be compromised. In this paper, we compare the performance of data distortion algorithms incorporating into a variety of churn prediction methods, and provide suggestions on choosing data distortion algorithms based on security needs.

KEYWORDS: Churn, Prediction, Privacy, Data distortion, Algorithm

INTRODUCTION

Churn prediction is a vital part of customer retention. It predicts whether a current customer will take the business to a competitor or voluntarily terminates service, so marketing campaigns can target at these potential churners for retention efforts. It has been applied to many industries such as telecommunications, banking, web services, etc. (Richter et. al. 2010; Popovic & Basic, 2009; Dror et. al. 2012). Churn prediction requires large amount of qualitative and quantitative customer data that is personal to the customer and sensitive in nature. Therefore, an important issue in churn prediction is how to protect customers’ privacy, especially when the customers’ data is delivered to third parties for churn analysis. Many statistical, machine learning, and data mining techniques have been proposed or adapted for churn prediction, such as regression, neural network, and decision trees (Mozer, Wolnieicz, Grimes, et. al, 2000; Neslin, Gupta, Kamakura, Lu, & Mason, 2006; Hadden, Tiwari, Roy, & Ruta, 2006). But to the best of our knowledge, there is very little research in literature that deals with protecting customers’ sensitive information in churn prediction (Yi et al, 2006).

We propose a strategy to protect customers’ privacy for churn prediction. The customer’s data are preprocessed using data distortion algorithms, and then churn prediction methods are applied to the distorted data. In this way, the original data cannot be derived or reconstructed from the distorted data so that the privacy of customer is preserved, and most importantly, the
distorted data can maintain the critical information of the original data such that the accuracy of churn prediction will not be compromised.

The remainder of the paper is structured as follows. We firstly review the privacy protection methods proposed in literature. Then we describe the classification algorithms used for churn prediction. Next we introduce the data distortion algorithms to be analyzed in the paper, followed by the metrics to evaluate data distortion algorithms. The experimental results are shown. At last, we present the conclusion of the paper.

LITERATURE REVIEW

There are two categories of methods to preserve data privacy. The first category tries to hide the identity of objects in the data set. For example, anonymity (Klosgen, 1995) remove identifiers (e.g. names, social security numbers, addresses, etc.) to protect privacy. However, the released data after removing identifiers may contain information that can be linked with other datasets to re-identify individuals or entities. A popular $k$-anonymity method was proposed to solve this problem (Sweeney, 2002).

The second category of methods tries to hide the data values when the data is sent to a third party for analysis. Data distortion is usually adopted for this type of methods which can avoid the above mentioned privacy leakage. Data distortion algorithms replace the original data value with some disguised value to preserve privacy. Many data distortion algorithms have been proposed in the literature, like Additive Noise, Resampling, Microaggregation, Lossy Compression, Rank Swapping (Domingo-Ferrer & Mateo-Sanz, 2001). Most of these methods are used in statistical databases to maintain some statistical characteristics of datasets, such as mean, sum, variance, etc., but they may not work well in keeping the performance of data mining algorithms like classification and clustering.

CHURN PREDICTION METHODS

There is a rich body of work on churn prediction. Among them, classification techniques are the most widely used. In classification, based on the attribute values of customer record, a customer will be assigned to one of two groups: a group of customers who would churn and a group of customers who would not churn. Hadden et al. provided a thorough review on churn prediction methodologies in (Hadden, Tiwari, Roy, & Ruta, 2007). Besides classification, other methods have also been applied to churn prediction like hazard modeling approach (Jamal & Bucklin, 2006), social network analysis (Dasgupta, Singh, Viswanathan, & Josi, 2008; Phadke, Uzunalioglu, Mendiratta, et. al. 2013), and some hybrid methods (Huang, Kechadi, 2013; Qi, Zhang, Liu, et. al. 2009). In this paper, we compare the performance of data distortion algorithms incorporating into the following churn prediction methods:

Support Vector Machine (SVM)

SVM is based on structural risk minimization theory (Vapnik, 1998). It has been successfully applied to many applications like face identification, text categorization, and bioinformatics. In SVM classification, the goal is to find a hyperplane that separates the examples with maximum margin. Coussement and Van den Poel have applied SVM in a newspaper subscription context to construct a churn model with a higher predictive performance (Coussement and Van den Poel, 2008).
**k - Nearest Neighbor (k-NN)**

The k-nearest neighbor algorithm (k-NN) was proposed by Fix and Hodges (Fix, Hodges, 1951). It is a method for classifying objects based on closest training examples in the feature space. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors. In (Brandusoiu, Toderean, 2013) three predictive models for subscribers churn in mobile telecommunications companies were built using k-NN, logistic regression, and Bayesian networks.

