Generalized Plackett-Luce Model for Label Ranking

Master's Thesis submitted to the
Faculty of Informatics of the *Università della Svizzera Italiana*
in partial fulfillment of the requirements for the degree of
Master of Science in Informatics
Intelligent Systems

presented by
Suttipong Mungkala

under the supervision of
Dr.Giorgio Corani
co-supervised by
Dr.Alessandro Antonucci

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Suttipong Mungkala
Lugano, January 20, 2015

To my beloved mother and father,
who always believe in me

Make everything as simple as possible,
but not simpler.

Albert Einstein (1879-1955)

Abstract

A label ranking is a sophisticated prediction task where the goal is to assign labels from a finite set of preferred labels, and to rank them according to the nature of the input. Thus, one has to predict the most probable rank of the label, given the characteristics of the object. The Plackett-Luce model has been used for the purpose of the label ranking and can be represented as a vase interpretation. The Plackett-Luce model is based on some restrictive assumptions. In this thesis, we study novel variants of the Plackett-Luce model, aimed at relaxing its most restrictive assumptions. We investigate various issues pertaining to the task of label ranking and propose new models to improve the performance of predictive models for label ranking problems. We extend the Plackett-Luce model, by having two vases (PL2) allowing each vase to have its own proportions whilst the traditional one has only one vase (PL1) for all stages. We also introduce the hybrid model, which performs statistical tests to select the best fit model between PL1 and PL2 model. We evaluate our models on some real-world benchmark datasets (i.e., Nascar[1] 2002 season dataset and KEBI[2] label ranking datasets). We show a number of advantages of the proposed models over the traditional model, especially on datasets with higher numbers of labels to rank.

[1] http://sites.stat.psu.edu/ dhunter/code/btmatlab/nascar2002.txt
[2] https://www.uni-marburg.de/fb12/kebi/research/repository/labelrankingdata

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to all the people that contributed to this project. I would like to thank my supervisor, Giorgio Corani for offering me the opportunity to work on this thesis, for the effort he guaranteed in helping me step-by-step in the development of the thesis, and especially for his priceless moral support and his patience answering my questions and helping me solve the issues encountered during the thesis. In addition, I would also like to thank Shuai and Alessandro and for their continual support and constructive criticisms throughout my studies.
Last but not least, I would like to thank my family and my friends for continuously supporting my life choices. Thank you all.

Contents

Figures

Tables

Chapter 1

Introduction

There are many websites on the Internet currently collecting information from many different users and using machine learning and statistical methods to benefit from it. Websites like Amazon[1] and Netflix[2] use the preferences of a group of people to make recommendations to other people. By using data about which movies each customer liked, Netflix is able to recommend movies to other customers that they may never have even heard. Amazon tracks the purchasing habits of its shoppers, and when you log onto the site, it uses this information to suggest products you might like. The ability to collect information and the computational power to interpret it has enabled great collaboration opportunities and a better understanding of users, which is worth a lot of money to both websites.

Figure 1.1. The "Recommended for You" feature on the Amazon.com website. The algorithm aggregates items from these similar customers, eliminates items the user has already purchased or rated, and recommends the remaining items to the user.

To understand the real world problems about the label ranking, let us consider a simple example. Suppose an on-line movie rental website (e.g., Netflix) has hundred thousands of movies and users have their own preferences on the movies. How do we use these preferences

[1] http://www.amazon.com
[2] http://www.netflix.com

of a group of people to make recommendations to other people, especially for new users? The historical preferences data may look like the data in Table 1.1. These records are considered as a label ranking dataset, where a set of preferences (labels) are known. Suppose a new user, Nico has no idea which movie she wants to watch at the glance. Applying a proper label ranking method would be able to predict some set of movies that Nico might like based on her similar characteristics to other existing users. Predicting a complete order of labels have some advantages over a single class or a subset of labels. Imagine a user wants to rent Superman movie but it is not available, the movie rental website however can recommend other movies that this user might also likes based on the movies that other users have previously rented.

Table 1.1. An example of label raking dataset from a movie rental database. The users are represented with their features (Name, Sex, Age and Status). The preferences over a set of three labels (Batman, Superman and Titanic). The symbol "$\succ$" expresses preferences, for example "Superman $\succ$ Titanic" means a user prefers Superman movie rather than Titanic movie.

Name	Sex	Age	Status	Preferences
John	M	20	Single	Superman $\succ$ Titanic $\succ$ Batman
Julia	F	25	Single	Titanic $\succ$ Batman $\succ$ Superman
Ana	F	30	Married	Batman $\succ$ Titanic
Jack	M	18	Single	Superman $\succ$ Batman
Nico	F	28	Single	???

As we have seen, the label ranking problem is not only the task of predicting a single class of label, but a complete order of a set of labels. The computation of this kind of problem is extremely time intensive, with $n! = O(2^n)$ as worst case, where n denotes the number of labels to rank. Thus, label ranking can be regarded as a standard classification task, with a class variable whose number of possible states increases exponentially in the number of labels. To produce a sensible probability distribution over such a large space we need some simplifying assumptions. Another challenge that can happen in real world situations is the case of incomplete ranking dataset. In Table 1.1, Jack provides his preference movies as $Superman \succ Batman$, information about $Titatic$ is not given by him, perhaps he does not like the movie or has not watched the movie.

This thesis proposed a generalization of one of the most popular approaches to label ranking, called *Plackett-Luce model* (PL1) [Luce, 1959] and [Plackett, 1975]. The PL1 model is the multistage model and can be considered as a vase model interpretation [Silverberg, 1980]. The traditional model has one vase and is based on the assumption of proportions of preferences or labels are equal in every stage [Marden, 1996], which might not be existed in the real-world datasets since the preferences might change over time. We relax this assumption by allowing different proportions of preferences in each stage.

