Churn Prediction with Support Vector Machine

PIMPRAPAI THAINIAM

EMIS8331 Data Mining
Table of Contents

» Customer Churn
» Support Vector Machine
» Research
Customer Churn

What are Customer Churn and Customer Retention?

» Customer churn (customer attrition) refers to when customers (subscribers or users) discontinue their subscription to that service.

» Customer retention is the activity a company undertakes to prevent customers from switching to alternative companies or cancelling the services (churn or attrition).

» Customer churn (customer attrition) is an important issue for mature industries.

» Customer churn is easy to define in subscription-based businesses.
Customer Churn

Subscription-based businesses
» Mobile phone service providers
» Insurance companies
» Cable companies
» Financial services companies
» Internet service providers
» Newspapers
» Magazines
Customer Churn

Why Churn Matters?

» Lost customers must be replaced by new customers.
» New customers are expensive to acquire.
Customer Churn

Different Kinds of Churn

» Voluntary churn : The customer decides to quit his contract himself because of unsatisfaction with the quality of service.

» Involuntary churn (Forced churn) : The company discontinues the contract itself rather than customer.

» Expected churn : The customer decides to quit his contract himself because of some reasons other than unsatisfaction, for example customer’s relocation.
Customer Churn

Different Kinds of Churn Model

» Predicting who will leave (Churn prediction) : This method is trying to predict which customer will leave and which will stay. The outcome for each customer will be binary outcome.

» Predicting how long customers will stay : This kind of churn modeling is survival analysis. The outcome will be hazard probability which is the probability that the customer will leave before tomorrow.
Support Vector Machine

» Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression.

» The original SVM algorithm was invented by Vladimir Vapnik and Alexey Chervonenkis in 1963.
Support Vector Machine

Linearly separable dataset
Support Vector Machine

Which line is the best line for classification?
Support Vector Machine

Y > X
Support Vector Machine

Why is bigger margin better?

$Y > X$
Support Vector Machine

\[ \vec{w} \cdot \vec{u} \geq c \]

*Given* \( c = -b \)*

Decision Rule:

- \( \vec{w} \cdot \vec{u} + b \geq 0 \) *then +*
- \( \vec{w} \cdot \vec{x}_+ + b \geq 1 \)
- \( \vec{w} \cdot \vec{x}_- + b \leq -1 \)

\( y_i \) such that  
- \( y_i = +1 \) *for + sample*
- \( y_i = -1 \) *for - sample*

- \( y_i (\vec{w} \cdot \vec{x}_+ + b) \geq 1 \)
- \( y_i (\vec{w} \cdot \vec{x}_- + b) \geq 1 \)

\( y_i (\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0 \)
Support Vector Machine

\[ \vec{w} = \frac{\vec{w}}{||w||} \]

is a unit vector and perpendicular to a median line.

\[ y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0 \quad \text{for } i \text{ in gutter} \]

\[ \text{Margin} = (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{w}}{||w||} = \frac{(1 - b) + (1 + b)}{||w||} = \frac{2}{||w||} \]
Support Vector Machine
Support Vector Machine

Maximize $\frac{2}{\|w\|}$  →  Maximize $\frac{1}{\|w\|}$  →  Minimize $\|w\|$  →  Minimize $\frac{1}{2}\|w\|^2$

Minimize $\frac{1}{2}\|w\|^2$
subject to
$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$ for $i = 1,2, \ldots, N$

Use Lagragian Method and Quadratic Programming to solve for $\vec{w}$ and $b$. 

$\vec{w} \cdot \vec{x} + b = 0$
Support Vector Machine

Lagrangian Method

\[
L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{N} \alpha_i \left[ y_i (\bar{w} \cdot \bar{x}_i + b) - 1 \right] \quad \text{--------- (1)}
\]

\[
\frac{\partial L}{\partial w} = \bar{w} - \sum_{i=1}^{N} \alpha_i y_i \bar{x}_i = 0 \quad \bar{w} = \sum_{i=1}^{N} \alpha_i y_i \bar{x}_i \quad \text{--------- (2)}
\]

\[
\frac{\partial L}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \quad \text{--------- (3)}
\]

Plug (2) and (3) into (1)

\[
L(\alpha) = \frac{1}{2} \left( \sum_{i=1}^{N} \alpha_i y_i \bar{x}_i \right) \cdot \left( \sum_{j=1}^{N} \alpha_j y_j \bar{x}_j \right) - \left( \sum_{i=1}^{N} \alpha_i y_i \bar{x}_i \right) \cdot \left( \sum_{j=1}^{N} \alpha_j y_j \bar{x}_j \right) - \sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i
\]

\[
L(\alpha) = -\frac{1}{2} \left( \sum_{i=1}^{N} \alpha_i y_i \bar{x}_i \right) \cdot \left( \sum_{j=1}^{N} \alpha_j y_j \bar{x}_j \right) + \sum_{i=1}^{N} \alpha_i
\]

\[
L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j
\]

\[\text{SMU.} \quad \text{--------- (1)}\]
Support Vector Machine

Linearly unseparable dataset

How can we separate this dataset?
Support Vector Machine

Linearly unseparable dataset
Support Vector Machine

Linearly unseparable dataset
Support Vector Machine

Linearly unseparable dataset \( 3D \text{ SVM} \)

Transformation function: \( \varphi(x, y) = xy(x^2 + y^2) \)

This is not kernel function.