**Neural Network (NN)**

An artificial neural network (Han, Kamber, and Pei, 2011) is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. A widely used error back propagation algorithm (BP algorithm) was proposed in (Lippmann, 1987). Sharma and Panigrahi (Sharma, Panigrahi, 2011) proposed a neural network (NN) based approach to predict customer churn in subscription of cellular wireless services.

**Logistic Regression**

Logistic regression (Han, Kamber, and Pei, 2011) is used for prediction of the probability of occurrence of an event by fitting data to a logit function. It is usually used as a baseline method in comparing churn prediction algorithms (Brandusoiu, Toderean, 2013; Coussement and Van den Poel, 2008; Huang, Kechadi, Buckley, 2012).

**Perceptron**

The perceptron (Han, Kamber, and Pei, 2011) is a binary classifier which maps its input $x$ to an output value $f(x)$ across the matrix.

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{else} \end{cases},$$

where $w$ is a vector of real-valued weights and $b$ is the “bias”, a constant term that does not depend on any input value. The value of $f(x)$ is used to classify $x$ as either a positive or a negative instance, in the case of a binary classification problem. Perceptron is often used as a baseline method in churn prediction (Huang, Kechadi, Buckley, 2012).

**Decision Tree**

Decision tree (Han, Kamber, and Pei, 2011) is one of the most commonly used tools for predictions and classification. A tree represents a collection of multiple rule sets. Each node in a decision tree is a test condition and the branching is based on the value of the attribute being tested.

Brandusoiu and Toderean compared three decision tree algorithms: Classification and Regression Tree, Chi-squared Automatic Interaction Detection Tree, and Quick Unbiased Efficient Statistical Tree in churn prediction (Brandusoiu, Toderean, 2013). They observed that for predicting both churners and non-churners, all three models have slightly the same performance.
Naïve Bayes

A Bayes classifier (Han, Kamber, and Pei, 2011) is a simple probabilistic classifier based on applying Bayes’ theorem with strong (naïve) independence assumptions. Vergraken et. al. (Verbraken, Verbeke, Baesens, 2014) investigated the predictive power of a number of Bayesian Network algorithms in churn prediction, ranging from the Naïve Bayes classifier to General Bayesian Network classifiers. The results of the experiments were rigorously tested and indicated that most of the analyzed techniques have a comparable performance.

DATA DISTORTION ALGORITHMS

All of the above mentioned methods are applied directly to customers’ data for churn prediction. The private information in the dataset is not protected. We have proposed some data distortion algorithms, which perturb the values of the original data set before it is processed by the churn prediction algorithms. In this section, we describe the data distortion methods we have proposed.

Subdivision Surface (SS) Based Algorithms

A subdivision surface (Peters, Reif, 2008), in the field of 3D computer graphics, is a method of representing a smooth surface via the specification of a coarser piecewise linear polygon mesh. The smooth surface can be calculated from the coarse mesh as the limit of a recursive process of subdividing each polygonal face into smaller faces that better approximate the smooth surface. Many subdivision methods have been proposed in literature.

Subdivision techniques, such as Catmull-Clark subdivision (Cashman, 2012), can be used for data distortion. For any given data set that is represented in a matrix $A$, each entry value $v$ of $A$ at $(i, j)$ can be viewed as a 3D point $(i, j, v)$. After a few steps of subdivision, the newly obtained matrix would be similar to the original matrix $A$, but not the same. The new matrix maintains most of the properties of $A$. Meanwhile, due to the random numbers $\alpha$ and $\beta$ used in the subdivision process, it is impossible to reverse the subdivision process to get the original matrix $A$ even if the subdivision scheme used is known. Therefore, the new matrix is a good substitute for the original matrix to preserve privacy.