We investigate various issues pertaining to the task of label ranking and propose the new methods to improve the performance of predictive models for label ranking problems. We describe the general idea of the first proposed model, namely the *Two-Vases Plackett-Luce Model* (PL2). We demonstrate if this model is doing the right thing, by applying it on known parameters created from the vase model distribution. We introduce the mixture between the PL2 model and PL1 model, so called *Hybrid Plackett-Luce Model* (PLH). The PLH model will automatically choose the best fit model to perform label ranking problems. We show how to select the best fit

model by running a statistical test. We evaluate the models on real-world datasets to compare the performances of our proposed models and the PL1 model. We implement a software package of the proposed models, along with an interesting example of learning the models to rank NASCAR [3] dataset and KEBI [4] label ranking datasets.

1.1 Outline of the Thesis

The rest of the document is organized as follows:

- **Chapter 2** takes a more detailed look at the label ranking problem and probabilistic models for solving such problem. We explain more in detail the idea of instance based label ranking. The methods to evaluate the predictive model performance are mentioned.
- **Chapter 3** reviews previous work related to label ranking based on PL1 model. The traditional one vase model interpretation and its parameter estimates using *maximum likelihood estimation* (MLE). Gradient Descent, *Newton-Raphson* (NR) and *Majorize/Minimize* (MM) algorithm to fit the model is discussed.
- **Chapter 4** explains our proposed models. We first describe the PL2 model interpretation and explain how it deals with the label ranking problem. We explain how to estimate parameters and predict the most probable rank given sample datasets. We introduce the PLH model which is the mixture model between the PL1 and PL2 model. Finally, we explain the method of selecting the best fit model for particular dataset.
- **Chapter 5** applies the PL2 model to known-parameters distribution to make sure if the model is doing the right thing. We explain the evaluation methodology which is followed along with the evaluation measures that we use to evaluate the effectiveness of the methods.
- **Chapter 6** reports the results of our proposed models. We present datasets used in the evaluation and then describe the baseline methods to which we compare our approach. Finally we present and discuss the experimental results.
- **Chapter 7** reviews R-package implemented for algorithms we developed in this thesis. The package is described and its use is illustrated through two real datasets analysis.
- **Chapter 8** concludes the contributions of this thesis and suggests some interesting directions for future work.

[3] http://sites.stat.psu.edu/ dhunter/code/btmatlab/nascar2002.txt
[4] https://www.uni-marburg.de/fb12/kebi/research/repository/labelrankingdata

Chapter 2

State of the Art in Label Ranking

Different types of machine learning algorithms have been approached. Ranking by pairwise comparison [Hüllermeier et al., 2008], which reduces the original problem into a set of smaller binary classification problems. Decision-tree learning such as CART can be adapted to solve label ranking problems by extending the concept of purity to label ranking data, [Yu et al., 2008]. The *k-nearest neighbors* (KNN) is an effective machine learning method for solving conventional classification problems. It is generally regarded as an instance-based learning or lazy method because hypotheses are constructed locally and the computation is deferred until the test dataset is obtained. A number of label ranking methods can be found in a survey by [Gärtner and Vembu, 2010].

Researchers have extended the KNN concept to handle the label ranking learning problem. The instance-based logistic regression method, [Cheng and Hüllermeier, 2009], which is a combination of instance-based learning and logistic regression techniques. It includes the statistics of the KNN as features in logistic regression. Mallows model was proposed for the label ranking task [Cheng and Hüllermeier, 2008], where an instance-based learning was used to fit this model. Instance-based approach simply stores the training data and defer the processing of these data until an estimation for a new instance is requested. Instance-based approach therefore have a number of potential advantages, especially in the context of the label ranking problem. Subsequently, in [Cheng and Hüllermeier, 2010], the label ranking method based on Plackett-Luce model was introduced as an alternative to the Mallows model. The Plackett-Luce model is appropriate when each observation provides either a complete ranking of all labels, or a partial ranking of only some of the labels.

The remaining parts of this chapter, we take a more detailed look at label ranking problem in Section 2.1. and describe the Placket-Luce model for solving such problem in Section 2.2. In Section 2.3, we explain more in detail the idea of instance-based label ranking. Evaluating the prediction of label ranking is provided in Section 2.4. The chapter ends with concluding remarks in Section 2.5.

2.1 Problem Setting

Let us first introduce the preference learning problem in its basic formulation together with the necessary formalism. Let $X = \{x_1, \cdots, x_k\}$ be an instance domain and $Y = \{y_1, \cdots, y_m\}$ be a finite set of preferences/labels. A ranking can be considered as a type of preference relation,

and therefore $y_i \succ_x y_j$ indicates that y_i is preferred over y_j given the instance x. To illustrate, suppose that instances are users (characterized by attributes such as age, income and gender) and "$\succ$" is a preference relation on a finite set of mobile phones such as Iphone, Samsung and Nokia.

Assume that π_x is a permutation of $\{1, 2, \cdots, m\}$ such that π_x is the position of label y_i in the rank associated with instance x. The class of permutations of $\{1, \cdots, m\}$ is denoted by Ω. We refer to elements $\pi \in \Omega$ as both permutations and rankings. Given a permutation π of the m labels, we regard it as an implicit specification of a set of preferences as follows:

$$y_{\pi_x}^{-1}(1) >_x y_{\pi_x}^{-1}(2) >_x \ldots >_x y_{\pi_x}^{-1}(m) \tag{2.1}$$

where $\pi^{-1_x}(i)$ is the index of the label that has rank i. For example, $\pi^{-1}(3) = 5$ means that the label with rank 3 is label 5. Note that, we do not consider incomplete rankings scenario in this thesis.

