Faculty of Engineering and Information Technology
University of Technology Sydney

Learning and satisfying customer needs for financial adviser

A thesis submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

by

Charles Yu-Chia Chu

November 2020

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Charles Yu-Chia Chu, declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature of Candidate: Production Note: Signature removed prior to publication.

Date: 7/11/2020

Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Guandong Xu for the continuous support of my Ph.D study and research, for his immense knowledge, patience, and motivation. His guidance helped me during the whole time of research and writing of this thesis. I could not have imagined having a better supervisor for my Ph.D study.

I also would like to appreciate my co-supervisor Dr. Shaowu Liu for providing me with continuous support throughout my Ph.D study and research. Without his professional guidance and persistent help, this thesis would not have been possible.

I thank my fellow labmates in Advanced Analytics Institute and colleagues in Colonial First State: James Brownlow, Ben Culbert, Bin Fu, and Todd Stevenson for the inspiring discussions, and for all the fun we have had in these years.

Last but not the least, I would like to thank my family: my wife , my daughters and my parents, for their unconditional support, both financially and emotionally throughout the whole PhD studying.

Charles Chu
Nov 2020 @ UTS

Contents

List of Figures

List of Tables

List of Publications

Papers published/under review

- Chu, C., Brownlow, J., and Meng, Q., et al. Combining heterogeneous features for time series prediction. In proceeding of the 2017 International Conference on Behavioral, Economic, Socio-cultural Computing (BESC). 2017.

- Chu, C., Xu, G., and Brownlow, J. et al. Deployment of churn prediction model in financial services industry. In proceedings of the 2016 International Conference on Behavioral, Economic and Socio-cultural Computing (BESC). 2016.

- Chu, C., Xu, G., and Brownlow, J.. Inferring customer's financial needs by multi-label learning. expert Systems with Applications. (Under review)

- Chu, C., Xu, G., and Brownlow, J.. Predicting changing of adviser by bipartite ranking. (Under review)

- Chu, C., Xu, G., and Brownlow, J.. Adviser recommendation based on heterogeneous graph. (To be submitted)

- Brownlow, J., Chu, C., and Xu, G., et al. A multiple source based transfer learning framework for marketing campaigns. In proceedings of the 2018 International Conference on Neural Networks (IJCNN). 2018.

- Brownlow, J., Chu, C., and Fu, B. et al. Cost-sensitive churn prediction in fund management services. In proceedings of the 23rd International Conference on Database Systems for Advanced Applications. 2018.
- Vo, N.N., Liu, S., and Chu, C. et al. Client churn prediction with call log analysis. In: proceedings of the 23rd International Conference on Database Systems for Advanced Applications. 2018.
- Culbert, B., Fu, B., and Chu, C. et al. Customer churn prediction in superannuation: A sequential pattern mining approach. In: proceedings of the 2018 Australasian Database Conference. 2018.

Research Reports of Industry Projects

- Discovering deep insights into SG contribution. Colonial First State, Dec 2016.
- Discovering deep insights into customer retention. Colonial First State, Dec 2015.
- Reshaping superannuation practice in Australia using big data analytics. ARC-Linkage Project, 2018-2021.

Abstract

Nowadays, people tend to seek help from professional financial advisers to manage investment and prepare for retirement. People in different financial situations usually have different financial objectives and needs when choosing the advisers they want to see. Therefore, understanding and satisfying customer needs are critical to financial planning services and financial businesses, due to the huge impact on people's financial wellness and retirement readiness.

Traditionally, several tools such as questionnaires, surveys, and statistical analysis, etc., have been employed in practice to manually collect feedback and reviews from a group of customers to understand their needs. However, with a growing number of customers and the emergence of big data, those tools are prohibitively time-consuming or even infeasible, especially when people are overwhelmed with complex customer data. To address this big data challenge, data mining techniques that could learn underlying patterns from big data effectively have been investigated and applied extensively in various fields. Inspired by that, this thesis focus on applying existing or designing new data mining methods to address challenges in inferring and satisfying customers' needs and expectations for financial advisers. Specifically, the main contributions of this thesis are listed as follows.

(1) An automatic end-to-end framework that follows the typical data mining process is designed and implemented to learn the multiple needs of every individual customer.at scale. To begin with, three possible needs are defined using domain knowledge and extracted from data for a selective group

of customers. Based on the labeled dataset, multi-label learning is then applied to build predictive models that could predictive multiple needs of other customers. The advantage of this framework is that it could exploit heterogeneous data sources and predict the personalized needs of individual customers without involving any manual work. Experimental results also verify its effectiveness in learning customer needs.

(2) A novel learning method is proposed for early detection of customers who are likely to stop engaging their current advisers, i.e., the customer churn prediction problem. Specifically, this problem is dealt with as a bipartite ranking problem, and a model is trained which can rank possible churners before those who are less likely to churners. Furthermore, to address the issue of extremely imbalanced data, i.e., there are few churners during a particular time, an instance-based transfer learning strategy is adopted to take advantage of auxiliary data that might of different distribution. In this way, only weights of those data that could improve model performance are increased iteratively, so they could be fully exploited to alleviate the issue of imbalanced data. Furthermore, a novel ranking-based measure is incorporated into the learning process to guide the process towards learning good rankings. Experimental results validate our method's effectiveness in improving model performance by utilizing only useful auxiliary data.

(3) Research on recommending financial advisers to customers is also investigated. To our knowledge, they are little research on this topic. To cope with the issue of a lack of explicit customers' preferences over advisers, a graph-based method is applied to organize customers and advisers in a heterogeneous network. Specifically, their connections are determined by similarities between them in terms of demographic and behavioral features. Furthermore, a random walk with restart process is run to identify advisers who are more preferable for a particular customer.

The effectiveness of these proposed methods has been validated through extensive experiments. In doing so, this research advances the understanding of customer needs for financial advisers, thus provide financial businesses better chances to satisfy and retain their customers. In summary, this thesis has proposed several effective methods that learn and satisfy customer needs for financial advisers from different perspectives, and their effectiveness has been validated by experiments. These achievements lay a good foundation for further research and applications.

Chapter 1

Introduction

1.1 Background

In today's complex business world, understanding and satisfying customer needs are at the core of customer relationship management for businesses. customer needs generally refer to the motivations people have when they buy or keep particular products and services. Understanding customer needs is beneficial to both businesses and customers. For businesses, understanding their customers' needs could help deliver more desirable products and services, to improve customer loyalty and retain them. For customers, their expectations could be better fulfilled if their needs could be known to businesses. Therefore, there have been various techniques and tools developed to analyze customer needs in past decades (Edvardsson et al. 2012; Wang et al. 2010; Rashid 2010; Xu et al. 2009; Wang and Tseng 2013).

1.1.1 Needs for financial advisers

Compared with other sectors, understanding customers' needs for financial adviser is way more important in the financial sector. Financial advisers are professionals that provide advice, assistance, plan and monitoring to help people in fund investment and retirement preparation. Figure 1.1 shows the relationship between customers, financial advisers, and financial services. It

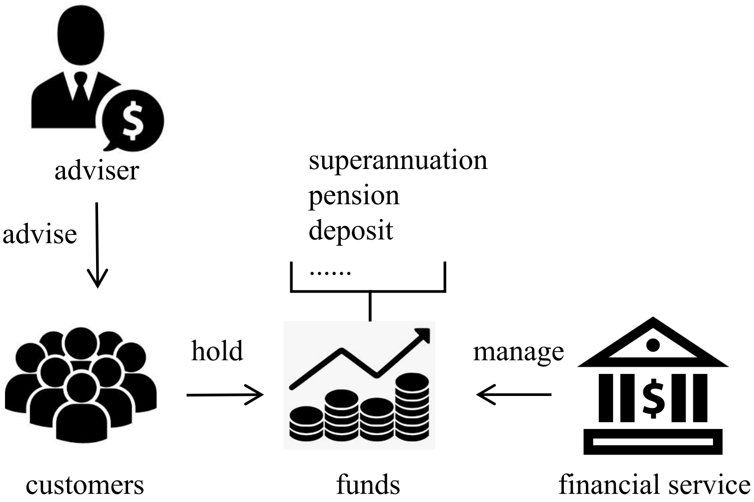


Figure 1.1: Customer, financial adviser, and financial service

can be seen from Figure 1.1 that financial services provide a list of investment options like superannuation, pension, and deposit, etc. for customers to choose, and financial advisers provide investment advice to customers using their professional knowledge. Financial advisers thus play an important role in help customers managing their investments, especially for those with a low level of financial literacy.

A lot of research has been conducted to examine the value of seeing financial advisers. Clare et al. 2017 categorized financial advice into three types according to the degree that is it tailored to each person, i.e., general financial product advice, scaled personal advice, and comprehensive advice. It has been proven that financial advice is related to positive financial activities including retirement preparation, saving behavior, and risk asset holdings, etc. (Marsden et al. 2011; Foerster et al. 2014; Kim et al. 2018). Study in (Lei and Yao 2016) also reveals that households that use financial advisers demonstrate better portfolio performance than those who do not. Furthermore, studies on factors cause people to seek financial advice has been a growing topic, for example, the correlation between financial literacy and financial advice demand (Calcagno and Monticone 2015)

Although financial advice has proved to be valuable generally, different people should have different needs when they are seeking financial advice, because of their own financial status, objectives, and risk tolerance. Therefore, it is reasonable for financial services and financial advisers to understand their needs to provide suitable investment options, advice, and plans. Moreover, it is more significant to understand customers' needs for financial advice, since a huge amount of money might be involved which has a huge impact on people's financial wellness and retirement preparation. Accordingly, effective methods for inferring and satisfying customers' needs for financial advisers is also vital to financial businesses.

1.1.2 Data mining techniques

Traditionally, a set of tools such as questionnaire and survey have been adopted to collect customer feedback and extract their needs. These traditional methods involve a lot of manual work and are not applicable to the rapidly growing amount of data. That is why data mining has become popular as a new technique to learn from customers. Data mining is the subject of discovering novel and potentially useful information from large amounts of data (Han et al. 2012). It has proved to be useful in understanding underlying patterns and insight into customer behavior, and have been extensively used in many sectors including including financial services (Zdanowicz 2004; Brownlow et al. 2018a), health care (Koh et al. 2011; de Oliveira et al. 2010), telecommunication (Huang et al. 2015), E-commerce (Jiang and Yu 2008), and bioinformatics (Jiang and McQuay 2012; Naulaerts et al. 2015) etc. Motivated by its extensive application and advantages in learning from big data, this thesis focus on applying data mining techniques to address several critical issues in inferring and satisfying customer needs for financial advisers which are outlined in the following sections.

1.2 Research issues

Customer acquisition and customer retention are the two essential parts to keep growing business and improving customer loyalty. To better fulfill these two parts, the following research issues are investigated in this thesis.

The first one is understanding every individual customer's multiple needs. The challenges in addressing this issue include: 1) intensive manual work and domain experts are required in traditional tools like survey and feedback (Edvardsson et al. 2012), making them infeasible to infer needs for a large number of customers at scale. 2) It is more difficult to identify people's needs for financial advisers because there are few data that could show customers' explicit needs or their interaction with advisers. Moreover, many customers might not be conscious of their financial needs due to a low level of financial literacy (Worthington 2013). Therefore, it is necessary to adopt suitable data mining techniques to extract customer needs from available data.

The second issue is the early detection of customers who are likely to stop engaging their current advisers. Different from the first issue that is related to customer acquisition, this issue is related to customer retention and is normally called customer churn prediction problem. To deal with this problem, many existing classification methods have been adopted to address it as a binary classification task Huang et al. (2015); Lu et al. (2014). However, it is more suitable to regard it as a bipartite ranking problem that aims to rank churners before non-churners, so businesses can focus their marketing resources on those at the top of a ranking. Moreover, imbalanced data is a key challenge since there are few churners during a period of time. While a lot of historical data are available, they cannot be directly used to help to learn from current data, since they might be of different sources. For example, customers in the past should have different reasons to become churners.

The third issue is adviser recommendation. Recommending the right financial advisers to customers is probably the most effective way to satisfy customer needs. Although recommendation techniques have been widely used in other industries (Ekstrand et al. 2018; Grbovic and Cheng 2018), we have not seen any research on financial adviser recommendation. Therefore, the adviser recommendation is investigated in this thesis. The key challenges here is the lack of data about customers' preference like the rating matrix in movie recommendation (Deshpande and Karypis 2004). Moreover, unlike people who could interact with multiple items in other applications, a customer has only one single adviser in most cases, leading to inferring their preferences more difficult.

1.3 Research contributions

To deal with the above issues, the following contributions are made in this thesis accordingly.

(1) An end-to-end automatic framework that treats customer needs identification as a data mining application is proposed. Specifically, features and predefined needs using domain knowledge are extracted for a selected group of customers. Based on this labeled dataset, a multi-label learning model is built which predicts a customer's multiple needs simultaneously.

(2) A novel method is proposed for the early detection of possible churners. To deal with the imbalanced data, an instance-based transfer learning strategy is adopted to identify those historical data that might useful for the current learning task via an iterative re-weighting mechanism. A ranking-based evaluation measure is also designed and used in the learning process, so detection of churners is solved as a bipartite ranking task which tries to rank possible churners before non-churners.

(3) A graph-based method is proposed for adviser recommendation. To deal with the cold-start challenge, the heterogeneous network of users and advisers is augmented by connecting similar users/advisers in terms of their profiles and behaviors. The random walk with restart process is then applied to exploit the network structure to find similar advisers to a given customer.

1.4 Structure of the thesis

This remainder of this thesis is structured as follows.

Chapter 2 reviews related foundations and current progress, including classification, multi-label learning, imbalanced data, and recommendation system.

Chapter 3 corresponds to the first contribution. A framework is proposed to infer customer needs for financial advisers, in which a multi-label learning model is built to predict a customer's multiple possible needs.

In chapter 4, the method for early detection of customers who are likely to close their relationship with their advisers is proposed. The problem is solved as a bipartite ranking task, and data of different sources are exploited to alleviate the imbalanced data issue.

In chapter 5, the issue of adviser recommendation is investigated. A graph-based method is proposed in which users and advisers are organized into a heterogeneous network, and their closeness could be captured by a random walk with restart process.

In chapter 6, the main contributions and conclusions are summarized again. The following research directions are also pointed out.

Chapter 2

Preliminaries and literature review

This chapter reviews and analyses the background, foundation, and recent progress related to the research conducted in this thesis. To begin with, the definition and traditional methods of customer needs identification are introduced. Next, relevant fundamentals, research challenges, and state-of-the-art techniques in data mining are summarized. Finally, the representative work of recommendation systems, which related to our research in Chapter 5, are presented.

2.1 Customer needs analysis

Existing methods of identifying customer needs generally differ in data sources explored to gather customer data and the analysis techniques used to identify customer needs. Traditional methods such as surveys, questionnaires, and interviews, etc., involve professionals to manually collect data and identify possible customer needs (Edvardsson et al. 2012). Despite being effective, the sample they collect may not be sufficient in some cases to obtain a complete understanding of general customer needs. Based on data from surveys or designed experiments, quantitative models are furthermore used to identify customer needs analytically. For example, variations of Kano's model are adapted to examine the relationship between customer satisfaction and the fulfillment of customer requirements (Wang et al. 2010; Rashid 2010; Xu et al. 2009). A Bayes factor-based sequential analysis process is implemented to capture emerging customer needs (Wang and Tseng 2013). Besides analyzing needs of customers that have been involved, machine learning models, e.g., decision tree, k-NN, and random forest, are also exploited to enable the prediction of needs of any unknown customers (Eckstein et al. 2016; Kühl et al. 2019; Kuehl et al. 2016). In addition to traditional data sources, electronic and user-generated data are being increasingly exploited nowadays. For instance, online reviews are analyzed to extract customer opinions directly or transformed using deep learning techniques to assist experts in identifying customer needs more easily (Zhu et al. 2011; Timoshenko and Hauser 2019). Twitter data are investigated to discover informative tweets that contain people's needs (Kühl et al. 2019; Kuehl et al. 2016). A web interface is used to ask customers to enter their needs directly (Schaffhausen and Kowalewski 2015).

2.2 The role of financial adviser

A large amount of research has already been done on the use of financial advisers, and these studies can be roughly divided into the following two categories.

2.2.1 Value of financial advisers

Firstly, a lot of work centre on the value of using financial advice. For example, financial advice proves to be related to positive financial activities including retirement preparation, saving behavior, and risk asset holdings, etc. (Marsden et al. 2011; Foerster et al. 2014; Kim et al. 2018). The study in (Lei and Yao 2016) reveals that households that use financial advisers demonstrate better portfolio performance than those who do not. Financial advice also helps with tax management and reduces tax liabilities as shown in (Cici et al. 2017).

Furthermore, (Finke 2013) provided evidence to demonstrate how financial advisers improve financial outcomes when the interests of the adviser and investors are aligned. Cummings (2013) found clear evidence that individuals with financial advisers are significantly and positively correlated to subsequent net wealth. French (2008) argues that clients can obtain passive management strategies that improve portfolio return.

2.2.2 Incentives to see financial advisers

Many studies have been done around factors that influence people to seek financial advice (Zeka et al. 2016). For example, a logistic regression model is employed in (Joo and Grable 2001) to learn the relationship between people's various factors, e.g., gender and income, and their final decision of seeking financial adviser. The Study in (Calcagno and Monticone 2015) investigates the effect of financial literacy on the demand for financial advice and finds that knowledgeable investors are more likely to consult advisers. The study in (Rickwood et al. 2017) reveals that self-efficacy and attitudes are the main factors to use financial advice. Additionally, there is also research on advisers' preferences of investors. For example, the model in (West 2012) shows that advisers are motivated to target more valuable customers.

2.3 Data mining and classification

With the emergence of big data techniques, people have been overwhelmed by massive data that is increasingly challenging for people to consume and gain desirable information effectively. Data mining, an interdisciplinary subject combining computer science, statistics, and database system, etc., is thus developed to address this challenge. Also known as *knowledge mining* and *data analysis*, data mining formally refers to the process of discovering interesting patterns and knowledge from large amounts of data (Han et al.

2012). In practice, data mining techniques could be applied to learning from various sources of data such as text (Liu 2013; Schapire and Singer 2000), image (Sun et al. 2019; Cabral et al. 2015), and video (Qi et al. 2007; Wang et al. 2011; Connolly et al. 2012). Nowadays, they have been extensively employed in production in fields including financial services (Zdanowicz 2004; Brownlow et al. 2018a), heath care (Koh et al. 2011; de Oliveira et al. 2010), telecommunication (Huang et al. 2015), E-commerce (Jiang and Yu 2008), and bioinformatics (Jiang and McQuay 2012; Naulaerts et al. 2015) etc.

2.3.1 Process of data mining implementation

A successful application of data mining in production involves multiple steps that are critical to the final performance, which are outlined in Figure 2.1.

Data collection

Data collection is the process of gathering data related to the data mining task from multiple sources. It is the first and foremost step because it determines the quality of data fed into a model and thus the upper limit of model performance. Usually, a centralized, well-structured, and homogeneous dataset is not available. By contrast, data, in reality, are mostly: 1) of different forms like text, image, and video, etc., 2) residing in heterogeneous sources, e.g., database, file system, and log records, etc., and 3) noisy and inconsistent because of system errors or different system measures. To address these issues, the ETL (extract, transform, load) tools have been implemented in model data warehouse systems. Moreover, several distributed systems have also been invented particularly to process and store big data, including BigTable (Chang et al. 2008), MapReduce (Dean and Ghemawat 2008), and spark (Zaharia et al. 2010) and so forth. Other issues like data discrepancy and data linkage should be tackled as well in the phase of data collection. The former means that data from different sources might be contradictory to each other, while the latter refers to the process of consolidating information about the same entity to get a richer view.

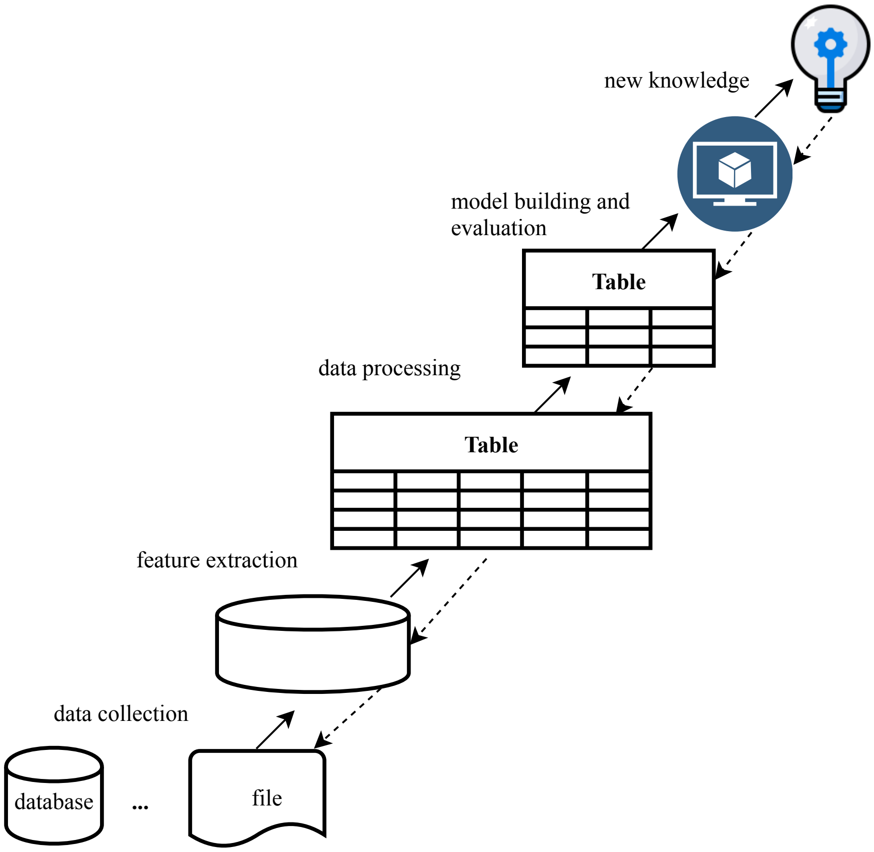


Figure 2.1: The process of data mining implementation

Feature extraction and learning

A feature represents a characteristic of a data object. For example, feature $age = 60$ is a specific measurement for someone with respect to age. Feature extraction refers to extract a set of features from original data, and it is vital because a set of discriminative features enables the underlying patterns to be discovered more easily. Different types of data usually have their unique ways of generating features accordingly. For example, methods such as bag-of-words (Liu 2013), LSA (Latent Semantic Analysis) (Hofmann 2001), and LDA (Latent Dirichlet Allocation) model (Blei et al. 2003) are used to extract features from *text* data, while various transformations are designed for *image* data (Nixon and Aguado 2019). Feature interaction, which is a combination of low-level features, is also extracted to obtain more expressive features using a boosted decision tree or deep neural network (He et al. 2014; Cheng et al. 2016; Guo et al. 2017; Lian et al. 2018). Finally, representation learning is becoming an increasingly hot topic, motivated by learning features automatically without human effort (Bengio et al. 2013).