SVD Based Algorithms

Singular Value Decomposition (SVD) is a popular matrix factorization method in data mining and information retrieval (Berry, Drmac, and Jesup, 1999). It has been used to reduce the dimensionality of (and remove the noise from the noisy) datasets in practice.

The SVD of the data matrix $A$ is written as

$$A = U\Sigma V^T,$$  \hspace{1cm} (2)

where $U$ is an $n \times n$ orthonormal matrix, $\Sigma = \text{diag}[\sigma_1, \sigma_2, \ldots, \sigma_s]$ ($s = \min(m, n)$) is an $n \times m$ diagonal matrix, and $V^T$ is an $m \times m$ orthonormal matrix. The SVD transformation has the property that the maximum variation among the objects is captured in the first dimension, as $\sigma_1 \geq \sigma_i$ for $i \geq 2$. Similarly, much of the remaining variation is captured in the second
Thus, a transformed matrix with a much lower dimension can be constructed to represent the structure of the original matrix faithfully. Define

$$A_k = U_k \Sigma_k V_k^T,$$

(3)

where $U_k$ contains the first $k$ columns of $U$, $\Sigma_k$ contains the first $k$ nonzero singular values, and $V_k^T$ contains the first $k$ rows of $V^T$. It has been proved that $A_k$ is the best $k$ dimensional approximation of $A$ in the sense of the Frobenius norm. The removed part $E_k = A - A_k$ can be considered as the noise in the original dataset. When used for privacy-preserving purpose, the distorted dataset $A_k$ can provide protection for data privacy, and at the same time, it keeps the utility of the original data as it can faithfully represent the original data structure (Xu, Zhang, et. al., 2006).

**Fast Fourier Transform (FFT) Based Algorithms**

Fourier related transform has been applied in many areas such as physics and image processing. It has some features that make it suitable for the data distortion task (Mukherjee, Chen, & Gangopadhyay, 2006): 1. It can preserve Euclidean distance between data values in the transformed domain. 2. It can be used as a compression technique by suppressing small coefficients and keeping the large ones. 3. The transformed value cannot be reconstructed.

**Geometric Transformation Algorithms**

Geometric transformation in imaging geometry such as translation, scaling, rotation and projection has been applied as data distortion algorithms. The key feature of the geometric transformation methods is preserving the Euclidean distance between any pair of points, thus they can maintain the prediction accuracy of Euclidean distance based classification methods (Chen & Liu, 2005; Ketel & Homaifar, 2005).

**Translation**

Translation is the task to move a point with coordinates $(X, Y)$ to a new location by using displacements $(X_0, Y_0)$. Given a vector $v$, translation can be represented by:

$$v' = Tv,$$

where $T = \begin{bmatrix} 1 & 0 & X_0 \\ 0 & 1 & Y_0 \end{bmatrix}$.

**Scaling**

Given a vector $v$, scaling along X and Y axes can be represented by:

$$v' = Sv,$$

where $S = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$.

**Rotation**

Given a vector $v$, rotation about the coordinate axes by an angle $\theta$ can be represented by:

$$v' = Rv,$$

where $R = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$. 

PERFORMANCE METRICS

In this section, we will compare the performance of the four categories of data distortion methods described in the previous section. We will use several metrics to measure the capability of each method in protecting privacy. To compare the efficiency and performance of the data distortion algorithms, the following metrics are calculated and tested.

Privacy Preservation Metrics

Traditionally, the privacy provided by a perturbation technique is measured as the variance between the actual and the perturbed values (K. Muralidhar, 1999). This measure is given by $\text{Var}(X-Y)$ where $X$ represents a single original attribute and $Y$ the distorted attribute. This measure can be made scaling invariant with respect to the variance of $X$ by $S = \frac{\text{Var}(X-Y)}{\text{Var}(X)}$. The above measure for quantifying privacy is based on how similar the original values and the modified values of an attribute are. We use $PM$ to denote the privacy metric, where $PM$ is the average of the $S$ for all attributes in a dataset.

Data Distortion Metrics

The magnitude of values in an attribute may reveal the information saved in the attribute (e.g. salary, age) and the rank of magnitude in an attribute may identify a customer (e.g. the millionaire in town). Thus data distortion algorithms are expected to change the rank of the magnitude of the data elements. We use the following metrics to measure the change in the distorted datasets.