To motivate this learning problem, let us consider the following example. Suppose we have a set of users which are represented by features (age, income and gender) and their ordered preferences over a set of mobile phones {Iphone, Samsung, Nokia}. The goal of this task is to learn how the users rank these mobile phones. The outcome is the prediction of the order of preference for new customers based on similarities between a new instance and training instances.

Table 2.1. An example of label ranking dataset. The users are represented with their features (age, income and gender). The preferences over a set of three labels Ω = {Iphone, Samsung, Nokia}.

User	Gender	Income	Status	Label Ranking
John	M	50k	Single	Iphone $\succ$ Samsung $\succ$ Nokia
Julia	F	40k	Single	Samsung $\succ$ Iphone $\succ$ Nokia
Ana	F	20k	Married	Nokia $\succ$ Samsung $\succ$ Iphone
Jack	M	60k	Single	Iphone $\succ$ Samsung $\succ$ Nokia
Nico	F	35k	Single	???

Table 2.2. Encoding a ranking with integers. $\pi(i)$ is the index of the label ranked in position i. The index of each label is 1:Iphone, 2:Samsung and 3:Nokia.

User	Gender	Income	Status	Iphone	Samsung	Nokia
John	M	50k	Single	1	2	3
Julia	F	40k	Single	2	1	3
Ana	F	20k	Married	3	2	1
Jack	M	60k	Single	1	2	3
Nico	F	35k	Single	?	?	?

2.2 Plackett-Luce Model

The initial idea of the PL1 model started by the Luce choice axiom [Luce, 1959], the probability of selecting an item over another from a set of many items is not affected by the presence

or absence of the other items in the set. Assume we have a set of items, and a set of choice probabilities that satisfy Luce choice axiom, and consider picking one item at a time out of the set, according to the choice probabilities. Such samples give a total ordering of items, which can be considered as a sample from a distribution over all possible orderings. The form of such distribution was first considered by [Plackett, 1975] in order to model probabilities in a horse racing. The other applications based on the Plackett-Luce model have been varied including document ranking [Cao et al., 2007], modelling electorates [Gormley and Murphy, 2005], and assessing potential demand for electric cars [Beggs et al., 1981].

We refer to the book by [Marden, 1996], which is a good source for the material in this section. The PL1 model is based on the positive scores $v_i, i = 1, \cdots, m$ where v_i corresponds to label i. The higher value of i the more probability that label is is ranked first. The v_i's are proportional to the probability that label i is ranked first. Assume a set of possible permutation $\pi = \{\pi_1, \cdots, \pi_n\}$ and $v = (v_1, \cdots, v_m) \geq 0$ are assigned to label $1, \cdots, m$. the probability to observe this is defined as:

$$PL1(\pi) = \frac{v_{\pi_1}}{v_{\pi_1} + ... + v_{\pi_m}} \cdot \frac{v_{\pi_2}}{v_{\pi_2} + ... + v_{\pi_m}} \cdot ... \cdot \frac{v_{\pi_{m-1}}}{v_{\pi_{m-1}} + ... + v_{\pi_m}} \cdot \frac{v_{\pi_m}}{v_{\pi_m}} \tag{2.2}$$

Note that the last term $\frac{v_{\pi_m}}{v_{\pi_m}} = 1$ does not affect the result of equation (2.2). To understand this equation let us give a simple example below.

Example 2.2.1 *Consider a rank $\pi = (B, A, C)$. Then we have that $p(\pi)$ is the probability of B being ranked first, times the probability of A being ranked second given that B is ranked first, times the probability of C being ranked third given that B and A are ranked first and second. The probability of this rank is then:*

$$P(\pi|v) = \frac{v_B}{v_B + v_A + v_C} \cdot \frac{v_A}{v_A + v_C} \cdot \frac{v_C}{v_C}$$

Alternatively, the PL1 model is the multistage model and can be considered as a vase model interpretation according to [Silverberg, 1980]. Imagine a multistage experiment where at each stage we are selecting a ball from a vase of coloured balls. The number of balls of each colour are in proportion to v_i. At the first stage, a ball o_1 is drawn from the vase. At the second stage, another ball is drawn, if it is the same colour as the first, then put it back, and keep on trying until a new colour o_2 is selected. The further stages are the same. Continue through the stages until there is only one ball you have not chosen. It is ranked last. We provide more detail including numerical examples regarding the vase interpretation in the next chapter.

2.3 Instance-Based Label Ranking

The author [Cheng and Hüllermeier, 2010] proposed an instance-based method for label ranking based on PL1 model. The idea is to predict the most probable rank for a given rank instance based on local information, i.e., labels of neighbors. In principle, the method is based on the K nearest neighbors approach.

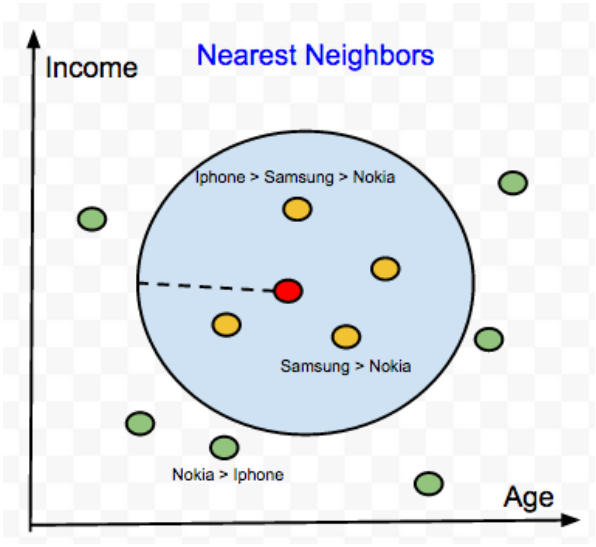


Figure 2.1. Instance-based label ranking: nearest neighbors. The users (circles) are represented with their features (age, income and gender) and preferences {Iphone, Samsung, Nokia}.