Data pre-processing

Data pre-processing is performed to improve data quality or fit into the model adopted. Below is a list of commonly used strategies for this purpose.

- Data cleaning. Original data are of poor quality mostly because of missing values, errors, redundancy, and abnormal/noisy values, etc. Thereby, steps such as eliminating outliers, correcting errors, redundancy detection, and filling in missing values should be performed.

- Data transformation. It is the process of transforming data from one format into another format, and there are several typical strategies. 1) Feature discretization that converts numerical features into categorical features. Classical methods such as ChiMerge and Chi2 are summarized in (Garcia et al. 2012). 2) Feature normalization/standardization that rescales the range of value a feature could take, e.g., $[0, 1]$. The purpose is to prevent the learning process from being dominated by some features simply because of the different value scales or making the features more discriminative (Xiao and Ye 2010).

- Data reduction. Data reduction could be performed in terms of both features and data objects. 1) Feature selection selects a subset of features that are most relevant to the learning task. It could be realized by measuring the correlation between features and targets like Pearson coefficient and chi-square, or determined automatically in models like randomForest (Guyon and Elisseeff 2003; Breiman 2001). 2) Feature reduction maps data into a low-dimensional space, classical methods include Principal component analysis (PCA), Linear Discriminant

Analysis (LDA), and factor analysis (Bishop 2006). 3) Instance selection is another pivotal part of data mining by identifying representative examples from massive data. Various sampling techniques like stratified and adaptive sampling are summarized in (Liu and Motoda 2013). Supervised learning strategies like active learning are also well-studied (Fu et al. 2013).

Model building and evaluation

Model building is the core of data mining and the actual step that undertakes the task of learning knowledge from data. Based on the output types, data mining tasks could be categorized into classification, regression, cluster, association rules mining, and recommendation, etc (Witten et al. 2011; Adomavicius and Tuzhilin 2005). Over decades, a diversity of theories and methods for learning models have been proposed, such as linear method, decision tree, and diverse variants of deep neural networks (Goodfellow et al. 2016). Generally, several major factors should be taken into consideration, to choose suitable models for a particular task. The first one is understanding your data since every method has its underlying assumption about the data. For example, linear methods assume that the data should be linearly separable. Thus the assumption embedded within the method chosen should fit the data. The second one is model interpretability which is the extent to which people can understand the model's result. It is important especially in the decision-making process and takes precedence over model accuracy in some cases. This is why classical methods, like linear regression and decision tree, are still commonly used, even though lots of more advanced methods have been proposed. Moreover, interpretable machine learning that aims to design interpretable models also emerges as a hot topic recently (Du et al. 2019; Molnar 2020). The third one is an appropriate level of model complexity to avoid under-fitting and over-fitting problems. Tools such as bias-variance tradeoff and MDL (Minimum Description Length) have been formulated to assist model complexity analysis and selection (Duda et al. 2012).

In reality, we would build and evaluate multiple models (Domingos 2012). The keys here are the process and criteria for model evaluation. Specifically, holding a separate test data and k-folds cross validation are the two common strategies to evaluate model, and different learning tasks have corresponding evaluation criteria. For example, accuracy, Area Under Curve (AUC), and F-measure are typical criteria for classification (Sokolova and Lapalme 2009).

Knowledge discovery

The final step is to discover new knowledge by applying models. As stated above, there are different types of learning tasks that correspond to discover different types of knowledge. For example, classification, such as spam detection and customer churn prediction, predicts the categories to which an instance belongs. Pattern mining identifies a set of frequent patterns underlying data, such as market basket analysis that is employed to identify goods that often bought together (Han et al. 2007). Recommendation infers customers' preferences and recommends them a list of products, e.g. movies and hotels (Covington et al. 2016; Grbovic and Cheng 2018).

Above all the main steps involved in practical data mining applications. It should be noted that the whole process is implemented iteratively as shown in Figure 2.1, so you may go back and forth to achieve a better result.

2.3.2 Supervised learning and classification

Data mining can also be divided into supervised, unsupervised, and semi-supervised learning, etc (Zhou 2017). Since the methods proposed in this thesis fall within the scope of supervised learning, its basic definitions and methods are therefore discussed further.

Supervised learning is the type of machine learning that has ground truth provided. It consists of two steps, i.e., learning a model using training data and making predictions on test data. To be specific, each instance in the training data is a pair of a feature vector and its true output, i.e., ground truth. In this way, learning a model is essentially learning a mapping between feature space and output space, which is then used to map unknown instances in test data to values in the output space. Let $\mathbb{X}$ be the feature space, and $\mathbb{Y}$ be the output space, the training data with n instances could be denoted as $D = \{(x_i, y_i)|1 \leq i \leq n\}$. Here x_i is a vector consists of multiple features such as *age*, *gender*, and *income* of a customer, and y_i is the corresponding output like whether he/she will close their account or not. The task of supervised learning could thus be formulated as learning the function $f_\theta : \mathbb{X} \rightarrow \mathbb{Y}$, which is then used to predict the output $\hat{y} \leftarrow f_\theta(x)$ for a given new instance x. The specific function is determined by parameters θ, which should be estimated using training data D. Nowadays, supervised learning is the most widely adopted data mining and machine learning technique, and have played an important role in many applications including spam detection (Ren and Ji 2017; Olatunji 2019), customer churn prediction (Brownlow et al. 2018a; Huang et al. 2015), image classification (Sun et al. 2019), opinion mining (Liu 2012; Hemmatian and Sohrabi 2019), protein analysis (Baldi and Pollastri 2002), social network analysis (Ahmed et al. 2016), and online recommendation (Wang et al. 2018; Ying et al. 2018), to name a few.

Classification is a special type of supervised learning with its output space consisting of discrete values. In specific, the output space consists of a set of labels $Y = \{y_1, y_2, \ldots, y_k\}$ in which k is the number of possible labels. For example, a tumour could be diagnosed as *benign* or *malignant*; a new report could be categorised as *politics*, *sports*, and *business* etc. Traditionally, it is assumed that an object could only have one label, and it is called binary classification when there are only two possible labels, or multi-class classification when there are more possible labels. The aim of classification is obviously to learn a mapping $f_\theta : \mathbb{X} \rightarrow Y$. Alternatively, it could also be regarded as learning the conditional probability of $p(y|x)$ given x from a probabilistic perspective. In this way, methods of classification could be *generative* or *discriminative* (Murphy 2012). Generative methods begin with learning the joint probability $p(x, y)$ and use Bayes theorem to obtain the conditional probability $p(y|x)$, while discriminative methods learn $p(y|x)$ directly. In terms of their assumptions about data, those methods could fall into the following major categories.

Instance-based models that assume similar instances in the same feature space should have similar outputs. The key is how to represent data. The basic k-nearest neighbors (KNN) method calculates similarities between data in the original feature space using measures like Euclidean distance (Wu et al. 2008). By contrast, the kernel method maps data into a higher space in which they are more separable by adopting various kernel functions including Gaussian kernel and Fisher kernel, etc (Shawe-Taylor et al. 2004).

Linear models which assume that underlying data are linearly separable, and predict the output as a function of linear combination of the features. Its general form is as $p(y|x) = f(\theta^T x)$. The most representative linear method for binary classification is logistic regression in which sigmoid function is adopted as f . It is thus defined as below.

$$p(y|x) = \frac{1}{1 + e^{-\theta^T x}} \tag{2.1}$$

Softmax regression is an extension of logistic regression for multi-class classification where $y \in \{1, \ldots, k\}$. Specifically, the probability of having the ith label is determined by

$$p(y = i|x) = \frac{e^{\theta_i^T x}}{\sum_{j=1}^{k} e^{\theta_j^T x}} \tag{2.2}$$

These two methods are special cases of a boarder family of models, called Generalized Linear Models (GLMs). More cases could be found in Bishop (2006). Besides, Support Vector Machine (SVM) is another classical linear method that tries to learn the decision boundary that represents the largest separation between two classes (Schlkopf et al. 2018).

Tree-based models that assume data are not linearly separable. Instead of learning a single decision boundary as in linear methods, tree models recursively partition the feature space into a set of rectangles. Figure 2.2 shows an example of a binary decision tree and how it can classify data that are not linearly separable. The core of those methods is the criteria for

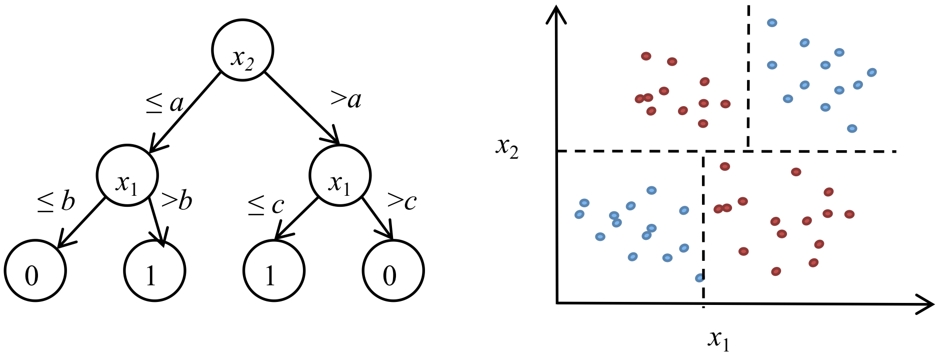


Figure 2.2: An toy example of decision tree and its partition

Instead of building a single tree, ensemble learning builds and combines multiple trees to make prediction (Zhou 2012). The decision tree is mostly used to build models in ensemble learning because it could generate low-bias but high-variance models, and ensemble learning combines multiple such models to reduce the overall variance, resulting in an accuracy improvement. One key to ensemble learning is how to generate multiple diverse models. To this end, several strategies are proposed, including 1) different data, such as the multiple samples of the training data generated using bootstrap in Bagging (Breiman 1996); 2) different sets of features as Random forest uses randomly selected features for every tree Breiman (2001); 3) different weights of instances like those "hard" instances are given more weights in next round of building a tree (Freund and Schapire 1996); 4) different learning objectives such as fitting the current residual in gradient boosting machine (Friedman et al. 2001). Nowadays, ensemble learning has become the most effective method of improving performance, and several efficient implementations have been widely used in real production and data competitions such as XGBoost, LightGBM, and CatBoost etc(Chen and Guestrin 2016; Ke et al. 2017; Prokhorenkova et al. 2018).

Neural network-based models that could theoretically simulate any continuous function other than just linear function, thus could handle data of any level of complexity. Originated from the last 60s, neural network is enjoying a renaissance in recent years because of the development of powerful computing techniques and hardware (Schmidhuber 2015). Figure 2.3 shows a basic neural network, i.e., multilayer perception, which could multiple layers of hidden units. After several rounds of transformations of the inputs, those high-level features represented by hidden units are more discriminative, leading to the underlying pattern more learnable. Besides the deep forward

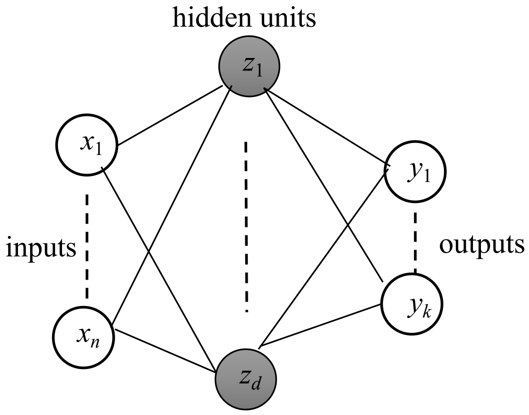


Figure 2.3: Multilayer perception with a hidden layer

Neural networks and aforementioned ensemble learning are two ways of learning composite models. The former achieves that in a multiplicative way as $y = f_n(f_{n_1}(\dots f_1(x)))$, while the latter essentially is an additive function as $y = \sum_{i=}^{n} w_i f_i(x)$. Recently, methods that borrow ideas from both of them have been proposed, e.g., deep forest, to train decision trees in multiple layers

(Zhou and Feng 2017; Feng et al. 2018).

2.4 Multi-label learning

Objects in reality tend to fall into multiple categories from different perspectives. For example, an image could be annotated as *outdoor* and *sunset*; a news report could be about *politics*, *economy*, and *domestic*, etc. Correspondingly, multi-label learning assumes that an instance could be associated with multiple labels simultaneously, and build models to predict all its relevant labels instead of just single one. Let $\mathbb{X}^d$ be a d-dimensional feature space, $\mathcal{Y} = \{y_1, y_2, \ldots, y_m\}$ be the set of all possible labels, $D = \{(x_i, Y_i) | 1 \leq i \leq n\}$ is then a labelled dataset in which $Y_i \subseteq \mathcal{Y}$ are the instance x_i's relevant/true labels. Multi-label learning thus could be formally defined as learning a mapping function $f(X) \rightarrow 2^C$ using D, i.e., the mapping from the space $\mathbb{X}$ to a space that consists of all the possible subsets of $\mathcal{Y}$.

Since being applied to text classification (McCallum 1999), a multitude of methods have been proposed to deal with a board range of multi-label data, including gene series (Zhang and Zhou 2006; Clare and King 2001), video (Qi et al. 2007; Hou et al. 2016), and image (Boutell et al. 2004; Wu et al. 2010), etc. Those methods are well summarized in Tsoumakas et al. (2010) and (Zhang and Zhou 2014), and they deal with the multi-label learning problem from following primary perspectives.

2.4.1 Algorithm adaption

The algorithm adaption strategy adapts traditional methods for single-label classification, to enable them to deal with multi-label data. The majority of classical methods have already been extended at the moment. For example, the k-NN algorithm is extended by using maximum a posteriori principle to determine the label set for unseen instances (Zhang and Zhou 2007). Based on this work, a framework is also proposed to unify instance-based learning, i.e., k-NN, and logistic regression, where label dependencies are exploited by taking the labels of neighbors as additional features (Cheng and Hullermeier 2009). In terms of decision tree, the C4.5 method is adapted where entropy is redefined to cater for multiple labels (Clare and King 2001), and alternating decision tree is used in conjunction with the boosting method in the ADTboost.MH method (De Comite et al. 2003). AdaBoost.MH and AdaBoost.MR have been designed based on the boosting method (Schapire and Singer 2000). The former reduces multi-label data to binary data and reduce Hamming loss, whereas the latter places relevant labels before irrelevant labels by minimizing ranking loss. Classical statistical models are also explored, such as Bayesian network (Gaag and de Waal 2006; Bielza et al. 2011), mixed model (McCallum 1999; Streich and Buhmann 2008), and Bayesian non-parametric method (Xuan et al. 2017), and top models (Rai et al. 2015; Burkhardt and Kramer 2019) etc. In terms of neural networks, a new loss function is incorporated into the basic feed-forward neural network to predict a better label ranking (Zhang and Zhou 2006). Other state-of-the-art deep learning techniques have been applied to multi-label learning as well, including the aforementioned deep neural network, CNN, and RNN (Yeh et al. 2017; Lyu et al. 2019; Wang et al. 2016a) etc.

Besides above methods, a number of other important machine learning paradigms, other than supervised learning, have also been incorporated into the process of multi-label learning. These paradigms include transductive learning (Feng and Xu 2010; Kong et al. 2013), semi-supervised learning (Cevikalp et al. 2020; Zhan and Zhang 2017; Niu et al. 2019; Tang et al. 2019), active learning (Yang et al. 2009; Cherman et al. 2019; Wu et al. 2020), cost-sensitive learning (Lin 2019; Teisseyre et al. 2019), multi-task learning(Huang et al. 2013; Sener and Koltun 2018), and transfer learning (Han et al. 2010; Banerjee et al. 2019) etc.

2.4.2 Problem transformation

Problem transformation converts multi-label data into single-label data, so existing methods could be applied directly. Intuitive methods, such as remov-

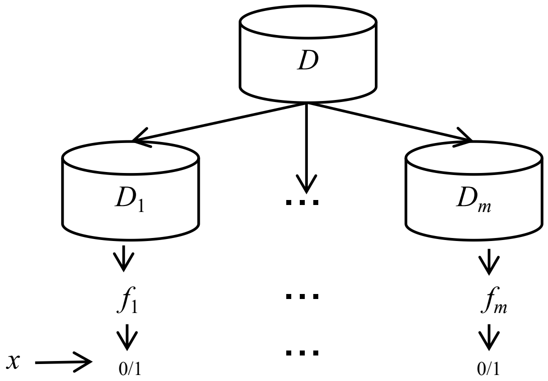


Figure 2.4: Binary relevance transformation

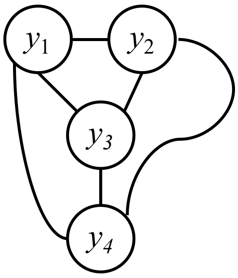
Multi-label to binary

This strategy converts multi-label data D into multiple binary data. For example, D could be converted into m datasets, where D_k corresponds to label y_k, is represented by $D_k = \{(x_i, c_i) | 1 \leq i \leq n\}$. Here D_k has the same instances as in D, while x_i's label c_i could be:

$$c_i = \begin{cases} 1 & \text{if } c_i \in Y_i \\ 0 & \text{if } c_i \notin Y_i \end{cases} \tag{2.3}$$

That is c_i is 1 if it is one of x_i's relevant labels, 0 if not. Then, models could be built to make predictions for corresponding labels respectively, and a multi-label learning problem is thus addressed as m binary classification problems as shown in Figure 2.4. This strategy is called binary relevance (BR), and have been adopted in various methods (Zhang et al. 2018). Although being straightforward, its drawback is that labels are treated respectively, resulting in dependencies among labels are ignored.

A large number of methods have been proposed to represent and exploit label dependency to improve model performance (Dembczynski et al. 2010b, 2012). A fair number of those methods which extend the BR method follow



(c)

Figure 2.5: Examples of label dependency structure: (a) Classifier chain, (b) Bayesian network, and (c)Fully connected network

Figure 2.5 shows three typical structures that have been explored. The first one is the classifier chain (CC) method, in which labels are randomly sorted and structured as a chain (Read et al. 2011). It then assumes that every label depends on all the preceding labels, i.e.,

$$pa(y_{r(k)}) = \{y_{r(j)} | 1 \leq j \leq k-1\} \tag{2.5}$$

A potential issue of this method is that it might result in improper label dependency, because of the randomness in sorting labels. Therefore, the probabilistic classifier chain (PCC) is proposed to learn the optimal order of labels by maximizing labels' joint probability (Dembczynski et al. 2010a). Additionally, conditional entropy between labels has been used to generate a better order of labels (Jun et al. 2019).

The second one is to learn a Bayesian network of labels as the middle one in Figure 2.5, and assume that every label depends only on its direct parents. Zhang et al. proposed the LEAD method in which a full Bayesian network of labels is built to model conditional dependencies between labels (Zhang and Zhang 2010). Restricted Bayesian network, e.g., a tree-structure of labels, is built to simplify the process of learning the Bayesian network, and every label is assumed to depend only on one other label (Fu et al. 2012). Similar ideas of building a tree structure have also been investigated in other methods (Sucar et al. 2014; Wu et al. 2016).

The last one model mutual dependencies between labels using undirected graph. Guo et al built a fully connected conditional dependency network over labels as the right one in Figure 2.5 (Guo and Gu 2011). It assumes every label depends on all other labels and learns a model $f(x, y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_m)$ for every label y_i. Besides original data, label dependencies could also derive directly from external knowledge graph such as *WordNet* (Lee et al. 2018).

Multi-label to multi-class

This strategy converts multi-label data into multi-class data. An intuitive method is to treat the set of an instance's all labels as a new single label. In this case, the label space is the power set of the original label space, i.e., $2^{\mathcal{Y}}$, so there are 2^m possible new labels. Despite being able to utilize label dependencies, this method suffers from the exponentially growing number of possible labels which leads to few or even no instances associated with some labels. The RAkEL model was proposed to deal with this issue, in which the original set of labels was divided into subsets of size k, based on each of which a model was built using LP transformation respectively (Tsoumakas et al. 2011). In this way, the number of labels for every sub-problem was reduced to $2^k \ll 2^m$. Tenenboim et al. similarly designed a method to split labels into subsets of varying sizes (Tenenboim et al. 2009). To void the randomness introduced in the RAkEL model, statistics measures like *Chi-square* and *Phi* coefficient were adopted to quantify label dependencies and thus identify more reliable label subsets (Tenenboim et al. 2010; Tsoumakas et al. 2009). Instead of solving multi-label learning as multiple multi-class sub-problems, the PS model proposed by Read (2008a,b) filtered out labels with a frequency less than a predefined threshold p to reduce the number of possible labels.

Multi-label to label ranking

For a given instance, methods using this strategy output a ranking of all labels in terms of their probabilities of being relevant labels, instead of a set of discrete labels. The mapping function therefore is $f(x) \to \langle r_1, \ldots, r_m \rangle$, in which r_j usually is a value in $[0, 1]$ that represents the probability of label y_j being x's relevant labels.

Label ranking could be learned at the following primary levels. Firstly, the preference between every pair of labels could be learned, and the final ranking is formed by combining all those pairwise preferences (Hullermeier et al. 2008; Madjarov et al. 2012). Instead of learning a full ranking, some methods learn only a partial ranking that aims to rank all relevant labels over irrelevant labels for any instance, while orders of labels within the group do not matter. For example, the Rank-SVM is built by maximizing the minimum margin shown in Equation (2.6) which represents the shortest distance between relevant and irrelevant labels (Elisseeff and Weston 2001).

$$\min_{(x_k, Y_k) \in D} \min_{(y_i, y_j) \in Y_k \times \overline{Y_k}} \frac{\langle w_i - w_j, x \rangle + b_i - b_j}{\|w\|} \tag{2.6}$$

Similar loss functions have also been integrated into extensions of other models for this purpose, including neural network and boosting, etc (Zhang and Zhou 2006; Freund et al. 2003). Lastly, a full label ranking could be learned directly as well. For example, the random with restart (RWR) principle is employed to adjust the relative weights of labels iteratively until convergence (Fu et al. 2012, 2013). Some typical methods of label ranking can also be found in (Vembu and Gärtner 2010).

2.5 Imbalanced learning

Imbalanced data means its instances are unequally distributed across different classes. It is very common in practical data mining applications, including the research we conduct in this thesis. For example, given a group of customers, it is likely that only 1% of them close their accounts (class 1, and the other 99% are of class 0). Let $D = \{D_P, D_N\}$ be a dataset, in which D_P and D_N denote the instances of minority class and majority respectively, imbalanced data formally refers to $|D_P| \ll |D_N|$.

