CS412 Project Report

Team members: Jeannelle Alford (jkalfor2), Joe Fetsch (jfetsch2), Aaron Park (aaronyp2)
Kaggle group: “Whatever”

Introduction

A central task of an insurance company is to predict the risk levels of its customers, both current and new, to properly price their policies. A customer who is more likely to get sick or get in an accident, and thus file a claim later, would pay a higher premium appropriate to their risk level. The goal of this project is to identify those life insurance clients that have high levels of risk by making predictions based on attribute variables, such as age, height, and BMI. The predictions were made by training a classifier with a past dataset that included the "response variable", which is categorized into 8 levels of risk, in addition to the same attributes included in the test set. The predictions are crucial to the company's profits, especially for determining the optimal price of the policies that is not too high to drive away new clients but also not too low to avoid selling cheap policies to high-risk customers and operating at a loss.

However, there are many challenges that companies face when making these predictions. Some of these challenges lie in the dataset itself, such as missing, noisy or inconsistent values and outliers. A particular challenge of the given dataset is the high number of variables, numbering over one hundred, that brings problems inherent in high-dimensional spaces. This challenge, commonly known as the "curse of dimensionality", causes difficulty in analyzing the dataset. The high number of combinations of values in different dimensions means that a huge amount of training data is required to make sure there are enough samples that correspond to the different combinations. A solution to these challenges is then to apply data preprocessing techniques, such as filling in the missing values, which is included in the current implementation, and reducing the dimensionality through PCA or heuristic identification of redundant or irrelevant attributes.

Our goal is to employ a classification algorithm to predict the risk level of insurance customers as accurately as possible. The data in training.csv is used to train a classification model, which is then used to make predictions about the data in testing.csv. By comparing different classification strategies, we can determine what algorithm or combination of algorithms works the best on the insurance data. So far, we have tried using a naive Bayes classification approach on the data.

Related Work

A paper by Jurek and Zakrzewska at the Technical University of Lodz uses naive Bayes to predict insurance risk. The interesting method employed by the researchers was to first perform cluster analysis on the data and then train separate Bayesian models for each of the clusters. This contrasts with our approach, which only learns one probability model.

The results were improved using the clustering technique, according to the authors. Another paper by Yao explores application of clustering algorithms to insurance pricing, using methods such as k-means clustering and partitioning, in order to group data objects into clusters and/or reduce the number of levels in risk factors to make the feature of each cluster more distinguishable. For our project, it may be more worthwhile to consider other baseline classification models, such as decision tree induction and rule-based classification.

A paper by Devale and Kulkarni also suggests clustering as a means of identifying target groups of potential new clients that are uninsured based on a similarity measure between predefined attributes. However, this type of clustering may not be appropriate to the given dataset, as it is in a high-dimensional space that contains many binary variables, for which k-means is not useful, and would also add a lot of complexity to implementation. A possible workaround is to select a few attributes for which similarity distance would be calculated.

The paper also suggests K-Nearest Neighbors as a classification method of mapping data, such as age, income, and occupation, into predefined groups. A strength of the algorithm would be that, given an enough representative training dataset, the algorithm would perform well. However, it is computationally expensive to find the *k* nearest neighbors when the dataset is large, as the distance of each data point to the training samples has to be computed. The curse of dimensionality also applies, so attribute selection is required. Since the algorithm is distance-based, the distance metric must be meaningful, especially for nominal variables. For nominal variables, a simple binary distance function that returns 1 if two values are identical and 0 otherwise may be used. A good value for *k*, the number of neighbors, needs to be determined experimentally, further complicating the classification process. Overall, the algorithm may perform well given the right training dataset and parameters but can be extremely slow, though classification time can be sped up by techniques such as partial distance (using only a subset of the attributes) and editing (removing data points in the training dataset that turn out to be useless in the classification process).

Implementation

Our algorithm uses a naive Bayes approach to the problem. Bayesian classification creates a statistical model which attempts to predict the class of a tuple based on learned probabilities. Every attribute value affects the probability of the tuple belonging to a certain class. Naive Bayes is naive in that it assumes the effect each attribute has upon the outcome is independent of the other attributes, an assumption that does not hold true in most real situations.

Bayesian classification tries to predict *P(H|X),* the probability of *H* given *X*, where *H* is a hypothesized value and *X* represents the known information. Probabilities are calculated based on the training data and are used to make predictions about new data. Bayes' Theorem states that

$$P(H|X) = (P(X|H) P(H)) / P(X)$$

Our classifier uses these principles to predict insurance risk based on probabilities in the training data, using every tuple in the training data as evidence. To account for missing values in the training data, we filled in the values with average data, depending on the type of attribute. Nominal attributes were filled in with the mode, ordinal and continuous attributes were filled in with the mean. For ordinal attributes, the mean was rounded.

Result Interpretation

The predictions made by our Bayes algorithm are very poor, achieving a quadratic weighted kappa score of only 0.09624. This suggests that naive Bayes is not an appropriate method of classification for this dataset. With insurance data, many of the attribute values will be correlated with each other, especially health-related attributes. BMI and family medical history, for example, will have a significant correlation with the customer's medical history. A weakness of our implementation is the fact that it naively assumes all attributes are independent. The naive Bayes algorithm was the only model that was implemented, so there is no other baseline model to which our classifier can be compared, but implementation of another classification method is warranted for improvement.

References

A. Jurek and D. Zakrzewska, "Improving Naïve Bayes models of insurance risk by unsupervised classification," *2008 International Multiconference on Computer Science and Information Technology*, Wisia, 2008, pp. 137-144.

J. Han. *Data mining : concepts and techniques*, 3rd ed. Morgan Kauffman, 2013, pp. 350-355.

J. Yao. "Clustering in Ratemaking: Applications in Territories Clustering," *2008 Discussion Paper Program*, Casualty Actuarial Society*,* 2008, pp. 170-192.

A. B. Devale. And R. V. Kulkarni, "Applications of Data Mining Techniques in Life Insurance," *International Journal of Data Mining & Knowledge Management Process Vol. 2 No. 4*, 2012, pp. 31-40.