Now consider a ranking instance $x = \{x_1, ..., x_K\}$ which is the nearest neighbors of x based on a similarity measure (e.g. distance functions) in a training set, and $K \in N$ is a number of neighbors. Each instance x_i is a ranking π_i of the labels $y \in Y$, and the number of labels to rank is $i \in \{2, ..., m\}$. $\pi(i)$ is the index of the label ranked in position i. Regarding the P-L model in equation (2.2), the probability to observe the rankings $\pi = \{\pi_1, ..., \pi_K\}$ in the K neighbors, given the parameters $v = (v_1, ..., v_M) \geq 0$ are assigned to each label:

$$PL1(\pi|v) = \prod_{k=1}^{K} \prod_{i=1}^{m_k - 1} \frac{v_{(i,k)}}{\sum_{j=i}^{m_k} v_{(i,j)}} \tag{2.3}$$

where $v_{(i,k)}$ is the score of the label ranked at the i^{th} position in the x_k instance.

The first summand is the number of instances according to the selected K neighbors. The second summand is $m_j - 1$ as the last ranked label in each instance is not included by the summation since it does not contribute to the probability in the PL1 model. Figure 2.2 shows an instance-based label ranking approach, where K instances are selected based on distance measures, $y_{\pi_x}(?)$ means instance x, any label y is ranked in any order which denoted by ? sign, and the learning function is mapping the rankings into a total order denoted by $y_{\pi_x}^{-1}(i)$.

Algorithm 1: Instance-Based Label Ranking

Input: **x**:query, τ:training data, k:integer
Output: **o**:most probable rank

1 search k nearest neighbors of **x** in τ
2 get neighbor ranks instances $\sigma = \{\sigma_1, \cdots, \sigma_k\}$
3 get parameter estimates of **o** using equation (2.3)
4 sort **o** in descending order to get most probable rank

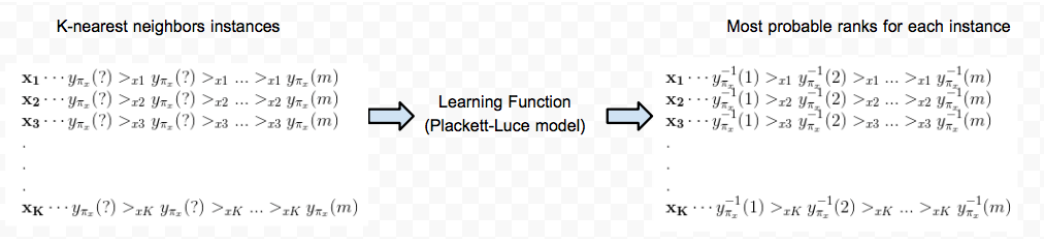


Figure 2.2. Instance-based label ranking approach.

2.4 Model Evaluation

Evaluating the predictions of label ranking is more complicated than the traditional classification. The evaluation of label ranking problems requires different measures than those used in the case of single label problems. Unlike the single label problems, which the classification of an example is correct or incorrect, in a label ranking problem, a prediction of a complete rank order may be partially correct or partially incorrect. This can happen when a predictive model correctly assigns a position on a rank to at least one of the labels it belongs to. A predictive model could also assign an example to one or more labels it does not belong to. Several measures have been proposed in the literature for the evaluation of label ranking problems such as Kendall's tau and Spearman's rank correlation coefficient.

Kendall's tau and Spearman's rank correlation coefficient are two commonly used methods to compare the predicted and the actual rank.

2.4.1 Kendall's Tau Correlation Coefficient

Let the actual position O_i of label i in the rank and its prediction O_j. For any sample of size m labels to rank, there are $N = \frac{m(m-1)}{2}$ possible comparisons of points (O_i, O_i) and (O_j, O_j). Suppose that C is number of pairs that are concordant, and suppose that D is number of pairs that are disconcordant. Kendall τ coefficient is defined as:

$$\tau = \frac{C - D}{N} \tag{2.4}$$

Kendall's τ has the range $[-1$ to $1]$. If the two ranks are the same the tau has value 1, if one rank is the reverse of the other the τ has value -1, and if the the two ranks are independent the τ is expected to be approximately zero.

2.4.2 Spearman's Rank Correlation Coefficient

Let the actual position O_i of label i in the rank and its prediction O_j. For any sample of size m labels to rank, the m raw scores (O_i,O_i) are converted to ranks (O_j, O_j). Suppose that d_i is the rank difference between the ranks of O_i and O_i. Spearman ρ coefficient is defined as:

$$\rho = 1 - \frac{6\sum d_i^2}{m(m^2-1)} \tag{2.5}$$

Example 2.4.1 *Given the actual rank* $1 \succ 2 \succ 3 \succ 4 \succ 5$ *and the estimated rank* $2 \succ 1 \succ 4 \succ 3 \succ 5$. *To obtain the* τ*, we first find the number of concordant and disconcordant pairs and follow equation* (2.4). *The* ρ *can be computed by using the deviation the difference between the actual and the estimated rank as shown in equation* (2.5).

Table 2.3. An example of Kendall's Tau and Spearman's Rank calculation.

Rank		Kendall		Spearman	
Actual	Estimate	C	D	d	d^2
1	2	3	1	1	1
2	1	3	0	1	1
3	4	1	1	1	1
4	3	1	0	1	1
5	5	–	–	0	0

From the above table, we can calculate the τ and ρ as follows:

$$\tau = \frac{C-D}{N}$$
$$\tau = \frac{8-2}{8+2}$$
$$\tau = 0.6$$

$$\rho = 1 - \frac{6\sum d_i^2}{m(m^2-1)}$$
$$\rho = 1 - \frac{6*4}{5(5^2-1)}$$
$$\rho = 0.8$$

Kendall τ has an intuitive interpretation, the proportion of concordant pairs minus the proportion of discordant pairs whereas Spearman ρ does not have a similar intuitive interpretation.