Imbalance data presents a challenge for data mining and building models since the learning process might be extremely skewed towards the dominant class. Consequently, the resulting model could not make accurate predictions for instances of the minor class, which people are more concerned about in most cases. For example, a model that predicts all aforementioned customers are of class 0 will have a 99% accuracy, but none of those close their accounts is identified successfully. Therefore, it is critical to have the minor class well-represented in the learning process. To this end, various methods have been proposed in past years, and most of them adopt one of the following strategies.

2.5.1 Resampling

Methods using this principle modify the distribution of instances to obtain a balanced dataset by sampling from the original data. It could be realized in two ways, i.e., over-sampling and under-sampling which have proved to be effective (Zhou and Liu 2006).

The basic over-sampling augments the data of minority class by copying some of them randomly, i.e., an additional dataset S, which is sampled from D_P, is added into D. As a result, the new data $D^* = \{D^*_P, D^*_N\}$ where $D^*_N = D_N$ and $D^*_P = D_P \cup S$. $|S|$ is determined by a predetermined ratio $\rho = |D^*_P|/|D^*_N|$ which is normally set to be 1 to obtain a balanced dataset. Besides sampling uniformly, instances could be assigned with different weights.

Brownlow et al. (2018a) weight customers that close their accounts in terms of their account balances to focus on retaining more valuable customers. Instead of copying existing data, synthetic data could be generated as well. For example, a synthetic instance could be created using Equation (2.7) as in the synthetic minority oversampling technique (SMOTE) (Fernández et al. 2018; He and Garcia 2009).

$$x_{new} = x_i + \delta \times (\hat{x_i} - x_i) \tag{2.7}$$

Here x_i is an instance of minority class and $\hat{x_i}$ is one of its k neighbours, and δ is a random number falls into $[0, 1]$.

By contrast, under-sampling tries to reduce the size of data of the majority class to obtain a more balanced dataset. A simple method is to remove instances selected uniformly, i.e., only keep a random sample S from D_N. This issue of this method is that useful knowledge of the majority class might be discarded, so a number of methods were proposed to keep more representative instances and remove those less important. For example, cluster-based methods divide data into clusters within which instances are similar to each other and keep only the cluster centroids (Lin et al. 2017). kNN method is also used to identify and remove less discriminative instances of majority class which are similar to someones of minority class (Mani and Zhang 2003). The SVM method is also extended to select informative instances by active learning (Ertekin et al. 2007).

Another mechanism of alleviating the issue of knowledge loss is the combination of down-sampling and ensemble learning (Liu and Zhou 2013; Galar et al. 2011). The basic paradigm is to create multiple samples of majority class $S_{i(1 \leq i \leq k)}$ to form multiple more balanced datasets $D_i^* = \{D_P, S_i\}_{(1 \leq i \leq k)}$, then models based on them are built and combined using the ensemble learning technique. Consequently, most of the instances in D_N are likely to be kept with the increasing of the number of samples k. For example, multiple samples S_i ($|S_i| = |D_P|$) are generated independently in the process of building random forest and bagging models (Chen et al. 2004). Independent sampling is essentially an unsupervised strategy since the effectiveness of a sample is not reviewed and used to guide the generation of other samples. To address this issue, sequential sampling using the boosting method was proposed to create samples that are informative as well as diverse (Liu et al. 2009; Galar et al. 2013). Specifically, in the iteration of creating sample S_i, any instance x in S_i which is correctly classified by the corresponding model f_i will be excluded from D. The idea is that knowledge embedded in x has been captured by f, so it is redundant and should not be included in the following samples.

2.5.2 Cost-sensitive learning

Cost-sensitive learning is another commonly adopted strategy for dealing with imbalanced data. It assumes the cost of misclassifying instances of the minority class is way higher than that of misclassifying instances of the majority class. In consequence, the model would be biased towards making accurate predictions for instances of minority class to reduce overall cost.

Generally, cost-sensitive learning methods could fall into the data level and the algorithm level. Methods of data level mostly adjust the weights or distribution of instances in terms of misclassification cost. A predefined cost matrix, which specifies costs of different types of misclassification, is necessary in this case (Elkan 2001). For example, the method *MetaCost* reassigns every instance to its optimal class that minimizes the expected cost of misclassification using Bayes optimal prediction (Domingos 1999). It is thus expected that some instances of majority class will be relabelled as minority class if the cost matrix is set appropriately, resulting in a more balanced data. Misclassification cost could also be used to adjust weights of instances (Zadrozny et al. 2003). For example, the weighting updating rule in boosting could be modified to $D_{t+1}(i) = D_t(i)\exp(-a_t C_i h_t(x_i) y_i)/Z_t$, so instances x_i's weight is proportional to its misclassification cost C_i (Sun et al. 2007). The drawback of these methods is that a cost matrix is difficult to obtain in many cases, and a practical solution is to set the misclassification cost of a particular class P inversely proportional to its size, i.e., $|D_P|$ (Chen and Guestrin 2016).

As to the algorithm level, many classical methods have been extended to become cost-sensitive using the following strategies. Firstly, the decision threshold could be adjusted to make it harder to misclassify instances of minority class (Weiss 2004). ROC analysis has been used to decide the optimal threshold by Maloof (2003). This strategy manipulates outputs of a model rather than data compared with the resampling methods. Secondly, a cost-sensitive loss function could be adopted to guide the learning process towards making accurate predictions, particularly for those costly instances. For example, loss functions that minimize the expected cost have been adopted in learning deep neural network(Khan et al. 2017; Wang et al. 2016b). Thirdly, extremely imbalanced data could be addressed as special types of machine learning problems such as one-class learning (Alam et al. 2020) and outlier detection (Boukerche et al. 2020).

Compared with resampling, cost-sensitive learning is more computationally efficient since it does not involve copying massive data. However, it needs domain knowledge to design the cost matrix, and the strategies adopted are only limited to specific learning algorithms.

2.5.3 Incorporating auxiliary data

Another way of dealing with imbalanced data is to exploit auxiliary data. It is less common compared with resampling and cost-sensitive learning, but is very promising with the availability of big data. The research in chapter 4 is actually inspired by this idea. Different from over-sampling that replicates data in labeled dataset D, an additional dataset S from other sources could be exploited to alleviate the imbalance issue. However, this dataset S cannot be used directly in most cases, because it is either unlabelled or of different distribution. i.e., $P(D) \neq P(S)$.

The semi-supervised technique has been investigated to deal with unlabelled data. For example, co-training methods are used to identify informative unlabelled data (Li et al. 2011). Graph-based semi-supervised learning is employed to find unlabelled instances that are similar to those of minority class (Zheng and Skillicorn 2016). Active learning has also been utilized to select unlabelled instances that are most likely to be of minority class (Li et al. 2012).

Transfer learning studies on how to learn data from different sources. Particularly, the instance-transfer based transferring has been applied to imbalanced data as well (Pan and Yang 2010). Two major strategies have been investigated. One is sampling data from auxiliary data by inferring the normalized distribution $\frac{P(S)}{P(D)}$ or casting auxiliary data into the target space, so sampled auxiliary data would have the same distribution of D (Brownlow et al. 2018b). The other is the boosting-based sampling, in which weights of auxiliary data are iteratively adjusted in terms of how they have helped classify the data in target data (Dai et al. 2007; Al-Stouhi and Reddy 2016). The research in Chapter 4 also follows this paradigm, the difference is that a novel cost-sensitive evaluation of prediction is proposed in our approach.

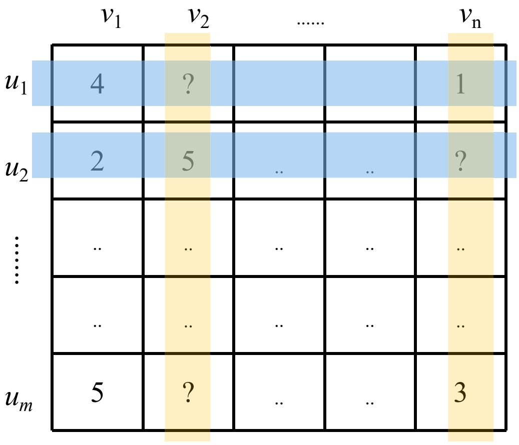
2.6 Recommendation system

The recommendation system is a type of technology to help customers find potentially desirable items when they are facing overwhelming information (Ricci et al. 2015). Arising as being used for movie recommendation (Sarwar et al. 2001), recommendation system has became prevalent nowadays and applied to recommending various types of items, including book (Ekstrand et al. 2018), music (Schedl 2019; Dror et al. 2011), friend in social networks (Verma et al. 2019), news (Zheng et al. 2018), youtube video (Covington et al. 2016), and hotels in Airbnb (Grbovic and Cheng 2018), etc.

The essence of recommendation system is finding items that match users' preferences. Therefore, two key elements of recommendation system are: (1) how to learn the representations of users and items; and (2) how to calculate the similarity between a user and an item, i.e., an estimation of the extent to which they match each other, in terms of their representations. A multitude of recommendation methods have been proposed with different strategies to solve these two key issues, and they are categorized and summarized in the following sections respectively.

2.6.1 Collaborative filtering

Neighbour-based collaborative filtering is the most straightforward and suc-
ce
us
m
m
u_i
co
fu
as
ite



	v_1	v_2	......		v_n
u_1	4	?			1
u_2	2	5	..	..	?
⋮	..	..	..	..	..
	..	..	..	..	..
u_m	5	?	..	..	3

Figure 2.6: An example of rating matrix

There are two major types of neighbor-base collaborative filtering techniques, i.e., user-based collaborative filtering and item-based collaborative filtering (Adomavicius and Tuzhilin 2005). The former assumes users with similar preferences should assign similar ratings the same item, while the latter assumes that a user should give similar ratings to similar items. Therefore, the key point is how to measure similarities between users/items. With a rating matrix R, user u_i and item v_j can be represented using relevant ratings, i.e., the i-th row $r_{i.}$ and j-th column $r_{.j}$ in R. Based on that, *Pearson correlation* and *cosine similarity* are the two most used measures to calculate similarities (Sarwar et al. 2001). In particular, *Pearson correlation* between two items is defined as in Equation (2.8), in which K is a set of users that both have rated item v_i and v_j, and $\overline{r_i}$ and $\overline{r_j}$ is the average rating respectively.

$$sim(i,j) = \frac{\sum_{k \in K}(r_{ki} - \overline{r_i})(r_{kj} - \overline{r_j})}{\sqrt{\sum_{k \in K}(r_{ki} - \overline{r_i})^2}\sqrt{\sum_{k \in K}(r_{kj} - \overline{r_j})^2}} \tag{2.8}$$

Cosine similarity between two items is defined in Equation (2.9)

$$sim(i,j) = cos(r_{.i}, r_{.j}) = \frac{r_{.i} \cdot r_{.j}}{||r_{.i}||_2 * ||r_{.j}||_2} \tag{2.9}$$

Since the sparsity of rating matrices, content-based filtering has also been investigated, in which additional data, e.g. user profiles and characteristics of items, are explored as well to achieve more accurate calculation of similarities. For example, topics embedded in the text are inferred in news recommendation (Lu et al. 2015). Acoustic features are extracted to better represent a piece of music Shao et al. (2009). Users' demographic information and genres of movies are utilized in movie recommendation (Li et al. 2015a).

2.6.2 Factorization-based model

In contrast to collaborative filtering, factorization-based models map users and features into the same latent feature space, enabling them to be compared directly. Figure 2.7 gives an illustration of the basic matrix factorization (MF) model (Koren and Volinsky 2009), in which rating matrix R is decomposed as $R = P^TQ$. As a result, user u_i is represented by the column P_i in matrix $P_{k \times n}$, and item v_j is represented by the column Q_j in matrix $Q_{k \times m}$. In this way, the similarity between u_i and v_j or the rating r_{ij} can be easily estimated as the dot product of two vectors, i.e., $\hat{r_i}j = P_i \cdot Q_j$

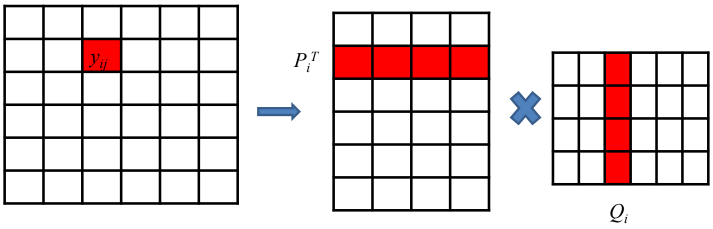


Figure 2.7: Matrix factorization of a rating matrix

Probabilistic MF model assumes that the rating matrix variable should follow a Gaussian distribution parametrised by P and Q (Salakhutdinov and Mnih 2008). So P and Q can be learned through minimising the loss function in Equation (2.10), in which $I^{ij} = 1$ if $r_{ij} > 0$, otherwise $I^{ij} = 0$.

$$L(P, Q) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} I^{ij} (r_{ij} - \hat{P}_i^T \hat{Q}_j)^2 + \lambda(||P||_F^2 + ||Q||_F^2) \tag{2.10}$$

To alleviate issues like sparse rating matrix and cold-starting problem, various extensions of the basic MF model have been proposed. For example, social relationships between users are utilized to learn similar latent features for socially related users (Ma et al. 2011; Ma 2013). Temporal factors are considered in location-based social network recommendation (Gao et al. 2013). User-generated contents, e.g., reviews, tags, and implicit feedback, have also been exploited extensively (Li et al. 2015b; Xu et al. 2011; Hu et al. 2008; Ren et al. 2014)

Compared with MF, factorization machines (FM) is a more general factorization model (Rendle 2012). Its definition is given in Equation (2.11).

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i \cdot x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle \, x_i \cdot x_j \tag{2.11}$$

Here v_i is the latent factor associated with x_i. We can see that MF is a special type of FM, and FM is a natural extension that can take additional features into consideration.

2.7 Summary

This chapter introduces the foundations and techniques related to research conducted in this thesis, which provide the theoretical and technical basis for the research conducted in this thesis and future work.

To begin with, the basic concepts and the main components of the data mining process are introduced, including data collection, feature extraction, and model building, etc. Next, classical methods of classification are summarized into categories, and a particular type of classification, i.e., multi-label learning is presented in detail which is related to our research in chapter 2. Followed by that, methods of dealing with imbalanced data are analyzed which motivate our proposed method in chapter 3. Finally, the analysis of representative methods for recommendation is given which related to the research in chapter 5.

Chapter 3

Learning customers' multiple needs for financial adviser

Understanding why customers need a financial adviser is critical to financial services due to its great impact on people's financial wellness and retirement readiness. Nowadays, many general methods of identifying customer needs have been designed. However, these methods cannot be applied directly to inferring customers' needs for financial advisers. The challenges include a lack of explicit customer needs and difficulty in inferring individual needs for mass customers. Therefore, an end-to-end framework that treats customer needs identification as a data mining application is proposed in this chapter. To begin with, a group of customers is selected with features and needs being extracted from data of multiple sources and expert knowledge. Based on this labeled dataset, a multi-label predictive model is built, which is then used to infer other customers' needs for financial advisers. In this way, customers' individual needs can be inferred at scale thanks to the automatic model. The framework is applied to real-world data and shows good performance.

The remainder of this chapter is structured as follows. Firstly, background and motivations are introduced. Related preliminaries and a formal definition of the problem are then given. Next, our method is described in detail. experiments are also presented, followed by conclusions finally.

3.1 Introduction

Understanding customer needs is at the core of a business's customer relationship management and marketing activities. It helps provide customers with more desirable products and services, resulting in customer satisfaction improvement and long-term profitability. In practice, a variety of businesses have been profiting from understanding and meeting customer needs, like banking (Stringfellow et al. 2004), insurance (Bae et al. 2005), IT services (Eckstein et al. 2016), social media (Kramer 2016; Kühl et al. 2019), and online retailers (Hu and Liu 2004), and so forth.

In the field of fund management, people tend to seek professional advice from financial advisers or delegate an adviser to invest in funds provided by fund providers. Generally speaking, people should have different needs or expectations when they turn to a financial adviser, because of their own particular circumstances. Obviously, it is fundamental to have a clear understanding of their needs for fund providers and financial planning services, since millions of dollars could be involved and the great impact on customers' financial well-being and retirement readiness.

A diversity of methods and tools have already been devised in practice to identify customer needs in both academia and industry. Traditional methods like questionnaires, surveys, and face-to-face interviews are used to learn customers' views on products and services (Edvardsson et al. 2012; Wang et al. 2010). Usually, intensive manual work and domain experts are required in these methods to gather customer data and identify customer needs. With the advent of big data techniques, more sources of customer data are available such as transactional databases, social media, and online reviews. Accordingly, advanced data mining techniques are investigated to learning customer needs efficiently (Stringfellow et al. 2004; Zhu et al. 2011; Timoshenko and Hauser 2019).

Despite those achievements, it is still very challenging for fund providers to understand customers' needs for financial advisers. Some of the main challenges are as follows. (1) Unlike businesses like telco companies that divide customers into segments and provide products to meet segment-level customer needs, fund providers and advisers should have personalized products that are tailored for every customer, because of their particular financial needs and risk tolerance. Customer needs thus should be identified individually instead of at the level of market segments. Given millions of customers a fund provider might have, it is impossible to infer customers' individual needs at scale using any methods involving manual work. (2) Compared with other needs, it is more difficult to identify people's financial needs for fund providers, because there are few data that could show customers' explicit needs or their interaction with advisers. By contrast, other businesses like online retailers have access to a huge number of customers' needs that are embedded in their reviews and ratings. Moreover, many customers are not able to articulate or even be conscious of their financial needs due to a low level of financial literacy (Worthington 2013). (3) Besides inferring the needs of existing customers, it is important as well to anticipate the needs of new customers to help them find suitable advisers and provide them with satisfying onboarding experience.

To address the above challenges, this chapter designs an automatic framework to infer customers' financial needs for advisers using data mining techniques. It follows the process of data mining as depicted in Chapter 2, and no manual work is involved. To create a labeled dataset for building models, a variety of features are defined and extracted from heterogeneous data sources such as customer profile and behavior, account status and performance, and others. Moreover, three possible financial needs, i.e., labels, are predefined based on domain knowledge and several assumptions we proposed. Based on a group of selective members with their features and labels being generated, the multi-label learning technique is then employed to build a predictive model, which can automatically infer the financial needs of customers at scale. The key point here is how the group of customers for building models is selected, so that needs to be estimated for them are most likely to be their true needs and the model built upon them is accordingly as accurate as possible. To this end, several principles are designed to select customers and define needs based on exiting research on the value of financial advice (Marsden et al. 2011; Foerster et al. 2014).

To summarize, the main contributions of our method in this chapter include: (1) An automatic framework of learning customers' needs for financial advisers at scale is proposed and implemented in practice. (2) Principles of selecting customers and inferring their needs/labels are designed, and financial needs prediction is thus addressed as a multi-label learning problem. (3) Extensive experiments with real-world data have been conducted to validate the effectiveness of the proposed framework.

3.2 Problem statement

This section presents the notations used and a formal statement of the problem solved in this chapter.

Given a set of customers $\mathcal{X} = \{x_1, x_2, \dots x_a\}$ and multiple data sources $\{S_1, S_2. \dots, S_b\}$ that contain data relating to X, the aim here is to learn a predictive model $f(x) \to y$, to infer his/her possible needs for financial advisers, given a customer x. $\hat{y}$ is a binary or numeric vector that indicates which needs x has or the probabilities of those needs being his/her true needs. This could be regarded as a multi-label learning problem as described in Chapter 2 if a group of customers whose needs are known could be obtained as a labeled training data. According to the data mining process shown in Figure 2.1, following sub-problems should be solved.

(1) Selection a group of customers $D \in \mathcal{X}$ which are appropriate for training models.

(2) construction a comprehensive view of instances in D using data sources $\{S_1, S_2. \dots, S_b\}$.

(3) Reasonable definitions of possible financial needs $\{c_1, c_2, \dots, c_m\}$ and generation of needs/labels for instances in D.

(4) Learning model $f_D(x) \to y$ to be applied to inferring needs of other

customers such as those in $\mathcal{X} - D$.

Our method to solve above problems is given in the following section. It should be noted that although there are similar methods that also apply data mining techniques to customer needs prediction (Kuehl et al. 2016; Kühl et al. 2019), the differences are: (1) needs/labels in training data are generated automatically in our method without involving any manual work, and (2) needs prediction is addressed as a multi-label ranking problem in our method rather than a binary or multi-class classification problem.

3.3 The process of learning customer needs

This section begins with a brief of the overall framework of learning customs needs, then introduces the details of each step respectively.

3.3.1 Overall framework

The process of predicting customer needs by discovering knowledge of a group customers whose needs could be extracted from existing data is illustrated in Figure 3.1. This framework follows the typical steps of a supervised learning process. Roughly speaking, a set of labeled instances represented by a pair of features and labels is generated to begin with. Then is the iterative process of building and evaluating models with different parameters to obtain the optimal one in terms of particular criteria. Finally, the model is used to predict labels for those new instances whose labels are unknown. In the setting of learning customer needs, instances refer to the customers, and labels are those particular needs. Specifically, in the first 3 steps in Figure 3.1, i.e., (1) customer selection, (2) feature engineering, and (3) needs calculation, a specific set of customers is selected and their features and labels are extracted to form a labeled dataset. Subsequently, a multi-label learning model is built in step (4) to predict a ranking of needs for those customers fed into step 5. Details of each step are described in the following subsections.