Rank Changed (RC)

Rank Changed (RC) is used to measure the average change of ranks for all the attributes. For example, if a hacker knows the person with the highest salary is the CEO in the company, then after locating the record with the highest value in the salary attribute, the hacker can easily know other information about the CEO by looking at the other values in the same record. Thus it is desirable that the salary of the CEO is no longer the highest after distortion. A good data distortion algorithm may change the ranks of some values in an attribute. Assume a dataset $A$ has $n$ data objects and $m$ attributes. $\text{Rank}_j^i$ denotes the rank (in ascending order) of the $j$th element in attribute $i$, and $\text{Rank}_j^{A_i}$ denotes the rank (in ascending order) of the distorted element $A_{ji}$. Rank Changed (RC) is defined as:

$$RC = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} |\text{Rank}_j^i - \text{Rank}_j^{A_i}|}{m \times n}$$  

(4)

Rank Kept (RK)

Rank Kept (RK) represents the percentage of elements that keep their ranks of magnitude in each attribute after the distortion. It is computed as:

$$RK = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \text{Rank}_j^i}{m \times n}$$  

(5)
\begin{equation}
R_k^i = \begin{cases} 
1, & \text{if } \text{Rank}_k^i = \text{Rank}_{ik}^i \\
0, & \text{otherwise}
\end{cases}
\end{equation}

Attribute Rank Changed (ARC)

It is possible to infer the content of an attribute from the magnitude of its values compared with the other attributes. For example, if the values of an attribute range from 20 to 75, one can infer that this attribute is about age. Thus it is desirable that the rank of the average value of each attribute changes after data distortion. Here we use the metric Attribute Rank Changed (ARC) to define the change of rank of the average value of all the attributes:

\begin{equation}
ARC = \frac{\sum_{i=1}^{m} |RAV_i - \overline{RAV_i}|}{m},
\end{equation}

where $RAV_i$ is the rank (in ascending order) of the average value of attribute $i$, while $\overline{RAV_i}$ is its rank (in ascending order) after the distortion.

Attribute Rank Kept (ARK)

Attribute Rank Kept (ARK) is defined to measure the percentage of the attributes that keep their ranks of average value after the distortion. So it is calculated as:

\begin{equation}
ARK = \frac{\sum_{i=1}^{m} C_k^i}{m},
\end{equation}

where $ARK^i$ is computed as:

\begin{equation}
ARK^i = \begin{cases} 
1, & \text{if } RAV_i = \overline{RAV_i} \\
0, & \text{otherwise}
\end{cases}
\end{equation}

The value of $RC$ and $ARC$ is proportional to the level of the distortion. On the contrary, the value of $RK$ and $ARK$ is inversely related to the level of distortion. The higher the value of $RC$ and $ARC$, and the lower the value of $RK$ and $ARK$, the more the original data matrix is distorted, which implies the data distortion algorithm is better in preserving privacy.

RESULTS

Experiments have been conducted to test the efficiency of the data distortion algorithms described in this paper. A variety of Churn prediction methods have been applied to the original dataset as well as the perturbed dataset to compare the prediction accuracy on the original dataset and the perturbed dataset. The data distortion algorithms are compared on prediction accuracy, computational time and their ability to protect the privacy of data.

The dataset used in the experiments was downloaded from SGI (SGI, 2011), which originated from UCI (UCI, 2011) but converted to MLC++ format. The data consists of 5000 samples of customers from a telecommunications company. There are 20 predictors which include customer account information, plan information, and usage information. A 5-fold cross validation is performed to obtain the prediction results. A 5-fold cross validation has 5 round. In each round, the dataset is partitioned into 5 groups, 4 of which are used as training data and 1 group
is used as test data. In the next round, the training data and test data rotate until every group has served as test data after 5 rounds.

**Prediction Accuracy**

It is ideal that the distorted data keeps the prediction accuracy of the original data no matter what prediction method is used. However, in reality, the data distorted by a certain algorithm may only work well for specific prediction methods. Several metrics are adopted to analyze prediction accuracy in this section. Some data distortion algorithms such as SVD, FFT, and SS generate different datasets with different input parameters.