2.5 Chapter Summary

We have explained a general idea of label ranking which is a special type of preference learning problem. We have shown how the problem can be solved using probabilistic models (e.g.,

Plackett-Luce model). Instance-base learning approach was introduced to let the probability distribution over ranks depend on the features. We have explained rank correlation coefficients (Kendall's tau and Spearman's rank) used to compare the predicted and the actual rank. The correlation coefficients are useful to evaluate the performance of predictive models for label ranking.

Chapter 3

One-Vase Plackett-Luce Model

The PL1 model, we introduced in Section 2.2, is among the most popular tools for label ranking. This model has been applied to many applications (e.g., document ranking [Cao et al., 2007], modelling electorates [Gormley and Murphy, 2005]) in machine learning, psychology or economy.

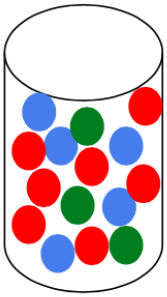
In this chapter, we take a more detailed look at the PL1 model. In particular we provide a so-called *vase* interpretation of the model described in Section 3.1. In Section 3.2, we explain the typical way to fit the model by *maximum likelihood estimation* (MLE) of the parameters. We describe a couple of methods to do this using Gradient Descent, *Newton-Raphson* (NR) and *Minorise/Maximise* (MM) algorithm. We provide numerical examples of how estimate parameters for label ranking using MM algorithm in Section 3.3. Section 3.4 concludes this chapter.

3.1 The Vase Interpretation

It is useful to use the vase model metaphor [Silverberg, 1980] to interpret rankings as output of stochastic processes in the PL1 model. Consider an experiment where at each stage a ball is drawn from a vase of labeled balls $v = \{v_1, ..., v_m\}$. The number of balls of each label in the vase is proportional to v_i where $i = 1, \cdots, m$. At the first stage, a ball is drawn from the vase, and the probability of this selection is $p(\pi_1)$. At the second stage, draw another ball. If it is different label than the first, then it is $p(\pi_2)$. If it happens to draw the same label v_1, then put it back and keep on trying until a new label is selected. Proceed the same approach until a ball of each label has been selected. It is clear that equation (2.2) represents the probability of this sequence. Let us consider a few examples below.

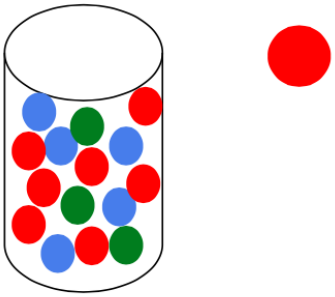
Example 3.1.1 *The vase consists of infinite labeled color balls (v_r, v_g, v_b). We draw balls from the vase until an ordering of labeled is obtained. Now consider drawing a ball at each stage as follows:*

- At the beginning, the vase consists of labeled color balls with the proportions of each label.



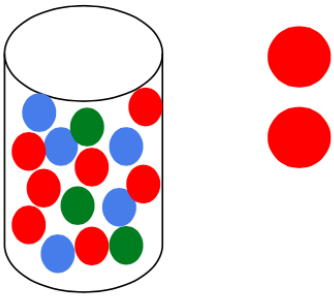
- 1st Stage: We draw a ball and it is red. The probability to extract a red ball at this stage is. $\frac{v_r}{v_r+v_g+v_b}$.

$$\frac{v_r}{v_r+v_g+v_b}$$

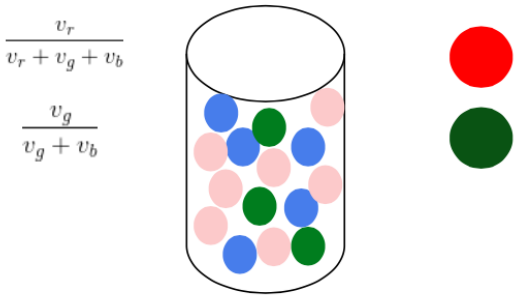


- 2nd Stage: We draw another ball. If it is red again, we put it back and keep drawing until we get a ball that is another color than red.

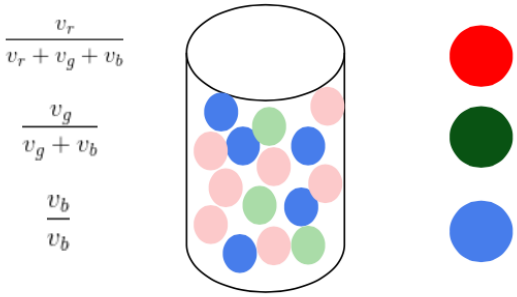
$$\frac{v_r}{v_r+v_g+v_b}$$



- 3rd Stage: We draw another ball and it is green. The probability to extract a green ball from the remaining balls is $\frac{v_g}{v_g+v_b}$.



- 4th Stage: At this stage, there is only blue balls left in the vase. The probability of this selection is obviously $\frac{v_b}{v_b}$.