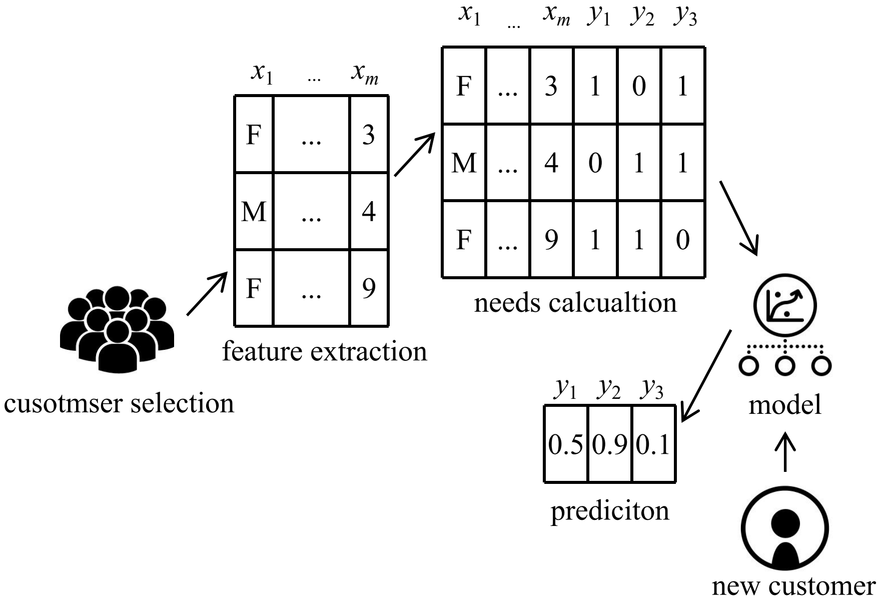


Figure 3.1: The framework of learning multiple customer needs

3.3.2 Customer selection

Different from other scenarios in which true labels are available, the labels, i.e., needs for financial advisers, are also inferred from data such as customer profiles and behaviors in our method. As a consequence, a potential issue is that incorrect needs/labels might be obtained if some customers' behavior is inconsistent with their true needs. For example, some financially illiterate customers may never save for retirement, though they do should have. Therefore, the first step is to select a group of customers whose needs are more likely to be correctly inferred from data, to increase the chance of building accurate models.

Based on studies in (Marsden et al. 2011; Foerster et al. 2014) which prove that financial advisers do help people in retirement planning and investment activities, several principles as below are designed to help select a group of customers and infer their labels.

(1) The first one is that those selected customers should have engaged with a financial adviser constantly for a period of time which is called a label window in our method. The assumption is that their behaviors are more

likely to be consistent with their needs with the help of advisers, thus the needs inferred form their behavior is more likely to be true.

(2) Moreover, those selected customers should keep staying with the same adviser for a period of time after the label window. The purpose is to make sure some of their needs during the observation time are indeed satisfied, so the estimated needs are reliable. For example, only if a customer still stays with the same adviser for several months after a poor annual investment return, it could be inferred that short-term return is not his/her need.

(3) To further reduce the chance of inferring incorrect needs, the third principle is that we only infer the relative importance of needs, instead of

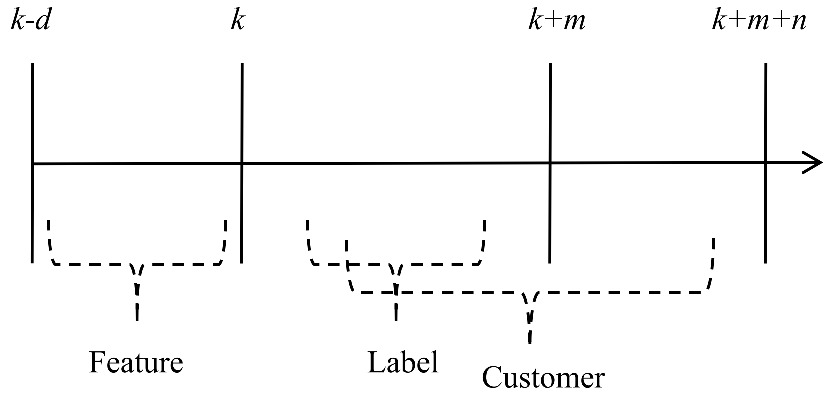


Figure 3.2: The time window to determine customers, features, and labels

To be specific, observation time in Figure 3.2 is month k to month $k+m$, and customer needs are estimated using data during this period. Only customers that had an adviser at month k and still stayed with the same adviser at month $k + m + n$ could be selected. This allows an additional n months to make sure the estimated needs are reliable.

3.3.3 Feature engineering

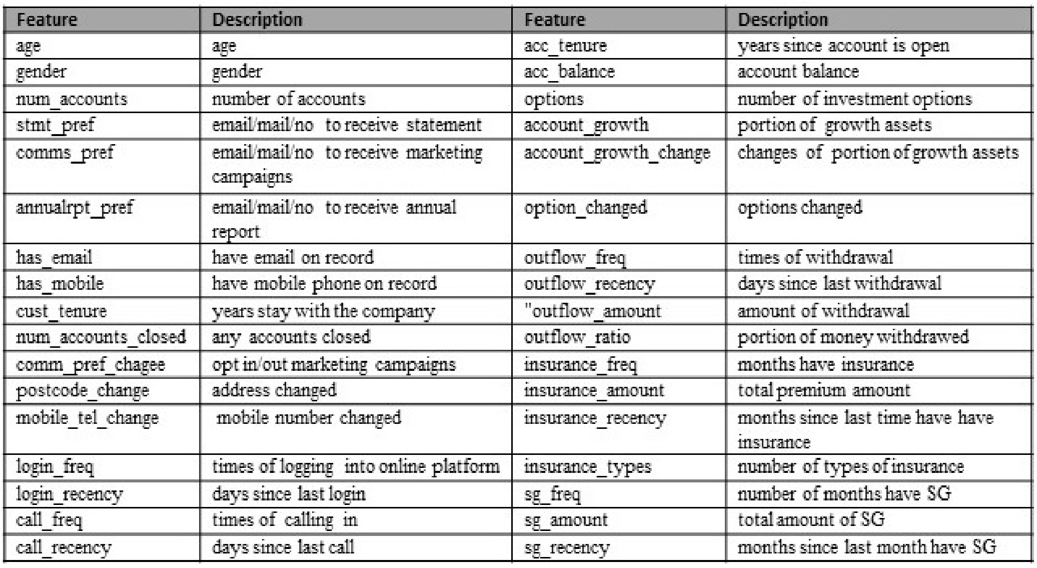
Features are information such as age and income that could be determinants of customer needs. Normally, an instance is usually represented as a feature vector before learning. The more insightful features we could obtain about customers, the more likely we are able to learn what their needs are. Therefore, we explore various sources of data sources in a real superannuation fund provider in Australia to extract more useful features. In particular, the following types of features are defined in our implementation, and more features are listed in Figure 3.3.

(1) **Demographic features**. These features provide information with regard to customers' current profiles, such as their *gender*, *age*, *location*, and *job* etc.

(2) **Behavioural features**. These features are related to customers' behaviors or interactions with the financial adviser and fund provider. Typical features of this type includes *moving home*, *change job*, and *frequency of calls/logging into online system*, and so on.

(3) **Account features**. Two types of account level features are extracted. One relates to an account's current status, such as *tenure*, *risk level*, and *balance*. The other describes how an account has changed in the past, i.e., *balance change* and *option change* etc.

It should be noted that features related to customer profile and account status should be defined using data at a particular time point, i.e., the end of month k in Figure 3.2, and features related to customer behavior and account changes should be defined using data from the past d months. The purpose here is to use what customers look like at the moment and what they did in the past to predict their needs in the future.



Feature	Description	Feature	Description
age	age	acc_tenure	years since account is open
gender	gender	acc_balance	account balance
num_accounts	number of accounts	options	number of investment options
stmt_pref	email/mail/no to receive statement	account_growth	portion of growth assets
comms_pref	email/mail/no to receive marketing campaigns	account_growth_change	changes of portion of growth assets
annualrpt_pref	email/mail/no to receive annual report	option_changed	options changed
has_email	have email on record	outflow_freq	times of withdrawal
has_mobile	have mobile phone on record	outflow_recency	days since last withdrawal
cust_tenure	years stay with the company	"outflow_amount	amount of withdrawal
num_accounts_closed	any accounts closed	outflow_ratio	portion of money withdrawed
comm_pref_chagee	opt in/out marketing campaigns	insurance_freq	months have insurance
postcode_change	address changed	insurance_amount	total premium amount
mobile_tel_change	mobile number changed	insurance_recency	months since last time have have insurance
login_freq	times of logging into online platform	insurance_types	number of types of insurance
login_recency	days since last login	sg_freq	number of months have SG
call_freq	times of calling in	sg_amount	total amount of SG
call_recency	days since last call	sg_recency	months since last month have SG

Figure 3.3: Some features used in needs prediction

3.3.4 Needs definition and extraction

The following three financial needs are predefined using domain knowledge in this chapter. They could be inferred from data related to customer, adviser, and fund provider respectively.

(1) **Retirement preparation**. Customers that have this need should be concerned about their retirement readiness, thus they should be more likely to save extra money into their superannuation account. For example, people in Australian could save extra money as non-compulsory contributions, such as personal contribution and salary sacrifice. Under certain circumstances like financial hardship, they are also allowed to withdraw from their superannuation account before retirement. Specifically, we infer that a customer should have this need during a particular period of time if he has a positive inflow. Let a_{in} be the total amount of all non-compulsory contributions and a_{out} be the amount of withdrawals, that is, a customer should have the need of retirement preparation if $a_{in} > a_{out}$ is observed.

(2) **Financial delegation**. Due to a relatively low level of financial literacy or capacity, some customers may delegate a financial professional to

manage their fund investment and deal with fund providers. In the database of a fund provider, the behavior of an adviser on behalf of his/her customers could be observed by their call logging and transaction records. Therefore, we could infer that a customer should have this need during a particular period of time if he/she has no interaction with the fund provider, and all transactions and calls related to him/her are done by the adviser. Let b_{io} be the number of transactions and calls made by a customer, and b_{ad} be the number of transactions and calls made by his/her adviser, that is, a customer should have the need of financial delegation if $b_{io} = 0$ and $b_{ad} > 0$ is observed.

(3) **Investment return**. Customers who care about the short-term investment return should be more likely to manage to gain better performance with the help of financial advice. By contrast, people who care less about short-term investment return would hold assets that are more conservative and are less likely to achieve significantly better performance. Thus, it is reasonable to infer a customer's priority on investment return by observing the performance of his/her fund. Particularly, Equation (3.1) is used to calculate fund performance during a given period of time.

$$ROI = \frac{bal_{end} + flow_{in} - (bal_{begin} + flow_{out})}{(bal_{begin} + flow_{in})} \tag{3.1}$$

In Equation (3.1), bal_{begin} and bal_{end} are the fund balance at the beginning and end of a given time frame respectively, and $flow_{in}$ and $flow_{out}$ are the total amount of inflow and outflow during that time respectively. Then ROI of a fund is compared with a baseline to see whether it is a better outcome or not. Specifically, let asx be the Australia stock market performance measured by the change of ASX index, it is inferred that a customer should have the need of short term investment return, if $ROI > ASX * a$ is observed. Here a is an adjustable factor.

Table 3.1 shows an example of selected customers after following the first 3 steps in Figure 3.1. To further reduce the chance of introducing false labels, we assume that if a need is not observed for a customer, it does not necessarily mean the customer does not have that need, but only that he is more concerned with other needs. Take the first line in Table 3.1 as example, it only means that *retirement preparation* is more important than the other two needs to the customer. Therefore, the predictive model should be built to learn a correct ranking of needs for customers.

Table 3.1: An example of customers with extracted features and needs

age	*gender*	$\cdots$	*balance*	*retirement prep*	*delegation*	*invest return*
50	F	$\cdots$	5,000	1	0	0
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
30	M	$\cdots$	50,000	0	1	1

3.3.5 Model building and prediction

With a labeled dataset D as shown in Table 3.1, predicting needs for other customers can be addressed as a multi-label ranking problem. Technically, the aim is to learn a function $p(x) =< p_1(x), p_2(x), \ldots, p_m(x) >$, which can predict a score for every label relative to their importance to x.

After investigation, the LabelRank (LR) method proposed in (Fu et al. 2014) is adopted to predict need ranking in our method, given its effectiveness and easy to implement. Its part of building model is straightforward, i.e., building a model for every label using BR transformation. In our implementation, the boosting trees algorithm is used to learn the models. When making predictions, label dependencies are exploited through a Random Walk with Restart (RWR) process, which is well-known for its capacity of learning an appropriate ranking. Specifically, the ranking prediction is done through an iterative process as shown in Equation 3.2.

$$\begin{cases} f(x)^0 = \frac{1}{m} \cdot \mathbf{1}_{|m|} \\ f(x)^{t+1} = \alpha \cdot T \cdot f(x)^t + (1-\alpha) \cdot p(x) \end{cases} \tag{3.2}$$

Here $f(x)^t$ is the prediction $f(x)$'s value in t-th iteration. $f(x)$ is assigned with an initial value, i.e., $f(x)^0$ to begin with, then it is updated repeatedly until convergence. In each step of this iterative process, a more appropriate score is learned for every label, resulting in a good label ranking finally. It should be noted that our method has been implemented using real data from one of the major superannuation fund providers in Australia, and its effectiveness in needs prediction is given in the next section in detail.

3.4 Experiments

Experiments are conducted in this section. Firstly, dataset preparation and evaluation metrics are given. After that, the experimental results are analyzed to show the performance of our method and the insights discovered from the learning process.

3.4.1 Data preparation

To begin with, labelled data are generated according to description in previous section. In order to examine the performance variation of our method in predicting needs for different length of time, we select six different point of time, i.e., the end of Jun 2015, Dec 2017, Jun 2016, Dec 2017, Jun 2017, Dec 2018 to create 6 different groups of customers. For each of these points of time, we firstly select personal super customers that have an adviser at that time and still stay with the same adviser at that end Dec 2018. Then we keep customer who pay their advisers servicing fee monthly, to include only customer that are more likely to get help from their advisers. Next, their needs are inferred using data from that time to the end of Jun 2018. Finally, their features are extracted using data go back 12 months from that time. In this way, six different datasets are created with labels inferring from data in 6, 12, 18, 24, 30 and 36 months respectively. The factor a in inferring need of investment return is set to 1.1 at the moment, which means a good invest return should be at least 10% better than the market. Table 3.2 gives the summary of these six datasets in detail.

Table 3.2: Description of datasets

m	$\|D\|$	Retirement prep	Financial delegation	Investment return
6	50,014	21.26%	17.28%	36.82%
12	42,055	23.57%	27.44%	26.92%
18	35,154	26.32%	33.37%	49.41%
24	29,943	27.59%	36.69%	18.81%
30	25,564	29.21%	39.38%	30.23%
36	21,971	30.52%	39.31%	57.81%

In table 3.2, m is the number of months to infer customer needs, $|D|$ is the number of customers, and the other three columns show the percentage of customers have that corresponding need respectively. It can be observed that while the percentage of customer have better investment return fluctuates, both the percentages of customers have needs of retirement preparation and financial delegation reach stable finally at m =36, which means it should be sufficient to look at data of 3 years to identify these two needs for customers.

3.4.2 Evaluation metrics and settings

Since the LR method adopted in our method is to predict a ranking of needs, two ranking-based metrics, i.e., one-error and ranking-loss, are therefore selected to evaluate our model's effectiveness accordingly.

(1) **One-error**. It evaluates the fraction of instances in dataset $|D|$ whose top-ranked label is not a true label.

$$\text{One-error}(f, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\delta(\arg\max_{y\in C} f(x_i, y) \notin Y_i) \tag{3.3}$$

(2) **Ranking loss**. Given a set of $\{(y_a, y_b)|(y_a, y_b) \in Y_i \times \overline{Y_i}\}$ that consists of all possible pairs of a true label and an untrue label of any instance x_i,

ranking loss computes, on average, the ratio of label pairs in which the true label is ranked after the untrue label in the predicted ranking of labels. Its definition is shown in Equation (3.4).

$$\text{R-Loss}(f, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{1}{|Y_i|\left|\overline{Y_i}\right|}\left|\{(y_a, y_b)|f(x_i, y_a) \leq f(x_i, y_b), (y_a, y_b) \in Y_i \times \overline{Y_i}\}\right| \tag{3.4}$$

Given a ranking of labels predicted for an instance, one-error tells us the possibility that the top one is predicted incorrectly, and ranking loss tells us the possibility of predicting a wrong ranking for a randomly selected pair of labels.

Two classical multi-label learning methods, i.e., LBIR and ClassiferChain (CC), are used for comparison. Similar to LR, they both exploit label dependencies to improve performance as well (Cheng and Hullermeier 2009; Read et al. 2011).

3.4.3 Results analysis

5-folds cross validation is used for model evaluation, in which data are split into 5 subsets of the same size, and 5 models are built using 4 of the 5 subsets and evaluated using the remaining one. The 5 results are then averaged as the final result.

Label ranking prediction performance

Firstly, we compare the LR method adopted in our method with the baselines which are also used in (Fu et al. 2014) for comparison. Table 3.3 presents the result in details, in which the left 3 columns show the results in terms of one-error, and the right 3 columns show results in terms of ranking loss.

Firstly, It can be seen from table 3.3 that all three learning methods show pretty good and similar results. It indicates that the variety of features we design has captured well the important factors that determine customer

Table 3.3: Experimental results in terms of one-error and ranking loss

m	LR	LBIR	CC	LR	IBLR	CC
6	0.222	0.228	0.231	0.182	0.190	0.189
12	0.240	0.246	0.251	0.185	0.192	0.193
18	0.189	0.195	0.197	0.171	0.177	0.178
24	0.256	0.265	0.268	0.189	0.197	0.196
32	0.246	0.255	0.261	0.197	0.205	0.208
36	0.193	0.195	0.199	0.191	0.195	0.199

needs, so advanced methods can achieve almost similar performance. Secondly, the LR method sees a slight decrease in both metrics on all datasets, compared with the other two. It means it does have an advantage in learning label ranking. Thirdly, the results do not provide clear evidence that the model performance has a consistent relationship with the number of months for inferring needs in our method. That could mean the model does not necessarily perform better in predicting needs in a long time, probably due to the changing circumstances customers are experiencing.

Important factors to every need

Besides model performance, we also examine the important features in the learning process. The following Figure 3.4 - 3.6 present the top 10 most important features to the learning of every need, which reveals the underlying factors that have an impact on customers' needs for a financial adviser.

Firstly, Figure 3.4 shows that age, account balance, and SG, i.e., income are the most important factors to retirement preparation. This is quite consistent with our common understanding of retirement. For example, preretirees with low balance should be more likely to save more money for retirement. Figure 3.5 shows that while account growth, i.e., risk tolerance, is the

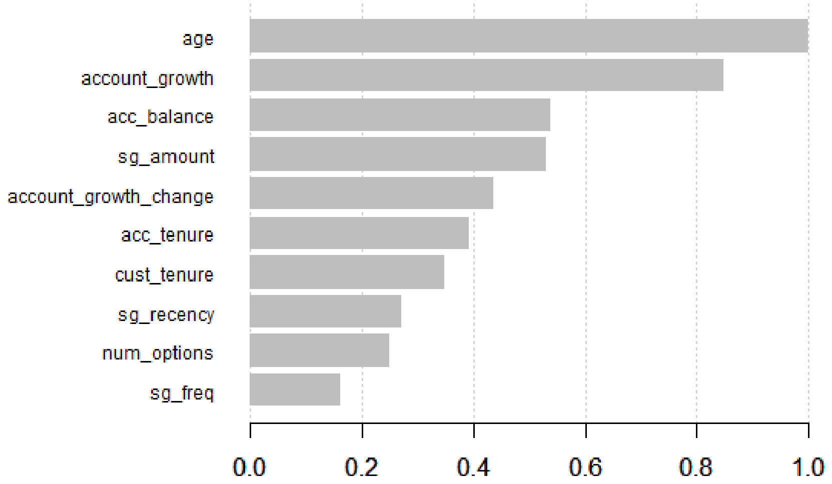


Figure 3.4: Top 10 important feature to retirement preparation

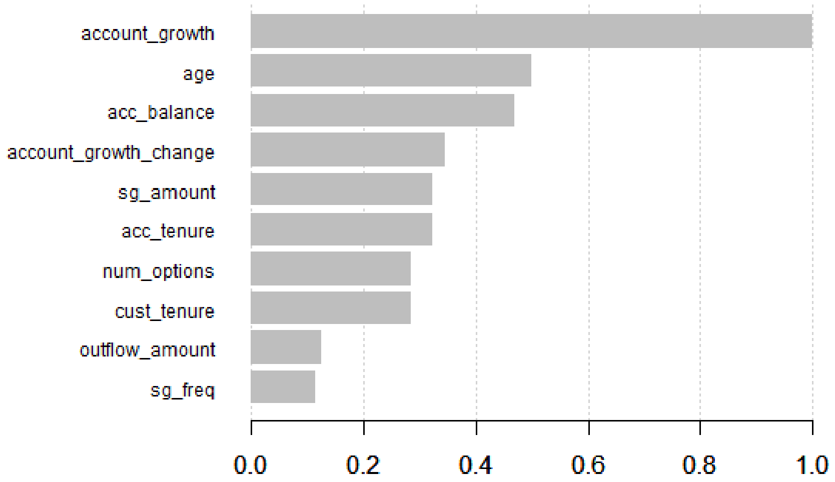


Figure 3.5: Top 10 important feature to financial delegation

first determinant to seek financial delegation, age and account balance are quite influential as well. When it comes to investment return, risk tolerance is much more reflective of customers' need for investment return, compared with other factors.

These figures together show us the factors we should look at when trying to understand customer needs for a financial adviser. Although financial professionals may already know some of those important factors. The advantage

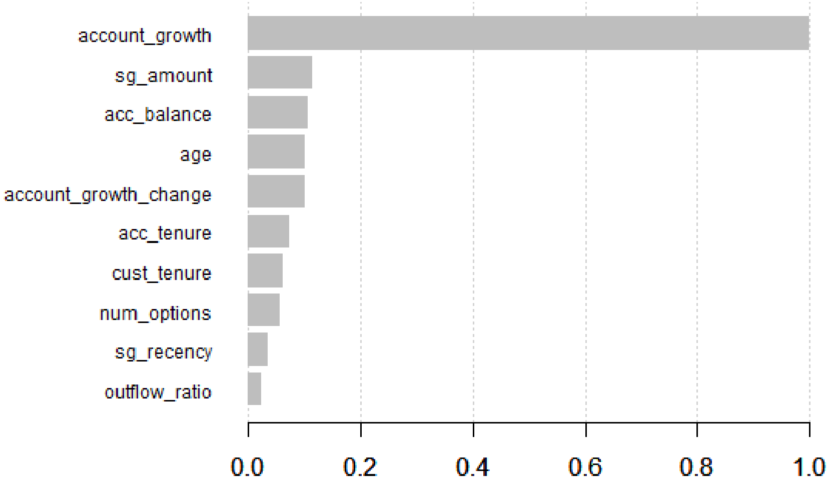


Figure 3.6: Top 10 important feature to investment return

of our method is that those factors are presented quantitatively and could be used to predict the needs of mass customers automatically.