(https://www.youtube.com/watch?v=3liCbRZPrZA)
Support Vector Machine

The Kernel Trick

The kernel function is the function that provides us the dot product of the 2 vectors in the space without visiting the space.

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \tilde{x}_i \cdot \tilde{x}_j$$

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \Phi(\tilde{x}_i) \cdot \Phi(\tilde{x}_j) \quad \text{(z-dimensional space)}$$

Kernel function: $K(\tilde{x}_i, \tilde{x}_j) = \Phi(\tilde{x}_i) \cdot \Phi(\tilde{x}_j)$
Commonly used kernel functions

Linear Kernel \[ k(x, y) = x^T y + c \quad c : \text{optional constant} \]

Polynomial Kernel \[ k(x, y) = (\alpha x^T y + c)^d \]
- \( \alpha : \text{slope} \), \( c : \text{constant} \), \( d : \text{polynomial degree} \)

Exponential Kernel \[ k(x, y) = \exp \left( - \frac{\|x - y\|^2}{2\sigma^2} \right) \]

Laplacian Kernel \[ k(x, y) = \exp \left( - \frac{\|x - y\|}{\sigma} \right) \]

Sigmoid Kernel \[ k(x, y) = \tanh(\alpha x^T y + c) \]
Churn Prediction with Support Vector Machine

Model of Customer Churn Prediction on Support Vector Machine

(by Xia, Guo-en, and Wei-dong Jin, 2008)

» Churn prediction for mobile telecommunication company.

» The average churn rate in mobile telecommunication is 2.2% per month.

» The acquisition cost of a new customer is about $300 - $600.

» The acquisition cost is about 5-6 times of retention cost of an existing customer.
The 2 datasets used are (1) Mobile telecommunication dataset (2) Home telecommunication carry dataset.

The researchers used Radial basis kernel function with $u = 0.12$ for dataset (1) and $u = 1$ for dataset (2).

Radial Basis Kernel: $k(x, y) = \exp(-u\|x - y\|^2)$

The accuracy rate, hit rate, coverage rate, and lift coefficient from SVM were compared with the artificial neural network, decision tree, logistic regression, and naïve bayesian classifier.
Support Vector Machine

<table>
<thead>
<tr>
<th>Customer state</th>
<th>Prediction churn</th>
<th>Prediction non-churn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual churn</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Actual non-churn</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

Accuracy Rate = \( \frac{A + D}{A + B + C + D} \)

Hit Rate = \( \frac{A}{A + C} \)

Coverage Rate = \( \frac{A}{A + B} \)

Lift Coefficient = \( \frac{\text{Hit Rate}}{\text{Churn Rate of the test set}} \)
Churn Prediction with Support Vector Machine

Prediction results from dataset (1)

<table>
<thead>
<tr>
<th>Model type</th>
<th>Accuracy rate</th>
<th>Hit rate</th>
<th>Coverage rate</th>
<th>Lift coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.9088</td>
<td>0.8333</td>
<td>0.4018</td>
<td>6.2186</td>
</tr>
<tr>
<td>ANN</td>
<td>0.8983</td>
<td>0.7538</td>
<td>0.3625</td>
<td>5.6256</td>
</tr>
<tr>
<td>Decision tree C4.5</td>
<td>0.8386</td>
<td>0.3869</td>
<td>0.3437</td>
<td>2.8876</td>
</tr>
<tr>
<td>Logistic regression</td>
<td>0.8716</td>
<td>0.6190</td>
<td>0.1160</td>
<td>4.6198</td>
</tr>
<tr>
<td>Naive bayesian classifiers</td>
<td>0.8782</td>
<td>0.7142</td>
<td>0.1562</td>
<td>5.3305</td>
</tr>
</tbody>
</table>

Prediction results from dataset (2)

<table>
<thead>
<tr>
<th>Model type</th>
<th>Accuracy rate</th>
<th>Hit rate</th>
<th>Coverage rate</th>
<th>Lift coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.5963</td>
<td>0.7141</td>
<td>0.1620</td>
<td>1.5975</td>
</tr>
<tr>
<td>ANN</td>
<td>0.5569</td>
<td>0.7500</td>
<td>0.0139</td>
<td>1.6779</td>
</tr>
<tr>
<td>Decision tree C4.5</td>
<td>0.5248</td>
<td>0.4657</td>
<td>0.4236</td>
<td>1.0417</td>
</tr>
<tr>
<td>Logistic regression</td>
<td>0.5890</td>
<td>0.7012</td>
<td>0.1412</td>
<td>1.5686</td>
</tr>
<tr>
<td>Naive bayesian classifiers</td>
<td>0.5549</td>
<td>0.6250</td>
<td>0.0116</td>
<td>1.3982</td>
</tr>
</tbody>
</table>
THANK YOU

Q & A