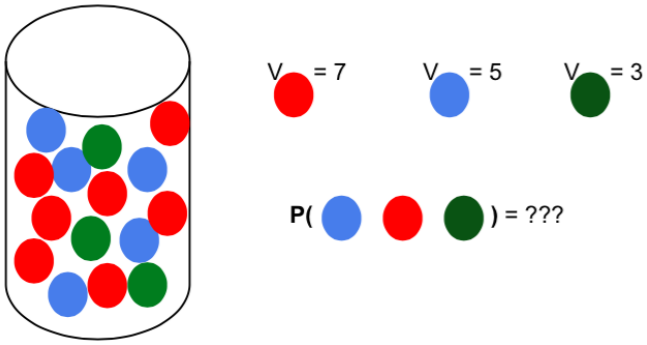
The probability of the selection of this order $r \succ g \succ b$ is

$$p(v_r, v_g, v_b) = \frac{v_r}{v_r+v_g+v_b} \cdot \frac{v_g}{v_g+v_b} \cdot \frac{v_b}{v_b}$$

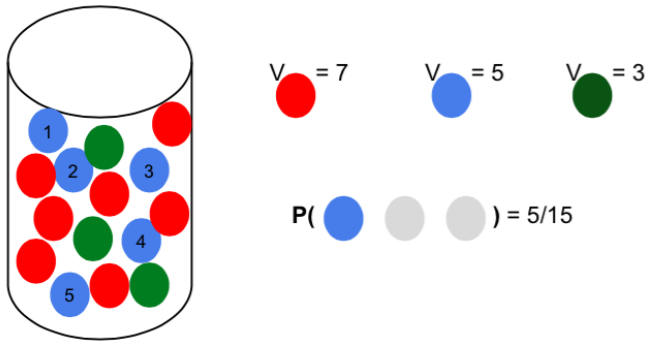
.

Example 3.1.2 *The vase consists of infinite labeled color balls* (v_r, v_g, v_b)*. The probability of selecting balls in an order of blue, red and green is computed as follows:*

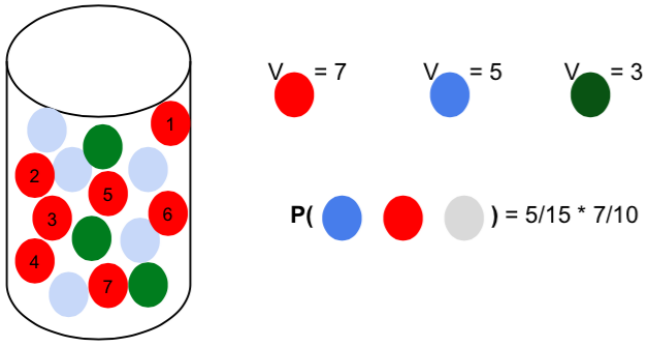
- The vase contains a finite number of labeled balls {red, green, blue}.



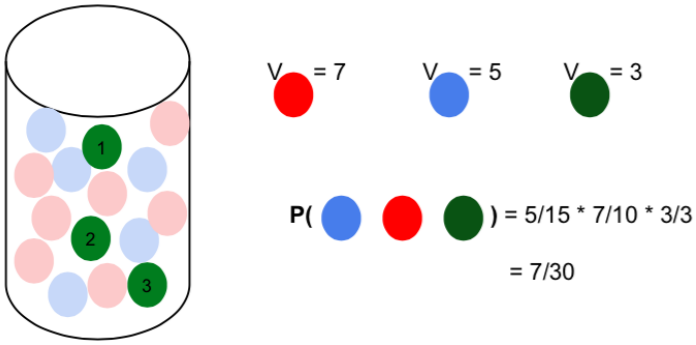
- 1st Stage: A blue ball is drawn from the vase.



- 2nd Stage: A red ball is drawn from the vase.



- 3rd Stage: A green ball is drawn from the vase.



Example 3.1.3 *Given the three labels A,B,C and the ranks, where $A \succ B$ denotes label A is preferred over label B.*

$$A \succ B \succ C$$

$$C \succ A \succ B$$

We can compute the probability of the dataset composed by the two above instances as follows:

$$\frac{v_A}{v_A+v_B+v_C}\cdot\frac{v_B}{v_B+v_C}\cdot\frac{v_C}{v_C}\cdot\frac{v_C}{v_C+v_A+v_B}\cdot\frac{v_A}{v_A+v_B}\cdot\frac{v_B}{v_B}$$

Illustratively the vase model interpretation is:

$$\underbrace{Vase}_{Stage_1}\rightarrow\underbrace{Vase}_{Stage_2}\rightarrow\ldots\rightarrow\underbrace{Vase}_{Stage_m}$$

The general formulation of the vase model can be written as:

$$PL1(\pi|v)=\prod_{i=1}^{m-1}\frac{v_i}{\sum_{j=i}^{m}v_j} \tag{3.1}$$

where v_i is the score of the label ranked at the i^{th} position. Note that this is an equation of conditional probability for one instance of the rank. The computation of the multiple instances are the product of each conditional probability.

3.2 Maximum Likelihood Estimation

Maximum likelihood estimation (MLE) is a classical approach to the learning of probabilistic models from data, which can be also applied to fit the PL1 model [Scholz, 1985]. In MLE, we first compute the *likelihood*, i.e., the probability of the data, given the chosen probability distribution model. The value of the parameters that maximize the likelihood are the maximum likelihood estimates. Once the maximum likelihood estimates are derived, the general theory of MLE provides standard errors, statistical tests, and other results useful for statistical inference. A disadvantage of the method is that if frequently requires strong assumptions about the structure of the data.