3.5 Conclusions

Understanding customer needs is especially important for fund management and financial planning services. Although many approaches have been adopted in practice, they might not be applicable to learning customer financial needs due to the unique challenges. For example, every individual customer's financial needs must be carefully met, and identifying people's financial needs is more difficult, partially because of a low level of financial literacy. To address those challenges, we propose an automatic end-to-end framework to infer all customers' needs without much manual work being involved in this chapter. Specifically, three possible needs, i.e., retirement preparation, financial delegation, and investment return, are defined base on domain knowledge and inferred from real data such as customer behavior, adviser behavior, and financial outcomes. Based on a well-designed group of features and a selected set of informative customers, a multi-label learning technique is then adopted to build a predictive model that can predict multiple customer needs simultaneously. Finally, this model is used to predict needs for all other customers. To build a reliable labeled data set for building models, several assumptions are made to select customers and extract needs. Arguably, our method can automatically extract true needs for the selected customer to the greatest extent, and the model built by using multi-label learning techniques can predict the needs ranked by relative importance to other customers. Extensive experiments are conducted using real data. The effectiveness of our method is well verified, and insights into influential factors to customer needs are also presented.

Chapter 4

Prediction of stopping engaging financial adviser by bipartite ranking

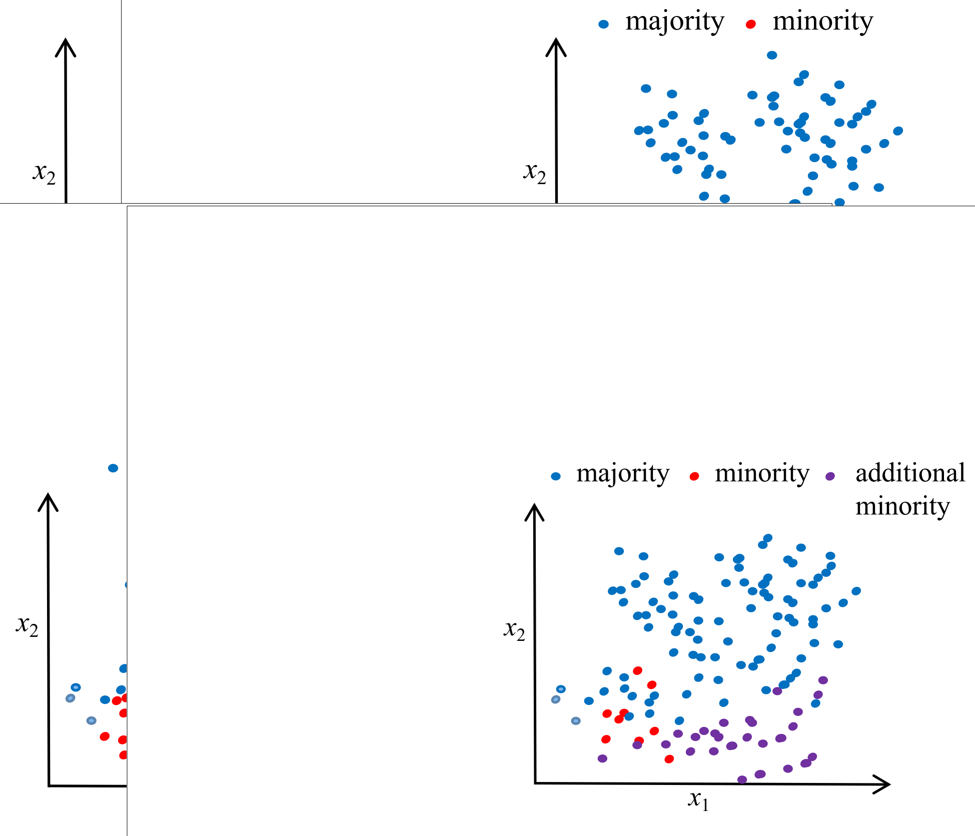
In addition to inferring needs for new customers as in the previous chapter, early detection of customers who are likely to end their relationships with their advisers is also important to financial services. It is a type of customer churn prediction problem for which various models have been proposed. However, two key issues still need further investigation. One is the extremely imbalanced data, i.e., only a few churners. The other is existing methods mostly solve it as a binary classification problem and try to classify all customers correctly, whereas it is more practical to solve it as a bipartite ranking problem that focuses on identifying only those churners. A novel method is proposed in this chapter to deal with these issues. Specifically, a more balanced data is formed by exploiting historical data of churners, and a boosting based transfer learning strategy is employed to help select the most informative data. Moreover, a new ranking-based evaluation measure is used to adjust data weights, guiding the learning process to focus on distinguishing between churners and non-churners.

The remainder of this chapter is organized structured as follows. The introduction of motivation and contributions is presented firstly. Then, preliminaries include related work and a formal statement of the problem are given. Next is the detailed description of our proposed method. After that is the experimental results and analysis, followed by conclusions finally.

4.1 Introduction

Customer acquisition and customer retention are the two fundamental elements of customer relationship management (Reinartz et al. 2005). Customer retention is more important to financial services, due to the cost of losing valuable customers. Therefore, early detection of customers who are likely to stop seeing their advisers is also important. It allows businesses to take action to improve their satisfaction and retain them as much as possible. As a matter of fact, this is a common challenge named *customer churn prediction* facing most industries nowadays, including telecommunication (Huang et al. 2015), online platform (Ngonmang et al. 2012), and finance (Culbert et al. 2018; Brownlow et al. 2018a) etc. Customer churn prediction can naturally be addressed as a classification problem using data mining techniques, and lots of classical methods have already been applied to this problem, including boosting (Lu et al. 2014), random forest (Xie et al. 2009), and neural networks (Ismail et al. 2015), etc.

One key issue facing customer churn prediction is the imbalanced data. It it because there are literally few churners during a period of time. For example, only around 0.5% of customers in a particular fund provider would stop engaging financial advisers in 3 months. Let $D = \{D^+, D^-\}$ be a dataset in which D^+ is a group of churners, and D^- are non-churners, it means $|D^+| \ll |D^-|$. As a result, the model built on D could be biased towards non-churners, thus are not able to identify churners well. To deal with imbalanced data, various techniques have been widely adopted, including under-sampling, over-sampling etc (He and Garcia 2009). The basic idea is to modify data distribution by manipulating data, to make data of minority



(d)

Figure 4.1: (a) Imbalanced data, (b) Under-sampling, (c) Over-sampling, and (d) auxiliary data

Figure 4.1 gives a toy example that explains the sampling methods. Firstly, sub-figure (a) shows the original imbalanced data in which a few red points are surrounded by blue points, making it difficult to discriminate them. Subfigure (b) shows a possible result of down-sampling that excludes blue points that are similar to some red ones, so the data of minority class are more learnable. By contrast, over-sampling copies some of the red points or generates similar ones as shown in subfigure (c), to make data of minority class less likely to be misclassified. Although be straightforward, the issue of these two sampling methods is that they cannot handle imbalanced data that is caused by failing to capture all data of minority class. For example, additional data of minority class as purple points in subfigure (d) could get us a more comprehensive understanding of data of minority class. However, a potential issue of those additional data is that part of them might be of different distribution. For example, churners in the past might have different reasons to recent churners. Another issue is that most existing methods treat customer churn prediction as a binary classification problem, i.e., the aim is to classify both churners and non-churners. In reality, It is more preferable to obtain a ranking of customers in which churners could be ranked as higher as possible, so businesses could focus their market capacity on customers on the top.

To solve above issues, a novel method based on transfer learning is proposed in this chapter. The basic idea is to alleviate the issue of imbalanced data by including historical data of minority class. a group of churners in the past will be included for building models. Since historical data might be of different distributions, we adapt a boosting-based transfer learning process (Dai et al. 2007) to select only those beneficial data. Specifically, the weights of historical data will be adjusted iteratively, and their weights will be increased if can improve the learning of current data. As a result, only those beneficial data would be included as their weighted will be increased constantly. Moreover, a ranking-based classification measure is adopted in our method, trying to learn a model that ranks churners at the top.

To summarize, the main contributions of our method in this chapter include: (1) an instance-based transfer learning strategy is adapted that exploit historical data to deal with imbalanced data; (2) a ranking-based measure is employed, so customer churn prediction is solved as a bipartite ranking problem; and (3) the effectiveness of our method is analyzed by experiments with real-world data.

4.2 Preliminaries

In this section, boosting-based methods for transfer learning are introduced firstly. Then the problem definition is given.

4.2.1 Boosting

Boosting is a ensemble learning technique that build multiple models sequentially. As the basic AdaBoost.M1 method (Freund and Schapire 1996) outlined in Algorithm 4.1, a model is built and used to classify all instances x_i in each round t. Then, the weights of those misclassified instances are increased, thus the model built in next round will focus on those "hard" instances.

```
Algorithm 4.1: AdaBoost.M1
  Input  : Training set: $D = (x_1, y_1), \ldots, (x_n, y_n)$, learning method:
           $h$, Number of rounds: $T$
  Weights initialization: $w_0(i) = 1/n$ for all instances $x_i$;
  for $t \leftarrow 1$ to $T$ do
      Learn a new model: $f_t(x) \rightarrow y$ using $D$ over distribution $p$;
      Calculate the error of $f_t$ : $\epsilon_t = \sum_{i=0}^{n} w_i I(h_t(x_i) \neq y_i)$;
      Set $\beta_t = (1 - \epsilon_t)/\epsilon_t$;
      Update weights: $w_{t+1}(i) = \frac{w_t(i)}{Z_i} \beta_t^{I(h_t(x_i) \neq y_i)}$
  Output: $f(x) = \arg\max_{y \in Y} \sum_{t: f_t(x_i) = y_i} \log \beta_t$
```

It could be seen that the boosting method could handle imbalanced data itself. Since data of minority class are harder to classify than data of majority class, the boosting method actually modifies the data distribution like a combination of down-sampling and over-sampling. Furthermore, many extensions of the boosting method have been proposed to exploit auxiliary data of different distributions (Dai et al. 2007; Yao and Doretto 2010). Given an additional dataset $S(P(S) \neq P(D))$, the *TrAdaBoost* method uses $D \cup S$

to build models in the boosting process. The difference is that the weight of any instance $x_i \in S$ will be increased if it is correctly classified instead of being decreased. The assumption is that those instances should be of the distribution $P(D)$, and are helpful to the classify instances in D. Our method proposed in this chapter is inspired by *TrAdaBoost*, the difference is that our method only concerns with classifying churners correctly. so S in our method consists of only data of minority class, and the classification error β is replaced with a ranking-based measure.

4.2.2 Problem statement

The problem in this chapter is to predict customers who are likely to stop engaging their advisers, i.e., churners. Let $D = \{(x_1, y_1), (x_1, y_2), \ldots, (x_n, y_n)\}$ be a group of current customers, in which $y_i \in \{1, 0\}$ indicates whether x_i is a churner or not. Also let $S = S_1 \cup S_2, \ldots, \cup S_d$ be multiple groups of churners who closed their relationship with advisers during different time in the past, i.e., for any pair $(x_i, y_i) \in S$, we always have $y_i = 1$. Formally, the problem is to learning a mapping function $f_{D \cup S}(x) \to r$ that maps a given x to a numeric value in $[0, 1]$ that represents x's probability of being a churner. Generally, predictions of churners $D^+ = \{(x, y)|(x, y) \in Dy = 1\}$ should be greater than those of non-churners $D^- = D - D^+$, to rank churners before non-churners. Therefore, the model f should maximise the AUC-based measure as shown in Equation (4.1).

$$\frac{\sum_{x_i \in D^+} \sum_{x_j \in D^-} I(f(x_i) - f(x_j))}{|D^+|\,|D^-|} \tag{4.1}$$

where $I(x) = 1$ if $x > 0$, and $I(x) = 0$ otherwise. This measure essentially calculates the average probability of $f(x_i) > f(x_j)$, i.e., x_i is ranked before x_j, given a random pair of $(x_i, x_j) \in D^+ \times D^-$.

4.3 Churner prediction with instance transferring

As stated above, classification is to learn patterns in data D in the form of a function f, which is used to making predictions for instances in data T. Traditionally, it is assumed that D and T have the same distribution, i.e., $P(D) = P(T)$, so the patterns in D also apply to T. In terms of customer churn prediction, it means that we should learn models from *most recent* churners to make predictions. However, the reality is that there usually are a few recent churners, while data of a large number of *historical* churners is available as accumulated over years. Let S be a set of historical churners, it usually has a different distribution, i.e., $P(S) \neq P(T)$, thus not all of them are helpful in making predictions for T.

To exploit only the useful part of S, an instance-based transfer learning strategy proposed in (Dai et al. 2007) is adopted in this chapter. let $E = \{x_i, x_2, \ldots, x_{|D|}, \ldots, x_{|D|+|S|}\}$ be the combined training set $D \cup S$, in which the first $|D|$ instances are from D, and the following $|S|$ instances are from S. This combined dataset E will be fed into the boosting process in which data weights are re-weighted iteratively so that only data useful to improve model performance will be intensified. The detailed description of our proposed method *imTraAdaBoost* is given in 4.2

To begin with, all instances in E are initialized with the same weight. In the t-th iteration, a model f_t is learned using E over its current weights w_t. Similar to basic AdaBoost, model f_t is then evaluated by examining how it performs on the training set. However, It should be noted that only D is used, instead of the whole E, to evaluate performance. This is because the purpose here is to exploit auxiliary data S to aid in learning on target D, so model performance on S should not be considered, otherwise the learning process will be biased. The other key element is the measure used for evaluating predictions. The binary $I(h_t(x_i) \neq y_i)$ used in AdaBoost is not appropriate for imbalanced data, because it is not able to reveal how the model perform

Algorithm 4.2: ImTraAdBoost

```
  Input  : dataset: $E = D \cup S$, learning method: $h$, and the number
           of rounds: $T$
1 Weights initialization: $w^0 = \left\langle w_1^0, \ldots, w_{|D|}^0, \ldots w_{|D|+|S|}^0 \right\rangle$;
2 for $t \leftarrow 1$ to $T$ do
3     Normalize weights $w^t = w^{(t-1)} / (\sum_i^{|D|+|S|} w_i^{(t-1)})$ ;
4     Learn a new model $f_t(x) \rightarrow r$ using $D$ over distribution $w^t$,
       $(0 \leq r \leq 1)$;
5     Make prediction on $E$: $f_t = \langle f_t(x_i), \ldots, f_t(x_{|D|+|S|}) \rangle$;
6     Calculate the error of $f_t$ on $D$:
       $\epsilon_t = \sum_{i=1}^{|D|} \frac{w_i^t}{\sum_{i=1}^{|D|} w_i^t} \cdot \gamma(x_i, f_t)$;
7     abort this round if $\epsilon_t \geq 0.5$;
8     Set $\beta_t = (1 - \epsilon_t)/\epsilon_t$ and $\beta = 1/(1 + \sqrt{2\ln(|S|)/T})$;
9     Update weights: $w_i^{t+1} = \begin{cases} w_i^t \cdot \beta_t^{\gamma(x_i, f_t)} & \text{if } x_i \in D \\ w_i^t \cdot \beta^{\gamma(x_i, f_t)} & \text{if } x_i \in S \end{cases}$
  Output: $f(x) = \sum_{t=1}^{T} Z_t f_t(x)$ where $Z_t = \ln(\beta_t) / \sum_{t=1}^{T} \ln(\beta_t)$
```

on minority class. Therefore, a new ranking-based measure is adopted which is shown in Equation (4.2).

$$\gamma(x_i, f) = \begin{cases} \frac{\sum_{x_j \in D^-} I(f(x_j) - f(x_i))}{|D^-|} & \text{if } x_i \in D^+ \\ \frac{\sum_{x_j \in D^+} I(f(x_i) - f(x_j))}{|D^+|} & \text{if } x_i \in D^- \end{cases} \tag{4.2}$$

Here $I(x) = 1$ if $x > 0$, otherwise $I(x) = 1$. Given an instance x_i, instead of comparing prediction $f(x_i)$ with its true class, this measure examines how well data of minority class are ranked before data of majority class. Given any $x \in D^+$, it calculates the percent of instances of the majority class that are ranked before it. Therefore, it only concerns with between-class ranking, whereas how instances of the same class are ranked dose not matter. $\gamma(x_i, f)$ falls into $[0, 1]$, and $\gamma(x_i, f) = 0$ if $x_i \in D^+$ is ranked before all instances in D^-, and $\gamma(x_i, f) = 1$ if $x_i \in D^+$ is ranked after all instances in D^-. The opposite is true for $x_i \in D^-$. Line 6 in Algorithm 4.2 is the weighted average of this measure over D, it is obviously more appropriate if businesses are interested in identifying only churners.

Next, instances are re-weighted as shown at line 9 in Algorithm 4.2. Instances from D follow the same rule used in Adaboost. Their weights will be increased exponentially with $\gamma(x_i, f_t)$ since $\beta_t > 1$, so next round will focus on those hard-to-rank instances. By contrast, since $\beta < 1$, x_i's weight will be decreased exponentially if $x_i \in S$. A large $\gamma(x_i, f_t)$ for $x_i \in S$ means that it might be from a different distribution. Therefore, it should be made less important in building models in the following rounds. When making predictions for new instances, all the models $(f_1, \ldots, f_T)$ are used, and the final output is a weighted average of all those models' predictions as shown at the last line in Algorithm 4.2. every model f_t is assigned with a weight $\ln(\epsilon_t)$ which expresses its confidence in the prediction.

Above is the detail of our proposed *ImTraAdBoost* method. In summary, it aims to learn a good bipartite ranking of churners and non-churners. To deal with the imbalanced data issue, auxiliary data that might be from different sources are exploited using an instance-based transfer learning technique. As a result, only those auxiliary data, which are estimated to be from the source of original data, are fully utilized.

4.4 Experiments

This section presents the experimental evaluation and comparison of our ImTraAdaBoost method and other advanced methods. To begin with, datasets, evaluation metrics, and experimental settings are introduced. Then, experimental results are analyzed.

4.4.1 Data sets

Seven datasets that consist of real-world data from a fund provider based in Australia are used in our experiments, and the detailed description of these datasets is given in Table 4.1.

Table 4.1: Description of datasets

dataset	date	$\|D\|$	$\|D^+\|$	$\|D^+\|/\|D\|$
D_1	Dec2017	33,124	880	2.66%
D_2	Mar2018	32,587	653	2.00%
D_3	Jun2018	31,838	868	2.73%
D_4	Sep2018	30,898	381	1.23%
D_5	Dec2018	30,153	836	2.77%
D_6	Mar2019	28,964	776	2.68%
D_7	Jun2019	27,155	725	2.67%

Specifically, these datasets are generated by selecting customers observed at seven particular points of time, i.e., the end of Dec 2017, Mar 2018, Jun 2018, Sep 2018, Dec 2018, Mar 2019, and Jun 2019. Within a particular dataset, those customers are labeled as *churner* if they would stop seeing their current advisers in the following three months, otherwise they are labeled as *non-churner*. In table 4.1, $|D|$ is the number of customers, $|D^+|$ is the number of churners, and $|D^+|/|D|$ is the percentage of churners. We can see that all those datasets are extremely balanced data, only less than 3 % of customers would stop seeing their advisers in three months. Since these datasets are generated chronologically, we are able to investigate how historical data could aid in learning from current data. For example, when learning models using dataset D_7, all those churners in preceding six datasets could be included as auxiliary data. Since churners in the past might be of different distributions, our proposed method here tries to identify and focus on only those that could have a positive effect on learning models.

4.4.2 Evaluation metrics

Our methods will output a ranking of customers in terms of their probabilities of being churners in following three months. Since we only concerned with ranking churners at the top, 3 relevant metrics are therefore used to evaluate and compare models, they are *recall*, *precision*, and *AUC*. The definition of *recall* is given in Equation (4.3)

$$R@k = \frac{\sum_{x \in Top(k)} |\{x|x \in D^+\}|}{|D^+|} \tag{4.3}$$

Here $Top(k)$ denotes the top K customers of a ranking. $|D^+|$ is the total number of true churners, and $|\{x|x \in D^+\}|$ is the number of true churners out of top K members. The definition of *precision* is given in Equation (4.4), and it measures how accurate the result is if we only look at the top K customers.

$$P@k = \frac{\sum_{x \in Top(k)} |\{x|x \in D^+\}|}{K} \tag{4.4}$$

It should be noted that increasing K will increase *recall* but decrease *precision*, so an appropriate K should be decided in practice. An AUC-based measure which is defined in Equation (4.1) is also used. Given a random pair of a churner and a non-churner, it essentially estimates the probability of a model being able to rank them correctly.

4.4.3 Result analysis

This subsection analyses the experimental results to validate our assumption about the historical data and validate our method's effectiveness in exploiting those data.

The usefulness of historical data

To begin with, we'd like to validate our assumption that historical data and current data might not be from the exactly same distribution. Therefore, it should be that only part of the historical data are useful to current learning task.

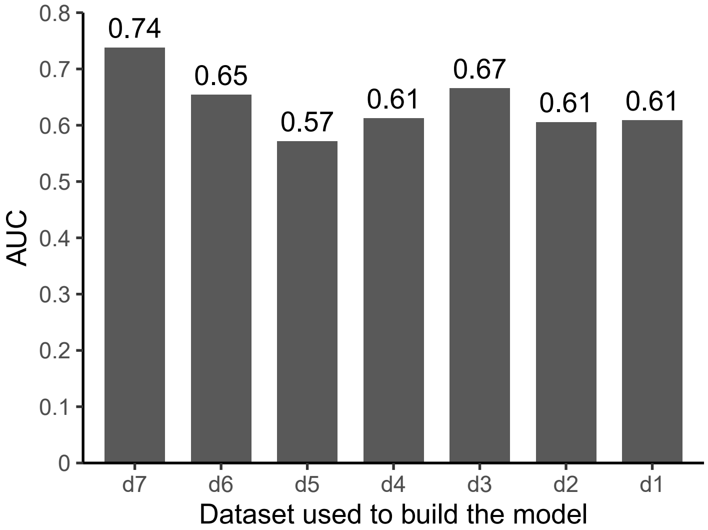
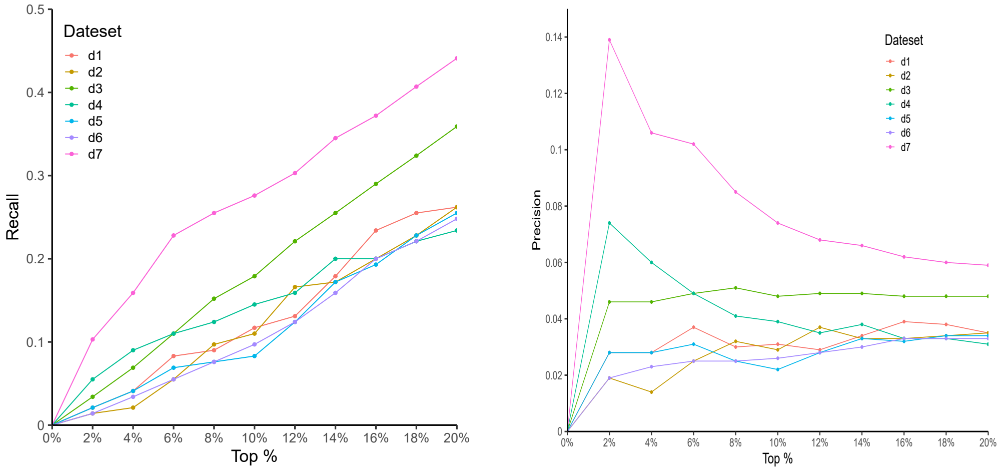


Figure 4.2: Comparison in terms of precision

To this end, we firstly learn multiple models using historical data to investigate their performance on current data. Specifically, the gradient boosting method (Chen and Guestrin 2016) is used to build models using dataset D_{1-6} as shown in Table 4.1 respectively. Then we split dataset D_7 into two parts, i.e., 20% of them are kept as test data, and 80% of them are used to build a model that should perform best on the test data since they are from the same source. All those seven models built on D_{1-6} and 80% of D_7 are then used to make predictions for instances in the test data, i.e., the 20% of D_7. The results in terms of the three metrics are shown in Figure 4.2 and 4.3, which clearly validate our assumption. Firstly, we can see that all models built using historical D_{1-6} have an AUC > 0.5 (better than random guess). It means that those historical data contain useful knowledge to some extent that could be exploited. However, these models perform significantly worse than the model built using D_7 with AUC being decreased by 9.5% - 22.5%. It means that these historical data could not be used directly, since some churners in the past may have different behavioral patterns or reasons to become a churner. Moreover, Figure 4.2 also shows that although recent data, e.g., D_6, is more useful, it does not necessarily mean that older data will be increasingly less useful, as the model using D_3 shows a relatively better result, probably because those data might follow some seasonal patterns. As a result, we cannot simply determine the usefulness of historical data in terms of the point of time they were observed. In conclusion, these three findings together prove that it is necessary to employ a more proper mechanism to identify useful data in the past, and it justifies the method we propose in this chapter.