**Precision**

Precision is the number of true positives (i.e. the number of items correctly predicted as belonging to the positive class) divided by the total number of items predicted as belonging to the positive class (i.e. the sum of true positives and false positives). It is calculated by:

\[
\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}
\]

In Churn prediction, *Precision* represents the percentage of real churners within all the predicted churners. Translation, Scaling, SVD and FFT perform well on all the prediction methods.

![Figure 1. Comparison of Precision](image)

**Privacy and Distortion Metrics**

Table 1 compares the data distortion algorithms on privacy and distortion metrics. For the privacy metric *PM*, rotation, scaling and SVD obtain the highest values while translation is the lowest. The *PM* values for the FFT based methods and SS method are also low.
Table 1. Comparison of distortion algorithms on privacy and distortion metrics

<table>
<thead>
<tr>
<th>Data</th>
<th>PM</th>
<th>RC</th>
<th>RK</th>
<th>ARC</th>
<th>ARK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation</td>
<td>2.23E-32</td>
<td>0</td>
<td>1</td>
<td>5.6</td>
<td>0.15</td>
</tr>
<tr>
<td>Scaling</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4.45</td>
<td>0.15</td>
</tr>
<tr>
<td>Rotation</td>
<td>1</td>
<td>1601.80</td>
<td>0.00092</td>
<td>5.7</td>
<td>0.05</td>
</tr>
<tr>
<td>SVD1 (k = 15)</td>
<td>0.9999</td>
<td>1548.40</td>
<td>0.05950</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SVD2 (k = 10)</td>
<td>0.9999</td>
<td>1655.80</td>
<td>0.00510</td>
<td>1.7</td>
<td>0.65</td>
</tr>
<tr>
<td>FFT1 (Cutoff = 0.9)</td>
<td>5.88E-10</td>
<td>1666.60</td>
<td>0.00300</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>FFT2 (Cutoff = 0.7)</td>
<td>1.02E-08</td>
<td>1668.50</td>
<td>0.00170</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SS</td>
<td>5.86E-08</td>
<td>1661.80</td>
<td>0.00097</td>
<td>5.3</td>
<td>0.15</td>
</tr>
</tbody>
</table>

For the data distortion metrics, SS and Rotation obtain the best values. Their RC and ARC values are among the highest and RK and ARK values are among the lowest.

**Computational Time**

The computational time to generate distorted data set by different distortion algorithms is compared in Figure 2. The fastest algorithms are geometric transformation algorithms, which are all completed within 0.1 second. The FFT based methods are also fast, which are completed in less than 1 second. With the cutoff rate drops from 0.9 to 0.7, the computational time decreases from 0.73 second to 0.28 second. SS method uses 3 seconds. The slowest are SVD based methods which consume more than 10 seconds. The shrink of rank in SVD contributes only marginally to its speed. When the rank drops from 15 to 10, only 0.2 second is saved in data distortion time.

**CONCLUSION**

In this paper, we compare several data distortion algorithms for a variety of churn prediction methods. There is no distortion algorithm that fits for all the prediction methods and excels in all performance metrics at the same time. Based on the experimental results, if prediction accuracy is the number one concern to the company, Scaling and the SVD based methods are good
choices. Data distorted by Scaling and the SVD based methods are the best to maintain prediction accuracy. Rotation, Scaling and SVD obtain the best values for the privacy metric, while SS and Rotation exceed others in data distortion metric analysis. Thus those methods may be considered if customers’ privacy is most important. Geometric transformation algorithms and FFT based methods are the fastest to generate distorted data when computational time is concerned. If the dataset contains information about millions of users, Geometric transformation algorithms and FFT based methods may be considered to reduce distorted data generating time. Overall speaking, Scaling is an outstanding data distortion algorithm for churn prediction, at least for the dataset used in the experiments. With the specific problem on hand and the security needs, users can choose a distortion method that balances the outcomes of all performance metrics.

The experimental results also show that some distorted data improve the prediction accuracy for several prediction methods. Thus the data distortion algorithms proposed can also be applied as data preprocessing methods before prediction to improve accuracy. For example, when Perceptron or Naïve Bayes methods are used for churn prediction, we can try FFT or geometric based methods to preprocess the dataset to improve prediction accuracy.

REFERENCES

References available upon request.