Let us consider an example of flipping a coin. We have observed the outcome from a series of n coin tosses. From this coin toss, we have observed x heads. We know that the process of generating heads on coin tosses is a binomial process because we only have two possible outcomes, and the probability p is the same on each coin toss. The likelihood of observing a particular number of heads in n trials is given by the following likelihood function:

$$L(p)=\binom{n}{x}p^x(1-p)^{n-x}$$

We don't know the value of p, but we want to estimate it by choosing $\hat{p}$ that maximizes $L(p)$ for the given n and x. For particular values of n and p, we could get $\hat{p}$ by trial and error, which a computer might simplify it. To obtain a general result that can be expressed in the abstract terms n and p, we need to find the maximum of the likelihood function for p. We can do this using calculus, the point where the derivative of the likelihood function is zero and the second derivative is negative is the maximum. It is usually easier to work with the logarithm of the likelihood function than with the likelihood function itself. The results for the log-likelihood function hold for the likelihood function:

$$\log L(p)=\log\binom{n}{x}+x\log(p)+(n-x)\log(1-p)$$

We take the derivative of this function with respect to p.

$$\frac{\partial \log L(p)}{\partial p} = \frac{x}{p} - \frac{n-x}{1-p}$$

When this function equals zero, we will have either a minimum or a maximum. So solve for p.

$$\begin{aligned} 0 &= \frac{x}{p} - \frac{n-x}{1-p} \\ \frac{x}{p} &= \frac{n-x}{1-p} \\ p(n-x) &= x(1-p) \\ pn - px &= x - px \\ p &= \frac{x}{n} \end{aligned}$$

It turns out the second derivative at this point is negative (not shown), so this is indeed a maximum. Thus our best estimate of p or MLE is $\hat{p} = x/n$.

3.2.1 Gradient Descent

The simplest algorithm for iterative minimization of differentiable functions is known as gradient descent [Burges et al., 2005]. Recall that the gradient of a function is defined as the vector of partial derivatives:

$$\bigtriangledown f(x) = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n}) \tag{3.2}$$

and that the gradient of a function always points towards the direction of maximal increase at that point. Equivalently, it points away from the direction of maximum decrease - thus, if we start at any point, and keep moving in the direction of the negative gradient, we will eventually reach a local minimum. The Gradient Descent algorithm works like this:

1. Initialization Pick a point x_0 as an initial guess.
2. Gradien Compute the gradient at the current guess, $v_n = \bigtriangledown f(x_n)$
3. Update Move by α (the step size) in the direction of that gradient $x_{n+1} = x_n - \alpha v_n$
4. Iterate Repeat steps 1-3 until the function $f(x)$ is close enough to zero (i.e., until $f(x_n) < \varepsilon$ for some small tolerance ε)

Note that the step size α, is simply a parameter of the algorithm and has to be fixed in advance. Note also that this is a first-order method; that is, we only look at the first derivatives of our function.

3.2.2 Newton-Raphson Algorithm

Although gradient descent works in theory, it turns out that in practice, it can be rather slow. To rectify this, we can use information from the second derivative of a function. The most basic method for second order minimization is Newton-Raphson (NR) algorithm [Tjalling., 1995]. We will first work through Newton's method in one variable, and then make a generalization for many variables.

- **One Variable**
  NR is an algorithm for finding the minima or maxima of a given function $f(x)$ in an iterative manner. For minima, the first derivative $f'(x)$ must be zero and the second derivatice $f''(x)$ must have a positive value, while for maxima f'(x) is again zero and the second derivative $f''(x)$ has a negative value. The method for this optimization, starting from a point x_n, the next point x_{n+1} in the iterative series is:

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} \tag{3.3}$$

  As we can see from this method that the first and second derivatives are required in the process. As with any iterative procedure, a convergence criterion must be selected at which the iterative process can be considered to be converged. Typical choices include the gradient $f'(x_n)$, the difference between the functional values of two consecutive iterations $df = f(x_{n-1} - f(x_n))$, or the difference between the values of x itself between two consecutive iterations $dx = x_{n+1} - x_n$.

- **Multiple Variables**
  The algorithm above works only for a single variable. For the multivariate case where a function $f : \mathbb{R}^n \to \mathbb{R}^n$. In this case, we can do the exact same derivation, but replacing derivatives with gradients and second derivatives with Hessian matrix (the matrix of second derivatives).

$$f'(x) \to \bigtriangledown f(x) \tag{3.4}$$

$$f''(x) \to H(f)(x) \tag{3.5}$$

  Thus, the update step becomes:

$$x_{n+1} = x_n - [H(f)(x_n)]^{(-1)} \bigtriangledown f(x_n). \tag{3.6}$$

  which is the optimization algorithm commonly called as Newton's method.

3.2.3 Majorize/Minimize (MM) Algorithm

The MM algorithm [Hunter, 2004] is an optimization strategy used to learn (approximate) MLE in the PL1 model. MM is applicable when each rank provides either a complete ranking of all labels, or a partial ranking of only some of the labels, or a ranking of the top labels. MM algorithm operates by creating a surrogate function that minorizes or majorizes the objective function. When the surrogate function is optimized, the objective function is driven uphill or downhill as needed. In minimization MM stands for majorize/minimize, and in maximization

MM stands for minorize/maximize. The EM algorithm from statistics is a special case of the MM algorithm. We take majorize/minimize and minorize/maximize version for example.

We first focus on the minimization problem. A function $g(\theta|\theta^{(k)})$ is said to majorize the funtion $f(\theta)$ at $\theta^{(k)}$ provided by:

$$f(\theta) \leq g(\theta|\theta^{(k)}), \forall \theta f(\theta^{(k)}) = g(\theta^{(k)}|\theta^{(k)}) \tag{3.7}$$

Then we choose a majorizing function $g(\theta|\theta^{(k)})$ and minimize it. Denote that $\theta^{(k+1)} = \arg\min_\theta g(\theta|\theta^{(k)})$. Iterate until $\theta^{(k)}$ converges.

Now let use take the maximization problem as an example, in which MM stands for minorize/maximize. To maximize $f(\theta)$, we minorize it by a surrogate function $g(\theta|\theta^{(k)})$ and maximize $g(\theta|\theta^{(k)})$ to produce the next iteration $\theta^{(k+1)}$. A function $g(\theta|\theta^{(k)})$ is said to minorize the function $f(\theta)$ at $\theta^{(k)}$ provied that $-g(\theta|\theta^{(k)})$ majorizes $-f(\theta)$.