(a)

(b)

Figure 4.3: (a) Comparison in terms of recall, (b) Comparison in terms of precision

Figure 4.3 shows similar results. The model built using data of the same source is much superior to other models in terms of both recall and precision.

Influence of different numbers of iterations

Since our proposed method adopts an iterative progress to identify useful historical data, we also experiment to examine how it performs with an increasing number of iterations. To this end, we split dataset D_6 into a training set D_t (80%) and a test set D_s (20%) again. Moreover, all churners in dataset D_{1-5} are added into the the training set, i.e., $D_t = D_t \cup D^+_{i(1 \leq i \leq 5)}$. Therefore, the training set includes a group of historical data that might have a negative impact if not being dealt with appropriately. Our proposed *ImTraAdBoost* method is used to build models with different numbers of iterations, and specific results in terms of the three metrics are given in Figure 4.4 and 4.5.

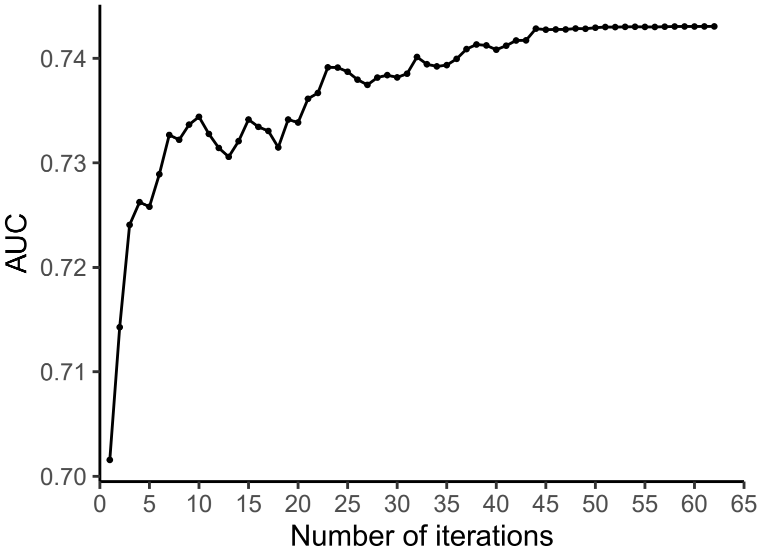
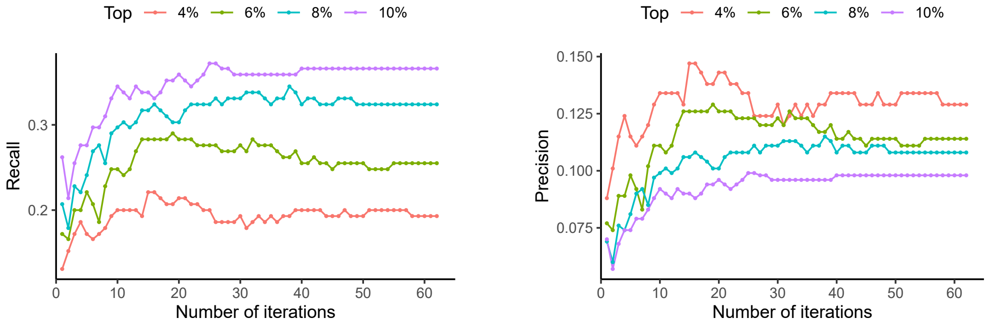


Figure 4.4: Change of AUC with increasing number of iterations



(a)

(b)

Figure 4.5: (a) Change of recall with increasing number of iterations, (b) Change of precision with increasing number of iterations

All these results validate the effectiveness of the method in exploiting data from different sources. As shown in Figure 4.4, AUC increases significantly in the first several iterations and converges to a value that is bigger than the AUC without historical data being added. We could infer that useful historical data might have been fully utilized at that point and further iterations will be not very helpful. Similar results can also be observed in terms of recall and precision as shown in Figure 4.5.

Comparison with other baselines

Furthermore, we also compare our method with other methods on multiple datasets. Given a dataset $D_i (i > 1)$ in Table 4.1, Let's call it a *target* dataset when it is used for building models, and the dataset consists of all churners in its preceding datasets, i.e., $D_1^+, \cup \ldots \cup D_{i-1}^+$ an *auxiliary* dataset. Datasets D_{2-7} are then used to compare the following four methods.

- A Gradient boosting machine (Gbm) (Chen and Guestrin 2016) model is built using only the target dataset, which does not consider the issue of imbalanced data.
- *EasyEnsemble* (Liu et al. 2009). It samples multiple subsets from the majority class, each of them will be combined with all data of minority class to build a Gbm model. This method shows the performance of the down-sampling method without adding into auxiliary data.
- Another Gbm model is built using the combination of the target dataset and auxiliary dataset. It shows how the model will perform if we simply include data of different distributions.
- Our proposed *ImTraAdBoost* model that iteratively re-weights auxiliary data, so only those informative data will play increasingly important roles.

All those models are implemented in R, and the package *xgboost* [1] is used which gives an efficient implementation of the Gbm method. All these six datasets are split into training data (80%) which are used to build models and test data (20%) which are used to evaluate model performance. The detailed results are shown in Table 4.2 - 4.4.

[1] https://github.com/dmlc/xgboost

Table 4.2: Model comparison in terms of AUC

dataset	Gbm	EasyEnsemble	Gbm+auxiliary	ImTraAdBoost
D_7	0.765	0.770	0.744	**0.775**
D_6	0.786	0.807	0.791	**0.812**
D_5	0.764	0.804	0.785	0.775
D_4	0.720	0.759	0.739	0.757
D_3	0.743	0.777	0.753	0.741
D_2	0.761	0.770	0.728	**0.774**

Firstly, Table 4.2 shows how these models perform in terms of AUC. It can be seen that our method *ImTraAdBoost* performs best on three out of the six datasets. Compared with *Gbm+auxiliary* that simply adds into auxiliary data, it shows better performance on four datasets, meaning that auxiliary data for those datasets might be of different distributions, so our method is more proper in those situation, especially on dataset D_7 and D_2 on which *Gbm+auxiliary* performs worse than building models only using the target dataset.

Table 4.3: Model comparison in terms of recall of top 10%

dataset	Gbm	EasyEnsemble	Gbm+auxiliary	ImTraAdBoost
D_7	0.234	0.228	0.221	**0.276**
D_6	0.397	0.404	0.365	**0.417**
D_5	0.345	0.381	0.339	**0.393**
D_4	0.325	0.273	0.299	**0.351**
D_3	0.241	0.259	0.207	0.241
D_2	0.275	0.282	0.191	**0.290**

Besides AUC, we also compare these models in terms of *recall* and *precision* by looking at the top 5% of rankings they output for the test data. From Table 4.3, we can see that our model is more superior to others, and performs best on five out of the six models. On average, our method outperforms *Gbm*, *EasyEnsemble*, and *Gbm+auxiliary* by 8.3%, 7.7%, and 21.3% respectively. Moreover, it is shown that *Gbm+auxiliary* has a worse performance than *Gbm* on all of these datasets, proving the negative effect of simply adding into auxiliary data without careful selection. Table 4.4 shows similar results as well. Our method again shows best performance on five out of six datasets, and its precision of the top 10% is increased by 8.1%, 5.7%, and 20.8%, compared with *Gbm*, *EasyEnsemble*, and *Gbm+auxiliary* respectively.

Table 4.4: Model comparison in terms of precision of top 10%

dataset	Gbm	EasyEnsemble	Gbm+auxiliary	ImTraAdBoost
D_7	0.105	0.102	0.098	**0.123**
D_6	0.179	0.182	0.164	**0.187**
D_5	0.161	0.177	0.158	**0.183**
D_4	0.068	0.057	0.062	**0.073**
D_3	0.110	0.118	0.094	0.110
D_2	0.092	0.095	0.064	**0.097**

In conclusion, the experimental results reveal that data of different distribution is useful to the current learning task, but using them directly will lead to negative results. Therefore, how to identify the part of these data that could aid in improving learning performance is crucial. Our model's effectiveness in this regard is validated as well by the experiments.

4.5 Conclusions

Early detection of customers who are likely to end their relationships with their advisers, i.e., churners, is another key point in satisfying customer needs. To deal with this problem, many existing classification methods have been adopted to address it as a binary classification task. However, it is more suitable to regard it as a bipartite ranking problem that aims to rank churners before non-churners, so businesses can focus their marketing resources on those at the top of a ranking. Moreover, imbalanced data is a key challenge since there are few churners during a period of time. To address these issues, a novel method is proposed in this chapter accordingly. Specifically, a more balanced data is formed by exploiting historical data of churners, and instance-based transfer learning strategy is employed to help select the most informative instances. Moreover, a new ranking-based evaluation measure is designed to adjust weights of auxiliary data, guiding the learning process to focus on ranking churners before non-churners. Experimental results clearly show that our method could effectively exploit data of different sources to improve the current learning task while avoiding the negative effects those data could have brought in.

Chapter 5

Adviser recommendation on heterogeneous graph

After identifying customers who might be in need of financial advisers, it is then critical to satisfy customers by recommending financial advisers that match their needs to them. Although an altitude of recommendation techniques have been proposed and applied in other industries to date, there is little research on financial adviser recommendation. Moreover, existing recommendation methods primarily infer customers' preferences from data like their ratings over a number of items or social relationships with other users. However, customers usually engage only a single adviser in most cases, and their social relationships between them are unknown to traditional financial services. It means those existing methods cannot be applied to adviser recommendation. To deal with these issues, this chapter proposes a new method to exploit side information like customers' demographic and behavior data to identify and connect similar users. As a result, users and advisers can be organized into a heterogeneous graph, thus adviser recommendation can be realized by finding those similar nodes which represent advisers for a given node that represents a customer. To find similar nodes, the random walk with restart (RWR) process is applied to exploit knowledge embedded in the graph structure to rank all nodes in terms of their similarities to a given node, so advisers ranked at the top could be recommended.

The remainder of this chapter is structured as follows. Firstly, the challenges in adviser recommendation and our contributions are introduced. Then, a formal statement of adviser recommendation and relevant foundations are presented. Next is the analysis of the proposed method in detail. Experimental analysis is also done then to verify the method, followed by conclusions finally.

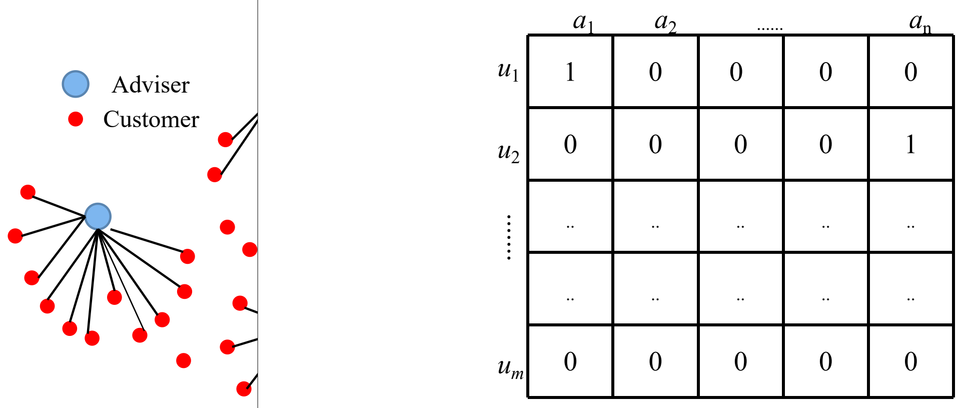
5.1 Introduction

Recommendation system is the technique to estimate customers' preferences and help them find the desirable items. For example, *Amazon* analyzes the books you brought in the past to recommend you the books you might like; *Youtube* recommends you the videos that are similar to videos you have watched. Nowadays, recommendation system is playing increasing roles in helping people deal with overwhelming information and has been applied to recommending various types of items including book (Ekstrand et al. 2018), video (Covington et al. 2016), and hotels Grbovic and Cheng (2018), etc.

Compare with above industries, helping customers identify financial advisers that match their financial statuses and objectives is more significant. Firstly, it is difficult for people to collect and compare the information about a large number of financial advisers in practice. In comparison, there are millions of books and videos available online to choose from. Secondly, engaging a suitable financial adviser is critical to people, especially those with a low level of financial literacy, because of the great influence on people's financial well-being. For the point of view of a financial service, recommending the right advisers is key to improve customer satisfaction as well.

In spite of the extensive applications, existing recommendation techniques, however, could not be used directly for adviser recommendation, because of the unique challenges. One of the biggest challenges is the lack of data from which customers' preferences could be inferred. For example, typ-

ical methods, e.g., collaborative filtering and matrix factorization, rely on a rating matrix shown in Figure 2.6 that encode customers' explicit feedback to
multiple items (Adom
Since a customer is o
matrix we could obtai
most rows have only a
is that it might be al
who are not with any



	a_1	a_2	......		a_n
u_1	1	0	0	0	0
u_2	0	0	0	0	1
⋮	..	..	..	..	..
	..	..	..	..	..
u_m	0	0	0	0	0

(a) (b)

Figure 5.1: (a) a graph of users and advisers, (b) The corresponding matrix

To address above issues, a number of graph-based methods which take advantage of social connections between customers to make recommendation (Ma 2013; Ma et al. 2012). In these methods, a customer's preferences are revealed by others connected to them. However, the challenge is that connections between customers are not available to most financial services as well. By contrast, connections between users and advisers, in reality, could be illustrated by the heterogeneous network shown in 5.1(a). In this network, a customer is connected only to the adviser he/she is with at the moment, and new customers like the red points at the center have no connections to any other nodes. Therefore, it is not possible to make recommendations based on such networks.

Therefore, a novel graph-based is proposed in this chapter to alleviate the issue of disconnected networks. Specifically, data such as customer demographic and behavioral data are exploited to calculate similarities between users/advisers, and a customer is then connected with his/her neighbors. In this way, the network in 5.1(a) is actually augmented so that every customer/adviser is connected to others, and recommendations can be realized by recommending advisers that are close to a customer in terms of the graph structure. To this end, a random walk process is applied to this heterogeneous network, which could output a ranking of advisers for a given customer. Experiments are conducted using real-world data, and the results verify our method's effectiveness in recommending advisers.

5.2 Preliminaries

In this section, the notations used and a formal statement of the problem are introduced firstly. Then related basics and methods are presented.

5.2.1 Problem statement

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be a set of customers, and $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ be a set of advisers. The relationship between customers and advisers can be represented by a matrix $R_{n \times m}$ in which $r_{ij} = 1$ means customer u_i is associated with adviser v_j, and $r_{ij} = 0$ means the opposite. As stated before, a customer could have only a single adviser, so $\sum_{j=1}^{m} r_{ij} = 1$. Usually, customer u_i and adviser v_j can be represented by a feature vector respectively, i.e., $u_i = \langle u_{i1}, u_{i2}, \ldots, u_{ik} \rangle$ and $v_j = \langle v_{j1}, v_{j2}, \ldots, v_{jd} \rangle$. These features, extracted from their demographic and behavioural data, could be *age*, *gender*, and *income* etc. An adviser could have additional features like *Number of customers* and *total assets under management* etc. The problem of adviser recommendation thus could be formally formulated as learning a function f using U, V, and R, which could predict a ranking of all advisers give a particular customer x as shown in Equation (5.1)

$$f_{U,V,R}(x) \rightarrow \langle f_1(x), f_2(x), \ldots f_m(x) \rangle \tag{5.1}$$

Here $f_j(x)$ is the score assigned to adviser v_j, and it is proportional to the probability of v_j being customer x's adviser. Thereby, advisers can be ranked in terms of their scores and those ranked at the top can be returned as the final recommendation.

5.2.2 Random walk-based recommendation

The random walk with restart (RWR) model is initially used in search engines to rank returned pages for a query (Haveliwala 2003). Pages are organized into a graph using the direct links between these pages. RWR is an iterative process that simulates a walker that randomly jumps between nodes via links between them. Specifically, the walker's probabilities of being at those nodes at time $t+1$ are as in Equation 5.2.

$$r^{t+1} = \alpha P \cdot r^t + (1-\alpha)e \tag{5.2}$$

Here e is the probability of jumping back to the starting node, and α is a value in $[0,1]$. r^t is updated iteratively until convergence. Starting from a node i, it is more likely to jump to nodes that have more paths to i, i.e., these nodes are more relevant to i. Therefore, the closeness between node i and other nodes is measured by the probabilities of jumping to those nodes by starting from i. Since being able to rank all other nodes for a given node, RWR has been naturally applied to various recommendation tasks (Gori and Pucci 2007; Jiang et al. 2018).

Inspired by the *itemrank* model (Gori and Pucci 2007), the method proposed in this chapter also runs the RWR model over the graph for recommendation. The difference is that a network consists of only customers are used in the *itemrank* model, while a heterogeneous graph consists of customers and advisers is constructed in our method, and auxiliary data such as user profiles are exploited to form more accurate links between these nodes. The details of our method are given in the next section.

5.3 Recommendation by RWR over graph

To realize the adviser recommendation, a graph-based method is proposed in section. The basic idea is to depict relationships among customers and advisers as a graph, then information embedded in the graph structure is exploited to find advisers close to a given customer. Therefore, this method consists of the two following phases.

5.3.1 Construction of a heterogeneous graph

Unlike other applications like social networks, explicit relationships between users and advisers are not accessible to financial services. Instead, detailed demographic information and historical behaviors related to customers and advisers are well recorded. Therefore, we aim to construct a graph of customers and advisers using such kind of data. Specifically, a graph $G = \langle V, W \rangle$ should be constructed. $V = \{u_{1 \ldots n}, v_{1 \ldots m}\}$ is a set of nodes consists of all users and advisers, and $W_{(n+m)\times(n+m)}$ is a matrix of size $(n+m)\times(n+m)$ which records how nodes are connected.

The connection between user u and adviser v can be easily determined, i.e., $w_{uv} = 1$ if v is u's current adviser, otherwise $w_{uv} = 0$. To determine the set of users a given user u should be connected to, his/her k nearest neighbors $\mathcal{N}(u)$ are firstly identified in terms of Euclidean distances between them. Then u is connected to those neighbors, so corresponding part of matrix W are defined as in Equation (5.3).

$$w_{uk(1 \leq u \leq m, 1 \leq k \leq m)} = \begin{cases} 1 & \text{if } k \in \mathcal{N}(u) \\ 0 & \text{if } k \notin \mathcal{N}(u) \end{cases} \tag{5.3}$$

A diversity of features such as demographic, behavioural and transnational features are exploited to learn similarity between customers. Those features, including *gender*, *age*, and *account balance*, are already available to traditional financial institutions. In this way, the sparse data issue in Figure 5.1 can be effectively solved and a more dense matrix is generated by connecting similar users. Therefore, users' preferences can be learnt by aggregating preferences of their neighbours. The corresponding network is shown in 5.2. Connections between advisers can be determined by the same way as well, i.e., any adviser v should be connected to his/her neighbors as

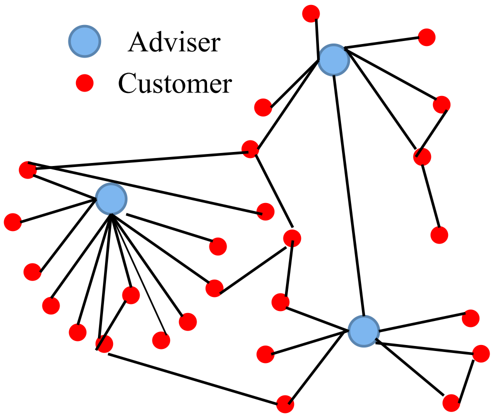


Figure 5.2: An augmented graph of users and advisers

In this way, we could get an augmented graph as shown in Figure 5.2. In this graph, any node is somehow connected to other nodes, making it is possible to find similar nodes by graph traversal.

5.3.2 RWR-based recommendation

The next step is to apply the RWR model on the graph constructed above to making recommendations. To score other nodes as their closeness to a given customer node u, the underlying idea of RWR is that node v's scores $p_u(v)$ should be determined by the scores of nodes which connect to it. Taking all nodes into consideration, the calculation is defined as shown in Equation (5.5)

$$p_u = \alpha \cdot T \cdot p_u + (1 - \alpha) \cdot d_u \tag{5.5}$$

Here $p_u = \langle p_u(u_1), \ldots, p_u(u_m), p_u(v_1), \ldots, p_u(v_n) \rangle$ is a vector of scores to all nodes, indicating their closeness to node u. T is the transition matrix in which t_{ij} is defined as

$$t_{ij} = w_{ij} / \sum_{k=1}^{k_{kj}} \tag{5.6}$$

In other words, T is obtained by normalizing each column of W. d_u is a vector of length $n + m$ in which only the entry corresponding node u is set to 1, all other entries are 0. Generally, d_u could be viewed as an initial guess of the scores which are spread across the graph through an iterative process as defined in Equation (5.7)

$$\begin{cases} p_u^0 = \frac{1}{(n+m)} \cdot \mathbf{1}_{|n+m|} \\ p_u^{t+1} = \alpha \cdot T \cdot p_u^t + (1 - \alpha) \cdot d_u \end{cases} \tag{5.7}$$

This process is repeated until p_u converges to a stable value or reaches a predefined number of iterations. Then, all advisers can be ranked by their scores, and those ranked at the top are returned as a recommendation to the particular customer u. It is should be noted that any adviser will has a chance to be recommended to a given customer u if their have customers similar to u. Furthermore, senior advisers with more customers are not necessarily to be recommended at the top, if they do not have many customers similar to a given customer. Therefore, the cold-starting issue in recommendation is avoid to some extent in our method.

Above is the description of our method which is called *adviserRank*, and it can be seen that every adviser's score is influenced by all other customers and advisers via the propagation of values across the graph. Therefore, it could further take advantage of influences among customers and users, compared with other methods such as collaborative filtering.

5.4 Experiments

Experiments are conducted to verify the effectiveness of the proposed method in section. Datasets and evaluation metric are introduced firstly, then the results are presented.

5.4.1 Data sets and evaluation metric

Real world data from a fund service based in Australia are used in the experiments. Specifically, four datasets that consists of customers and advisers from four different states, i.e., NSW, VIC, QLD and WA respectively are created. Only those customers who keep paying fees to their advisers are selected, to make sure they are somewhat satisfied with the adviser. Table 5.1 shows the details of those datasets, in which $|U|$ is the number of customers, and $|A|$ is number of advisers. Customer profiles and transactional data are also explored to create more than 100 features for customers.

Table 5.1: Description of datasets

dataset	location	$\|U\|$	$\|A\|$
D_1	NSW	10,316	219
D_2	VIC	5,765	207
D_3	QLD	4,363	231
D_4	WA	4,828	167

Since a customer only has a single adviser, a particular metric is designed to evaluation model performance. Given a customer x, let v_x be his/her true adviser, and $top10(x)$ is the top 10 advisers recommended to x by a model, this new metric *topError* is defined as Equation (5.8)

$$topError = \frac{\sum_{x \in D} |\{x | v_x \notin top10(x)\}|}{|D|} \tag{5.8}$$

Therefore, this metric calculates the percentage of customers whose true advisers are not in the recommended top 10 customers in a dataset.

5.4.2 Results analysis

our proposed method is compared with the two classical methods. i.e.,

Table 5.2: Model performance in terms of top 10 recommended advisers

dataset	CF	MF	*adviserRank*
D_1	0.293	0.234	0.195
D_2	0.324	0.247	0.227
D_3	0.241	0.246	0.203
D_4	0.290	0.263	0.225

- user-based CF. this method uses the Equation (5.3) to find similar customers, and return the advisers those customers have.
- matrix factorization. this method is applied with the matrix W defined in the last section and outputs scores as well for pairs of a customer and an adviser.

Each dataset is split into a training set (80%) and a test set (20%). The three methods are then used to build models on the training set, and their performance on the test set are presented in table 5.2. From table 5.2, it can be seen that our proposed method performs best on all datasets. Therefore, it prove that the augmenting the graph of customers and advisers by connecting similar users/advisers is effective in understanding customer preferences

5.5 Conclusions

Recommending the right financial advisers to customers is also essential to meet customer needs. However, there is little research on adviser recommendation. Therefore, we try to introduce a straightforward approach for recommendation to deal with this problem. Two unique challenges in adviser recommendation are (1) a lack of data about interaction between customers and advisers. So we don't have customers' explicit preferences over advisers; (2) new starter issue. it is because it is allowed to recommend advisers only to new customers who are not with any adviser at the moment according to policies in practice, while we know even less about them. To deal with these issues, this chapter proposes a new method to exploit side information like customers' demographic and behavior data to identify and connect similar users. As a result, users and advisers can be organized into a heterogeneous graph, thus adviser recommendation can be realized by finding those similar nodes through a random walk with restart process. Experiments are also conducted, and the results show that the adopted method performs better than other classical methods for recommendation.

Chapter 6

Conclusions and future work

In this chapter, conclusions of the whole thesis and plan for future work are presented.

6.1 Conclusions

Understanding and meeting customers' needs for financial advisers is vital to financial business. General methods for inferring customer needs are not applicable because of the growing amount of data and unique challenges in learning customer financial needs. Therefore, this thesis focus on applying data mining techniques to address several key issues in understanding and meeting customer needs for financial advisers. These issues are (1) inferring potential needs of customers who have not with any financial adviser (2) early detection of customers who are likely to stop seeing their current advisers, and (3) recommending advisers to those are in need of advisers. To address these issues, contributions are made as below.

(1) To infer customers' needs for financial advisers, an automatic data mining process is implemented and a particular multi-label learning technique is applied to predict customers' multiple needs simultaneously. The advantage of this method is that no manual work is required, and customer needs are learning from a combination of different data sources including

customer profiles, transaction records, and account status, etc. It gives a comprehensive understanding of customers. The experimental results show this framework is effective in learning customer needs at scale, and also reveal the most important factors like *age*, *income*, etc, that motivate customers to seek financial advice.

(2) To detect customers who would stop engaging their current advisers, i.e., churners, a novel method is proposed that can rank churners before non-churners. Specifically, an instance-based transfer learning strategy is adopted to exploit historical data to alleviate imbalanced data. Also, a ranking-based measure is designed to make sure the learning process is towards distinguishing churners and non-churners. Experimental results show that historical data is helpful to the current learning process, but they cannot be directly used since part of them are from different sources. Our proposed method, however, could identify the useful part of historical data through an iterative process in which data weights are adjusted according to their usefulness in improving model performance.

(3) To recommend suitable financial advisers to customers, a graph-based method is proposed in this thesis. customers and advisers are organized into a network, and adviser recommendation is realized by finding similar nodes in the graph. To construct a graph that could show accurate relationships between customers, demographic and behavioral data are used to calculate their similarities. RWR process is then run on the graph to identify similar nodes. Experimental results show that this method could better identify preferable advisers for a particular customer.

Deployment and application of theses models could bring significant benefits to customers and the businesses. To customers, satisfying their financial needs could get them better chance to achieve their retirement goals. For example, the model in chapter 3 could help identify customers' potential needs to enable the businesses and financial advisers provide tailored services. In return, improving customer satisfaction could help financial services and advisers better retain the high-value customers. In practice, customer churn could cause a loss of millions of dollars especially for financial institutions. In particular, our model in chapter 4 could enable businesses to early detect and retain those customers who are not satisfied, which is at the core of modern customer relationship management.

6.2 Future work

Above research outcomes provide a good foundation for future investigation into the issues in customer needs learning. In the future, the following directions could be followed.

(1) Applying more advanced methods such as deep learning based methods to adviser recommendation. The research in this thesis is an initial investigation into this problem, so a relatively straightforward method is adopted. In the future, more sophisticated models will be investigated.

(2) The interpretability of models. Experimental results in this thesis verify that the proposed models are effective in inferring customer needs, early detection of churners, and adviser recommendation. However, little insight can be revealed such as customers' motivations to stop seeing advisers and why they have such needs, etc. These insights are more interesting to businesses most times. In the future, how to explain model outputs is another focus.

Bibliography

Adomavicius, G. & Tuzhilin, A., 2005, 'Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions', *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749.

Ahmed, C., ElKorany, A. & Bahgat, R., 2016, 'A supervised learning approach to link prediction in twitter', *Social Network Analysis and Mining*, vol. 6, no. 1, p. 24.

Al-Stouhi, S. & Reddy, C. K., 2016, 'Transfer learning for class imbalance problems with inadequate data', *Knowledge and information systems*, vol. 48, no. 1, pp. 201–228.

Alam, S., Sonbhadra, S. K., Agarwal, S. & Nagabhushan, P., 2020, 'One-class support vector classifiers: A survey', *Knowledge Based System*, vol. 196, p. 105754.

Bae, S. M., Ha, S. H. & Park, S. C., 2005, 'A web-based system for analyzing the voices of call center customers in the service industry', *Expert Systems with Applications*, vol. 28, no. 1, pp. 29–41.

Baldi, P. & Pollastri, G., 2002, 'A machine learning strategy for protein analysis', *IEEE Intelligent Systems*, vol. 17, no. 2, pp. 28–35.

Banerjee, S., Akkaya, C., Perez-Sorrosal, F. & Tsioutsiouliklis, K., 2019, 'Hierarchical transfer learning for multi-label text classification', *Proceedings* *of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6295–6300.

Bengio, Y., Courville, A. & Vincent, P., 2013, ‘Representation learning: a review and new perspectives’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828.

Bielza, C., Li, G. & Larranaga, P., 2011, ‘Multi-dimensional classification with bayesian networks’, *International Journal of Approximate Reasoning*, vol. 52, no. 6, pp. 705–727.

Bishop, C. M., 2006, *Pattern recognition and machine learning*, Springer.

Blei, D. M., Ng, A. Y. & Jordan, M. I., 2003, ‘Latent dirichlet allocation’, *the Journal of machine Learning research*, vol. 3, pp. 993–1022.

Boukerche, A., Zheng, L. & Alfandi, O., 2020, ‘Outlier detection: Methods, models, and classification’, *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–37.

Boutell, M. R., Luo, J., Shen, X. & Brown, C. M., 2004, ‘Learning multi-label scene classification’, *Pattern recognition*, vol. 37, no. 9, pp. 1757–1771.

Breiman, L., 1996, ‘Bagging predictors’, *Machine Learning*, vol. 24, no. 2, pp. 123–140.

Breiman, L., 2001, ‘Random forests’, *Machine Learning*, vol. 45, no. 1, pp. 5–32.

Brownlow, J., Chu, C., Fu, B., Xu, G., Culbert, B. & Meng, Q., 2018a, ‘Cost-sensitive churn prediction in fund management services’, *In proceedings of the 23rd International Conference on Database Systems for Advanced Applications*, Springer, pp. 776–788.

Brownlow, J., Chu, C., Xu, G., Culbert, B., Fu, B. & Meng, Q., 2018b, ‘A multiple source based transfer learning framework for marketing campaigns', *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp. 1–8.

Burkhardt, S. & Kramer, S., 2019, 'A survey of multi-label topic models', *ACM SIGKDD Explorations Newsletter*, vol. 21, no. 2, pp. 61–79.

Cabral, R., De la Torre, F. et al., 2015, 'Matrix completion for weakly-supervised multi-label image classification', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 121–135.

Calcagno, R. & Monticone, C., 2015, 'Financial literacy and the demand for financial advice', *Journal of Banking and Finance*, vol. 50, pp. 363–380.

Cevikalp, H., Benligiray, B. & Gerek, O. N., 2020, 'Semi-supervised robust deep neural networks for multi-label image classification', *Pattern Recognition*, vol. 100, p. 107164.

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A. & Gruber, R. E., 2008, 'Bigtable: a distributed storage system for structured data', *ACM Transactions on Computer Systems*, vol. 26, no. 2, pp. 1–26.

Chen, C., Liaw, A. & Breiman, L., 2004, 'Using random forest to learn imbalanced data', Tech. rep., University of California, Berkeley.

Chen, T. & Guestrin, C., 2016, 'Xgboost: A scalable tree boosting system', *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 785–794.

Chen, W., Yan, J., Zhang, B., Chen, Z. & Yang, Q., 2007, 'Document transformation for multi-label feature selection in text categorization', *Proceedings of the 7th IEEE International Conference on Data Mining*, Omaha, NE, pp. 451–456.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M. et al., 2016, 'Wide & deep learning for recommender systems', *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 7–10.

Cheng, W. & Hullermeier, E., 2009, 'Combining instance-based learning and logistic regression for multilabel classification', *Machine Learning*, vol. 76, no. 2-3, pp. 211–225.

Cherman, E. A., Papanikolaou, Y., Tsoumakas, G. & Monard, M. C., 2019, 'Multi-label active learning: key issues and a novel query strategy', *Evolving Systems*, vol. 10, no. 1, pp. 63–78.

Cici, G., Kempf, A. & Sorhage, C., 2017, 'Do financial advisors provide tangible benefits for investors? evidence from tax-motivated mutual fund flows', *Review of Finance*, vol. 21, no. 2, pp. 637–665.

Clare, A. & King, R. D., 2001, 'Knowledge discovery in multi-label phenotype data', *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2001)*, Freiburg, Germany, pp. 42–53.

Connolly, J.-F., Granger, E. & Sabourin, R., 2012, 'An adaptive classification system for video-based face recognition', *Information Sciences*, vol. 192, pp. 50–70.

Covington, P., Adams, J. & Sargin, E., 2016, 'Deep neural networks for youtube recommendations', *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 191–198.

Culbert, B., Fu, B., Brownlow, J., Chu, C., Meng, Q. & Xu, G., 2018, 'Customer churn prediction in superannuation: a sequential pattern mining approach', *Australasian Database Conference*, Springer, pp. 123–134.

Dai, W., Yang, Q., Xue, G. & Yu, Y., 2007, 'Boosting for transfer learning', *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pp. 193–200.

De Comite, F., Gilleron, R. & Tommasi, M., 2003, ‘Learning multi-label alternating decision trees from texts and data’, *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition*, Springer, pp. 35–49.

de Oliveira, L., Andreao, R. & Sarcinelli-Filho, M., 2010, ‘The use of bayesian networks for heart beat classification’, *Advances in Experimental Medicine and Biology*, vol. 657, no. 3, pp. 217–231.

Dean, J. & Ghemawat, S., 2008, ‘Mapreduce: simplified data processing on large clusters’, *Communications of the ACM*, vol. 51, no. 1, pp. 107–113.

Dembczynski, K., Cheng, W. & Hullermeier, E., 2010a, ‘Bayes optimal multilabel classification via probabilistic classifier chains’, *Proceedings of the 27th International Conference on Machine Learning (ICML2010)*, Haifa, Israel, pp. 279–286.

Dembczynski, K., Waegeman, W., Cheng, W. & Hullermeier, E., 2010b, ‘On label dependence in multi-label classification’, *Proceedings of the 2nd International Workshop on Learning From Multi-label Data*, Haifa, Israel, pp. 5–12.

Dembczynski, K., Waegeman, W., Cheng, W. & Hullermeier, E., 2012, ‘On label dependence and loss minimization in multi-label classification’, *Machine Learning*, vol. 88, no. 1-2, pp. 5–45.

Deshpande, M. & Karypis, G., 2004, ‘Item-based top-n recommendation algorithms’, *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177.

Domingos, P., 1999, ‘Metacost: A general method for making classifiers cost-sensitive’, *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 155–164.

Domingos, P., 2012, ‘A few useful things to know about machine learning’, *Communications of the ACM*, vol. 55, no. 10, pp. 78–87.

Dror, G., Koenigstein, N. & Koren, Y., 2011, ‘Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy’, *Proceedings of the fifth ACM conference on Recommender systems*, ACM, pp. 165–172.

Du, M., Liu, N. & Hu, X., 2019, ‘Techniques for interpretable machine learning’, *Communications of the ACM*, vol. 63, no. 1, pp. 68–77.

Duda, R. O., Hart, P. E. & Stork, D. G., 2012, *Pattern classification*, John Wiley & Sons.

Eckstein, L., Kuehl, N. & Satzger, G., 2016, ‘Towards extracting customer needs from incident tickets in it services’, *Proceedings of 18th IEEE Conference on Business Informatics (CBI)*, , vol. 1pp. 200–207.

Edvardsson, B., Kristensson, P., Magnusson, P. & Sundström, E., 2012, ‘Customer integration within service development—a review of methods and an analysis of insitu and exsitu contributions’, *Technovation*, vol. 32, no. 7-8, pp. 419–429.

Ekstrand, M. D., Riedl, J. T. & Konstan, J. A., 2010, ‘Collaborative filtering recommender systems’, *Foundations and Trends in Human-Computer Interaction*, vol. 4, no. 2, pp. 81–173.

Ekstrand, M. D., Tian, M., Kazi, M. R. I., Mehrpouyan, H. & Kluver, D., 2018, ‘Exploring author gender in book rating and recommendation’, *Proceedings of the 12th ACM conference on recommender systems*, pp. 242–250.

Elisseeff, A. & Weston, J., 2001, ‘A kernel method for multi-labelled classification’, *Proceedings of Advances in Neural Information Processing Systems 14*, Vancouver, British Columbia, Canada, pp. 681–687.

Elkan, C., 2001, ‘The foundations of cost-sensitive learning’, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, Washington, USA, pp. 973–978.

Ertekin, S., Huang, J. & Giles, C. L., 2007, 'Active learning for class imbalance problem', *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 823–824.

Feng, J., Yu, Y. & Zhou, Z.-H., 2018, 'Multi-layered gradient boosting decision trees', *Advances in neural information processing systems*, pp. 3551–3561.

Feng, S. & Xu, D., 2010, 'Transductive multi-instance multi-label learning algorithm with application to automatic image annotation', *Expert Systems with Applications*, vol. 37, no. 1, pp. 661–670.

Fernández, A., Garcia, S., Herrera, F. & Chawla, N. V., 2018, 'Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary', *Journal of artificial intelligence research*, vol. 61, pp. 863–905.

Finke, M., 2013, 'Financial advice: does it make a difference?', Oxford University Press Oxford, UK, pp. 229–48.

Foerster, S., Linnainmaa, J., Melzer, B. & Previtero, A., 2014, 'The costs and benefits of financial advice', .

Freund, Y., Iyer, R., Schapire, R. E. & Singer, Y., 2003, 'An efficient boosting algorithm for combining preferences', *Journal of machine learning research*, vol. 4, no. Nov, pp. 933–969.

Freund, Y. & Schapire, R. E., 1996, 'Experiments with a new boosting algorithm', *Proceedings of the Thirteenth International Conference on Machine Learning*, Bari, Italy, pp. 148–156.

Friedman, J. H. et al., 2001, 'Greedy function approximation: A gradient boosting machine.', *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232.

Fu, B., Wang, Z., Pan, R., Xu, G. & Dolog, P., 2012, 'Learning tree structure of label dependency for multi-label learning', *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 159–170.

Fu, B., Wang, Z., Xu, G. & Cao, L., 2014, 'Multi-label learning based on iterative label propagation over graph', *Pattern Recognition Letters*, vol. 42, pp. 85–90.

Fu, Y., Zhu, X. & Li, B., 2013, 'A survey on instance selection for active learning', *Knowledge and information systems*, vol. 35, no. 2, pp. 249–283.

Gaag, L. C. V. D. & de Waal, P. R., 2006, 'Multi-dimensional bayesian network classifiers', Studený, M. & Vomlel, J. (eds.) *Proceedings of the Third European Workshop on Probabilistic Graphical Models*, Prague, Czech Republic, pp. 107–114.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H. & Herrera, F., 2011, 'A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484.

Galar, M., Fernández, A., Barrenechea, E. & Herrera, F., 2013, 'Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling', *Pattern Recognition*, vol. 46, no. 12, pp. 3460–3471.

Gao, H., Tang, J., Hu, X. & Liu, H., 2013, 'Exploring temporal effects for location recommendation on location-based social networks', *Proceedings of the 7th ACM Conference on Recommender Systems*, ACM, pp. 93–100.

Garcia, S., Luengo, J., Sáez, J. A., Lopez, V. & Herrera, F., 2012, 'A survey of discretization techniques: taxonomy and empirical analysis in supervised learning', *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 734–750.

Goodfellow, I., Bengio, Y. & Courville, A., 2016, *Deep learning*, MIT press.

Gori, M. & Pucci, A., 2007, 'Itemrank: a random-walk based scoring algorithm for recommender engines', *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 2766–2771.

Graves, A., 2012, 'Supervised sequence labelling', *Supervised sequence labelling with recurrent neural networks*, Springer, pp. 5–13.

Grbovic, M. & Cheng, H., 2018, 'Real-time personalization using embeddings for search ranking at airbnb', *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 311–320.

Guo, H., Tang, R., Ye, Y., Li, Z. & He, X., 2017, 'Deepfm: a factorization-machine based neural network for ctr prediction', *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1725–1731.

Guo, Y. & Gu, S., 2011, 'Multi-label classification using conditional dependency networks', *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI2011)*, pp. 1300–1305.

Guyon, I. & Elisseeff, A., 2003, 'An introduction to variable and feature selection', *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182.

Han, J., Cheng, H., Xin, D. & Yan, X., 2007, 'Frequent pattern mining: current status and future directions', *Data mining and knowledge discovery*, vol. 15, no. 1, pp. 55–86.

Han, J., Kamber, M. & Pei, J., 2012, *Data mining concepts and techniques*, USA: Morgan Kaufmann.

Han, Y., Wu, F., Zhuang, Y. & He, X., 2010, 'Multi-label transfer learning with sparse representation', *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 8, pp. 1110–1121.

Haveliwala, T. H., 2003, 'Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search', *IEEE transactions on knowledge and data engineering*, vol. 15, no. 4, pp. 784–796.

He, H. & Garcia, E. A., 2009, 'Learning from imbalanced data', *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284.

He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S. et al., 2014, 'Practical lessons from predicting clicks on ads at facebook', *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, ACM, pp. 1–9.

Hemmatian, F. & Sohrabi, M. K., 2019, 'A survey on classification techniques for opinion mining and sentiment analysis', *Artificial Intelligence Review*, vol. 52, pp. 1495–1545.

Hochreiter, S. & Schmidhuber, J., 1997, 'Long short-term memory', *Neural computation*, vol. 9, no. 8, pp. 1735–1780.

Hofmann, T., 2001, 'Unsupervised learning by probabilistic latent semantic analysis', *Machine Learning*, vol. 42, no. 1-2, pp. 177–196.

Hou, S., Zhou, S., Chen, L., Feng, Y. & Awudu, K., 2016, 'Multi-label learning with label relevance in advertising video', *Neurocomputing*, vol. 171, pp. 932–948.

Hssina, B., Merbouha, A., Ezzikouri, H. & Erritali, M., 2014, 'A comparative study of decision tree id3 and c4. 5', *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 2, pp. 13–19.

Hu, M. & Liu, B., 2004, 'Mining and summarizing customer reviews', *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177.

Hu, Y., Koren, Y. & Volinsky, C., 2008, 'Collaborative filtering for implicit feedback datasets', *2008 Eighth IEEE International Conference on Data Mining*, Ieee, pp. 263–272.

Huang, Y., Wang, W., Wang, L. & Tan, T., 2013, 'Multi-task deep neural network for multi-label learning', *2013 IEEE International Conference on Image Processing*, IEEE, pp. 2897–2900.

Huang, Y., Zhu, F., Yuan, M. et al., 2015, 'Telco churn prediction with big data', *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 607–618.

Hullermeier, E., Furnkranz, J., Cheng, W. & Brinker, K., 2008, 'Label ranking by learning pairwise preferences', *Artificial Intelligence*, vol. 172, no. 16, pp. 1897–1916.

Ismail, M. R., Awang, M. K., Rahman, M. N. A. & Makhtar, M., 2015, 'A multi-layer perceptron approach for customer churn prediction', *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 7, pp. 213–222.

Jiang, J. Q. & McQuay, L. J., 2012, 'Predicting protein function by multi-label correlated semi-supervised learning', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1059–1069.

Jiang, Y. & Yu, S., 2008, 'Mining e-commerce data to analyze the target customer behavior', *First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008)*, IEEE, pp. 406–409.

Jiang, Z., Liu, H., Fu, B., Wu, Z. & Zhang, T., 2018, 'Recommendation in heterogeneous information networks based on generalized random walk model and bayesian personalized ranking', *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 288–296.

Joo, S.-H. & Grable, J. E., 2001, ‘Factors associated with seeking and using professional retirement-planning help’, *Family and consumer sciences research journal*, vol. 30, no. 1, pp. 37–63.

Jun, X., Lu, Y., Lei, Z. & Guolun, D., 2019, ‘Conditional entropy based classifier chains for multi-label classification’, *Neurocomputing*, vol. 335, pp. 185–194.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. & Liu, T.-Y., 2017, ‘Lightgbm: A highly efficient gradient boosting decision tree’, *Advances in neural information processing systems*, pp. 3146–3154.

Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A. & Togneri, R., 2017, ‘Cost-sensitive learning of deep feature representations from imbalanced data’, *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3573–3587.

Kim, K. T., Pak, T.-Y., Shin, S. H. & Hanna, S. D., 2018, ‘The relationship between financial planner use and holding a retirement saving goal: A propensity score matching analysis’, *Financial Planning Review*, vol. 1, no. 1-2, p. e1008.

Koh, H. C., Tan, G. et al., 2011, ‘Data mining applications in healthcare’, *Journal of Healthcare Information Management*, vol. 19, no. 2, p. 65.

Kong, X., Ng, M. K. & Zhou, Z.-H., 2013, ‘Transductive multilabel learning via label set propagation’, *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 704–719.

Koren, R., Y. Bell & Volinsky, C., 2009, ‘Matrix factorization techniques for recommender systems’, *Computer*, vol. 8, no. 42, pp. 30–37.

Kramer, M. M., 2016, ‘Financial literacy, confidence and financial advice seeking’, *Journal of Economic Behavior & Organization*, vol. 131, pp. 198–217.

Krizhevsky, A., Sutskever, I. & Hinton, G. E., 2017, ‘Imagenet classification with deep convolutional neural networks’, *Commun. ACM*, vol. 60, no. 6, pp. 84–90.

Kuehl, N., Scheurenbrand, J. & Satzger, G., 2016, ‘Needmining: identifying micro blog data containing customer needs’, *Proceedinds of the 24th European Conference on Information Systems*, .

Kühl, N., Mühlthaler, M. & Goutier, M., 2019, ‘Supporting customer-oriented marketing with artificial intelligence: automatically quantifying customer needs from social media’, *Electronic Markets*, pp. 1–17.

Lee, C., Fang, W., Yeh, C. & Wang, Y. F., 2018, ‘Multi-label zero-shot learning with structured knowledge graphs’, *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1576–1585.

Lei, S. & Yao, R., 2016, ‘Use of financial planners and portfolio performance’, *Journal of Financial Counseling and Planning*, vol. 27, no. 1, pp. 92–108.

Li, F., Xu, G. & Cao, L., 2015a, ‘Coupled matrix factorization within non-iid context’, *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 707–719.

Li, S., Ju, S., Zhou, G. & Lin, X., 2012, ‘Active learning for imbalanced sentiment classification’, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 139–148.

Li, S., Wang, Z., Zhou, G. & Lee, S. Y. M., 2011, ‘Semi-supervised learning for imbalanced sentiment classification’, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 1826–1831.

Li, X., Xu, G., Chen, E. & Li, L., 2015b, ‘Learning user preferences across multiple aspects for merchant recommendation’, *2015 IEEE International Conference on Data Mining*, IEEE, pp. 865–870.

Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X. & Sun, G., 2018, 'Xdeepfm: combining explicit and implicit feature interactions for recommender systems', *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 1754–1763.

Lin, H.-T., 2019, 'Advances in cost-sensitive multiclass and multilabel classification', *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3187–3188.

Lin, W.-C., Tsai, C.-F., Hu, Y.-H. & Jhang, J.-S., 2017, 'Clustering-based undersampling in class-imbalanced data', *Information Sciences*, vol. 409, pp. 17–26.

Liu, B., 2012, 'Sentiment analysis and opinion mining', *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167.

Liu, B., 2013, *Web Data Mining*, Berlin: Springer.

Liu, H. & Motoda, H., 2013, *Instance selection and construction for data mining*, vol. 608, Springer Science & Business Media.

Liu, X.-Y., Wu, J. & Zhou, Z.-H., 2009, 'Exploratory undersampling for class-imbalance learning', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550.

Liu, X.-Y. & Zhou, Z.-H., 2013, 'Ensemble methods for class imbalance learning', *Imbalanced Learning: Foundations, Algorithms and Applications*, pp. 61–82.

Loh, W.-Y., 2011, 'Classification and regression trees', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 14–23.

Lu, N., Lin, H., Lu, J. & Zhang, G., 2014, 'A customer churn prediction model in telecom industry using boosting', *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1659–1665.

Lu, Z., Dou, Z., Lian, J., Xie, X. & Yang, Q., 2015, 'Content-based collaborative filtering for news topic recommendation', *Twenty-ninth AAAI conference on artificial intelligence*, .

Lyu, F., Wu, Q., Hu, F., Wu, Q. & Tan, M., 2019, 'Attend and imagine: Multi-label image classification with visual attention and recurrent neural networks', *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 1971–1981.

Ma, H., 2013, 'An experimental study on implicit social recommendation', *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, ACM, pp. 73–82.

Ma, H., King, I. & Lyu, M. R., 2012, 'Mining web graphs for recommendations', *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 1051–1064.

Ma, H., Zhou, D., Liu, C., Lyu, M. R. & King, I., 2011, 'Recommender systems with social regularization', *Proceedings of the fourth ACM international conference on Web search and data mining*, ACM, pp. 287–296.

Madjarov, G., Gjorgjevikj, D. & Dzeroski, S., 2012, 'Two stage architecture for multi-label learning', *Pattern Recognition*, vol. 45, no. 3, pp. 1019–1034.

Maloof, M. A., 2003, 'Learning when data sets are imbalanced and when costs are unequal and unknown', *ICML-2003 workshop on learning from imbalanced data sets II*, , vol. 2pp. 2–1.

Mani, I. & Zhang, I., 2003, 'knn approach to unbalanced data distributions: a case study involving information extraction', *Proceedings of workshop on learning from imbalanced datasets*, , vol. 126.

Marsden, M., Zick, C. D. & Mayer, R. N., 2011, 'The value of seeking financial advice', *Journal of Family and Economic Issues*, vol. 32, no. 4, pp. 625–643.

McCallum, A., 1999, 'Multi-label text classification with a mixture model trained by em', *AAAI99 Workshop on Text Learning*, pp. 1–7.

Molnar, C., 2020, *Interpretable Machine Learning*, Lulu. com.

Murphy, K. P., 2012, *Machine learning: A probabilistic perspective*, The MIT Press.

Naulaerts, S., Meysman, P. et al., 2015, 'A primer to frequent itemset mining for bioinformatics', *Briefings in Bioinformatics*, vol. 16, no. 2, pp. 216–231.

Ngonmang, B., Viennet, E. & Tchuente, M., 2012, 'Churn prediction in a real online social network using local community analysis', *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, IEEE, pp. 282–288.

Niu, X., Han, H., Shan, S. & Chen, X., 2019, 'Multi-label co-regularization for semi-supervised facial action unit recognition', *Advances in Neural Information Processing Systems*, pp. 909–919.

Nixon, M. & Aguado, A., 2019, *Feature extraction and image processing for computer vision*, Academic press.

Olatunji, S. O., 2019, 'Improved email spam detection model based on support vector machines', *Neural Computing and Applications*, vol. 31, no. 3, pp. 691–699.

Pan, S. J. & Yang, Q., 2010, 'A survey on transfer learning', *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359.

Prokhorenkova, L. O., Gusev, G., Vorobev, A., Dorogush, A. V. & Gulin, A., 2018, 'Catboost: unbiased boosting with categorical features', *Proceedings of the 2018 Annual Conference on Neural Information Processing Systems*, pp. 6639–6649.

Qi, G.-J., Hua, X.-S., Rui, Y. et al., 2007, ‘Correlative multi-label video annotation’, *Proceedings of the 15th International Conference on Multimedia*, ACM, New York, USA, pp. 17–26.

Quinlan, J. R., 1986, ‘Induction of decision trees’, *Machine Learning*, vol. 1, no. 1, pp. 81–106.

Rai, P., Hu, C., Henao, R. & Carin, L., 2015, ‘Large-scale bayesian multi-label learning via topic-based label embeddings’, *Advances in Neural Information Processing Systems*, pp. 3222–3230.

Rashid, M. M., 2010, ‘A review of state-of-art on kano model for research direction’, *International Journal of Engineering Science and Technology*, vol. 2, no. 12, pp. 7481–7490.

Read, J., 2008a, ‘Multi-label classification using ensembles of pruned sets’, *Proceedings of the 2008th IEEE International Conference on Data Mining (ICDM2008)*, Pisa, Italy, pp. 995–1000.

Read, J., 2008b, ‘A pruned problem transformation method for multi-label classification’, *Proceedings of the 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008)*, Christchurch, New Zealand, pp. 143–150.

Read, J., Pfahringer, B., Holmes, G. & Frank, E., 2011, ‘Classifier chains for multi-label classification’, *Machine learning*, vol. 85, no. 3, pp. 333–359.

Reinartz, W., Thomas, J. S. & Kumar, V., 2005, ‘Balancing acquisition and retention resources to maximize customer profitability’, *Journal of marketing*, vol. 69, no. 1, pp. 63–79.

Ren, Y. & Ji, D., 2017, ‘Neural networks for deceptive opinion spam detection: An empirical study’, *Information Sciences*, vol. 385, pp. 213–224.

Ren, Z., Peetz, M.-H., Liang, S., van Dolen, W. & de Rijke, M., 2014, ‘Hierarchical multi-label classification of social text streams’, *Proceedings of the*

37th international ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp. 213–222.

Rendle, S., 2012, ‘Factorization machines with libfm’, *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 3, p. 57.

Ricci, F., Rokach, L., Shapira, B. & Kantor, P., 2015, *Recommender systems handbook*, Springer.

Rickwood, C. M., Johnson, L. W., Worthington, S. & White, L., 2017, ‘Customer intention to save for retirement using a professional financial services planner’, *financial planning research journal*, vol. 1, no. 1.

Salakhutdinov, R. & Mnih, A., 2008, ‘Probabilistic matrix factorization’, *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*, pp. 1257–1264.

Sarwar, B., Karypis, G., Konstan, J. & Riedl, J., 2001, ‘Item-based collaborative filtering recommendation algorithms’, *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295.

Schaffhausen, C. R. & Kowalewski, T. M., 2015, ‘Large-scale needfinding: methods of increasing user-generated needs from large populations’, *Journal of Mechanical Design*, vol. 137, no. 7.

Schapire, R. E. & Singer, Y., 2000, ‘Boostexter: a boosting-based system for text categorization’, *Machine Learning*, vol. 39, no. 2, pp. 135–68.

Schedl, M., 2019, ‘Deep learning in music recommendation systems’, *Frontiers in Applied Mathematics and Statistics*, vol. 5, p. 44.

Schlkopf, B., Smola, A. J. & Bach, F., 2018, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, The MIT Press.

Schmidhuber, J., 2015, ‘Deep learning in neural networks: An overview’, *Neural Networks*, vol. 61, pp. 85–117.

Sener, O. & Koltun, V., 2018, ‘Multi-task learning as multi-objective optimization’, *Advances in Neural Information Processing Systems*, pp. 527–538.

Shao, B., Wang, D., Li, T. & Ogihara, M., 2009, ‘Music recommendation based on acoustic features and user access patterns’, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1602–1611.

Shawe-Taylor, J., Cristianini, N. et al., 2004, *Kernel methods for pattern analysis*, Cambridge university press.

Sokolova, M. & Lapalme, G., 2009, ‘A systematic analysis of performance measures for classification tasks’, *Information processing & management*, vol. 45, no. 4, pp. 427–437.

Streich, A. P. & Buhmann, J. M., 2008, ‘Classification of multi-labeled data: a generative approach’, *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases-Part II*, Springer-Verlag, Antwerp, Belgium, pp. 390–405.

Stringfellow, A., Nie, W. & Bowen, D. E., 2004, ‘Crm: profiting from understanding customer needs’, *Business Horizons*, vol. 47, no. 5, pp. 45–52.

Sucar, L. E., Bielza, C., Morales, E. F., Hernandez-Leal, P., Zaragoza, J. H. & Larrañaga, P., 2014, ‘Multi-label classification with bayesian network-based chain classifiers’, *Pattern Recognition Letters*, vol. 41, pp. 14–22.

Sun, Y., Kamel, M. S., Wong, A. K. & Wang, Y., 2007, ‘Cost-sensitive boosting for classification of imbalanced data’, *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378.

Sun, Y., Xue, B., Zhang, M. et al., 2019, ‘Evolving deep convolutional neural networks for image classification’, *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 394–407.

Tang, C., Liu, X., Wang, P., Zhang, C., Li, M. & Wang, L., 2019, ‘Adaptive hypergraph embedded semi-supervised multi-label image annotation’, *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2837–2849.

Teisseyre, P., Zufferey, D. & Słomka, M., 2019, ‘Cost-sensitive classifier chains: selecting low-cost features in multi-label classification’, *Pattern Recognition*, vol. 86, pp. 290–319.

Tenenboim, L., Rokach, L. & Shapira, B., 2009, ‘Multi-label classification by analyzing labels dependencies’, *Proceedings of the 1st International Workshop on Learning from Multi-label Data*, Citeseer, Bled, Slovenia, pp. 117–132.

Tenenboim, L., Rokach, L. & Shapira, B., 2010, ‘Identification of label dependencies for multi-label classification’, *Proceedings of the Second International Workshop on Learning from Multi-Label Data*, pp. 53–60.

Timoshenko, A. & Hauser, J. R., 2019, ‘Identifying customer needs from user-generated content’, *Marketing Science*, vol. 38, no. 1, pp. 1–20.

Tsoumakas, G., Dimou, A., Spyromitros, E., Mezaris, V., Kompatsiaris, I. & Vlahavas, I., 2009, ‘Correlation-based pruning of stacked binary relevance models for multi-label learning’, *Proceedings of the 1st International Workshop on Learning from Multi-Label Data*, pp. 101–116.

Tsoumakas, G., Katakis, I. & Vlahavas, I., 2010, ‘Mining multi-label data’, *Data Mining and Knowledge Discovery Handbook*, Springer, pp. 667–685.

Tsoumakas, G., Katakis, I. & Vlahavas, I., 2011, ‘Random k-labelsets for multilabel classification’, *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I., 2017, ‘Attention is all you need’, *Advances in Neural Information Processing Systems*, pp. 6000–6010.

Vembu, S. & Gärtner, T., 2010, 'Label ranking algorithms: A survey', *Preference learning*, Springer, pp. 45–64.

Verma, J., Gupta, S., Mukherjee, D. & Chakraborty, T., 2019, 'Heterogeneous edge embedding for friend recommendation', *European Conference on Information Retrieval*, pp. 172–179.

Wang, J., Huang, P., Zhao, H., Zhang, Z., Zhao, B. & Lee, D. L., 2018, 'Billion-scale commodity embedding for e-commerce recommendation in alibaba', *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, pp. 839–848.

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C. & Xu, W., 2016a, 'Cnn-rnn: A unified framework for multi-label image classification', *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2285–2294.

Wang, J., Zhao, Y., Wu, X. et al., 2011, 'A transductive multi-label learning approach for video concept detection', *Pattern Recognition*, vol. 44, no. 10, pp. 2274–2286.

Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q. & Kennedy, P. J., 2016b, 'Training deep neural networks on imbalanced data sets', *2016 international joint conference on neural networks (IJCNN)*, IEEE, pp. 4368–4374.

Wang, Y. & Tseng, M. M., 2013, 'Identifying emerging customer requirements in an early design stage by applying bayes factor-based sequential analysis', *IEEE Transactions on Engineering Management*, vol. 61, no. 1, pp. 129–137.

Wang, Z., Hu, Y. & Chia, L.-T., 2010, 'Multi-label learning by image-to-class distance for scene classification and image annotation', *Proceedings of the ACM International Conference on Image and Video Retrieval*, pp. 105–112.

Weiss, G. M., 2004, 'Mining with rarity: a unifying framework', *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 7–19.

West, J., 2012, 'Financial advisor participation rates and low net worth investors', *Journal of Financial Services Marketing*, vol. 17, no. 1, pp. 50–66.

Witten, I. H., Frank, E. & Hall, M. A., 2011, *Data mining: practical machine learning tools and techniques*, USA: Morgan Kaufmann.

Worthington, A. C., 2013, 'Financial literacy and financial literacy programmes in australia', *Journal of Financial Services Marketing*, vol. 18, no. 3, pp. 227–240.

Wu, F., Han, Y., Tian, Q. & Zhuang, Y., 2010, 'Multi-label boosting for image annotation by structural grouping sparsity', *Proceedings of the 18th International Conference on Multimedia*, ACM, Firenze, Italy, pp. 15–24.

Wu, J., Sheng, V. S., Zhang, J., Li, H., Dadakova, T., Swisher, C. L., Cui, Z. & Zhao, P., 2020, 'Multi-label active learning algorithms for image classification: Overview and future promise', *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–35.

Wu, Q., Tan, M., Song, H., Chen, J. & Ng, M. K., 2016, 'Ml-forest: A multi-label tree ensemble method for multi-label classification', *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2665–2680.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y. et al., 2008, 'Top 10 algorithms in data mining', *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37.

Xiao, Y. & Ye, W., 2010, 'Survey of feature normalization techniques for robust speech recognition', *Journal of Chinese information processing*, vol. 24, no. 05, pp. 106–116.

Xie, Y., Li, X., Ngai, E. & Ying, W., 2009, 'Customer churn prediction using improved balanced random forests', *Expert Systems with Applications*, vol. 36, no. 3, pp. 5445–5449.

Xu, G., Gu, Y., Dolog, P., Zhang, Y. & Kitsuregawa, M., 2011, 'Semrec: A semantic enhancement framework for tag based recommendation', *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*, .

Xu, Q., Jiao, R. J., Yang, X., Helander, M., Khalid, H. M. & Opperud, A., 2009, 'An analytical kano model for customer need analysis', *Design Studies*, vol. 30, no. 1, pp. 87–110.

Xuan, J., Lu, J., Zhang, G., Da Xu, R. Y. & Luo, X., 2017, 'A bayesian nonparametric model for multi-label learning', *Machine Learning*, vol. 106, no. 11, pp. 1787–1815.

Yang, B., Sun, J.-T., Wang, T. & Chen, Z., 2009, 'Effective multi-label active learning for text classification', *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Paris, France, pp. 917–926.

Yao, Y. & Doretto, G., 2010, 'Boosting for transfer learning with multiple sources', *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 1855–1862.

Yeh, C.-K., Wu, W.-C., Ko, W.-J. & Wang, Y.-C. F., 2017, 'Learning deep latent spaces for multi-label classification', *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 2838–2844.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L. & Leskovec, J., 2018, 'Graph convolutional neural networks for web-scale recommender systems', *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 974–983.

Zadrozny, B., Langford, J. & Abe, N., 2003, 'Cost-sensitive learning by cost-proportionate example weighting', *Third IEEE international conference on data mining*, IEEE, pp. 435–442.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., Stoica, I. et al., 2010, 'Spark: cluster computing with working sets.', *HotCloud*, vol. 10, no. 10-10, p. 95.

Zdanowicz, J. S., 2004, 'Detecting money laundering and terrorist financing via data mining', *Communications of the ACM*, vol. 47, no. 5, pp. 53–55.

Zeka, B., Antoni, X., Goliath, J. & Lillah, R., 2016, 'The factors influencing the use of financial planners', *Journal of Economic and Financial Sciences*, vol. 9, no. 1, pp. 76–92.

Zhan, W. & Zhang, M.-L., 2017, 'Inductive semi-supervised multi-label learning with co-training', *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1305–1314.

Zhang, M.-L., Li, Y.-K., Liu, X.-Y. & Geng, X., 2018, 'Binary relevance for multi-label learning: an overview', *Frontiers of Computer Science*, vol. 12, no. 2, pp. 191–202.

Zhang, M.-L. & Zhang, K., 2010, 'Multi-label learning by exploiting label dependency', *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD2010)*, ACM, Washington, DC, USA, pp. 999–1008.

Zhang, M.-L. & Zhou, Z.-H., 2006, 'Multi-label neural networks with applications to functional genomics and text categorization', *IEEE Transactions On Knowledge And Data Engineering*, vol. 18, no. 10, pp. 1338–1351.

Zhang, M.-L. & Zhou, Z.-H., 2007, 'Ml-knn: A lazy learning approach to multi-label learning', *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048.

Zhang, M.-L. & Zhou, Z.-H., 2014, 'A review on multi-label learning algorithms', *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837.

Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N. J., Xie, X. & Li, Z., 2018, 'Drn: A deep reinforcement learning framework for news recommendation', *Proceedings of the 2018 World Wide Web Conference*, pp. 167–176.

Zheng, Q. & Skillicorn, D. B., 2016, 'Spectral graph-based semi-supervised learning for imbalanced classes', *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, IEEE, pp. 960–967.

Zhou, Z.-H., 2012, *Ensemble methods: foundations and algorithms*, CRC Press.

Zhou, Z.-H., 2017, 'A brief introduction to weakly supervised learning', *National Science Review*.

Zhou, Z.-H. & Feng, J., 2017, 'Deep forest: towards an alternative to deep neural networks', *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3553–3559.

Zhou, Z.-H. & Liu, X.-Y., 2006, 'Training cost-sensitive neural networks with methods addressing the class imbalance problem', *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63–77.

Zhu, J., Wang, H., Zhu, M., Tsou, B. K. & Ma, M., 2011, 'Aspect-based opinion polling from customer reviews', *IEEE Transactions on affective computing*, vol. 2, no. 1, pp. 37–49.