MM can be used for the finding parameters in ranking problems. [Hunter, 2004] has described this algorithm for ranking problems based on Bradley-Terry [Bradley and Terry, 1952] model and Plackett-Luce model. We first explain the MM algorithm for the simpler Bradley-Terry model and then consider it to fit the PL1 model.

Example 3.2.1 *MM algorithm for Bradley-Terry model*

Consider a sports league with n teams. Assign team i the skill level v_i, where $v_1 = 1$ for identifiability reason. Bradley and Terry proposed the model:

$$P(i \quad \text{beats} \quad j) = \frac{v_i}{v_i + v_j} \tag{3.8}$$

If w_{ij} is the number of times i beats j, then the likelihood of this information is:

$$L(v) = \prod_{i \neq j} \left(\frac{v_i}{v_i + v_j} \right)^{w_{ij}} \tag{3.9}$$

We estimate v by mazimizing $f(v) = \ln L(v)$ and then rank the teams on the basis of the estimates. This is equivalent to the log-likelihood function:

$$f(v) = \sum_{i \neq j} w_{ij} \Big[\ln v_i - \ln(v_i + v_j) \Big] \tag{3.10}$$

Suggests that we linearize on the term $\ln(v_i + v + j)$ to separate parameters. By the supporting hyperplane property and the convexity of $\ln(.)$

$$\ln y \geq -\ln x - x^{-1}(y - x). \tag{3.11}$$

The above inequality produces the surrogate as follows:

$$g(\theta|\theta^{(k)}) = \sum_{i \neq j} w_{ij} \left[\ln v_i - \ln(v_i^{(k)} + v_j^{(k)}) - \frac{v_i + v_j}{v_i^{(k)} + v_j^{(k)}} + 1 \right] \tag{3.12}$$

Because the parameters are separated, we can easily find the optimal point by using the following equation:

$$v_i^{(k+1)} = \frac{\sum_{i \neq j} w_{ij}}{\sum_{i \neq j} \left(\frac{w_{ij} + w_{ji}}{v_i^{(k)} + v_j^{(k)}} \right)} \tag{3.13}$$

3.3 Fitting the Vase Model

The MLE approach described in section 3.2 can be used to learn the PL1 model. Assume that a dataset of K complete rankings is available. The log-likelihood of these data is given by those parameters that maximize equation (3.1):

$$L_{PL1}(v) = \sum_{k=1}^{K} \sum_{i=1}^{m_k-1} \left[\log v_{(i,k)} - \log \sum_{j=i}^{m_k} v_{(i,j)} \right] \tag{3.14}$$

where $v_{(i,k)}$ is the score of the label ranked at the i^{th} position in the x_k instance. A ranking instance $x = \{x_1, ..., x_K\}$ and $K \in N$ is a number of instances.

The MLE of the parameters can thus be obtained using a standard method, e.g. the Newton-Raphson algorithm. In [Hunter, 2004], the author describes an alternative way to fit the PL1 model using a MM algorithm, which shown to be faster than the standard one. The MLE works well in general but it requires a strong assumption to guarantee convergence according to Hunter (Assumption 1): *in every possible partition of the individuals in two non-empty subsets, some individuals in the second set ranks higher than some individual in the first set at least once*. In many cases, this assumption will not be satisfied. As an example of Nascar dataset. The dataset has 4 drivers placed last in every race they participated, which violates this assumption. Therefore these drivers should be removed from the analysis. We take up this example further in section 4.2, demonstrating some different results of fitting the one vase model and the two vases model. MM is an iterative algorithm that minorities the log-likelihood:

$$Q^n(v) = \sum_{k=1}^{K} \sum_{i=1}^{m_k-1} \left[\log v_{(i,k)} - \frac{\sum_{j=i}^{m_k} v_{(i,j)}}{\sum_{j=i}^{m_k} v_{(i,j)}^k} \right] \tag{3.15}$$

where n is the current iteration of the MM algorithm, and therefore $v_{(i,j)}^n$ is the current parameter estimation. For the iteration step, the algorithm can be solved analytically with the following equation:

$$v_t^{n+1} = \frac{w_t}{\sum_{k=1}^{K} \sum_{i=1}^{m_k-1} \delta_{ikt} [\sum_{i=1}^{m_k} v_{(i,j)}]^{-1}} \tag{3.16}$$

where w_t is the number of rankings in which the $t-th$ individual is ranked higher than last, given that $t = 1, \cdots, m$. δ_{ikt} is the indicator of the event that individual t has a rank no better than i in the $k-th$ ranking. v_t^{n+1} is the parameter estimate for the $t-th$ individual in a permutation of m. To understand equation (3.16), let us consider a couple of numerical examples.

Example 3.3.1 *Suppose the three labels to rank are A, B, C and the ranks:*

$$A \succ B \succ C$$

$$B \succ C \succ A$$

Suppose the initial parameters for each labels are $o_A^0 = o_B^0 = o_C^0 = 1$. The estimated parameters at the first iteration is (the results are normalized to one). Finding the estimated parameters that fit the P-L model can be computed as follows:

- The first MM iteration yields:

$$v_A^1 = \frac{1}{(v_A^0 + v_B^0 + v_C^0)^{-1} + (v_B^0 + v_C^0 + v_a^0)^{-1} + (v_C^0 + v_A^0)} = 0.2703$$
$$v_B^1 = \frac{2}{(v_A^0 + v_B^0 + v_C^0)^{-1} + (v_B^0 + v_C^0)^{-1} + (v_B^0 + v_C^0 + v_A^0)^{-1}} = 0.5405$$
$$v_C^1 = \frac{1}{(v_A^0 + v_B^0 + v_C^0)^{-1} + (v_B^0 + v_C^0)^{-1} + (v_B^0 + v_C^0 + v_A^0)^{-1} + (v_C^0 + v_A^0)^{-1}} = 0.1892$$

- The second MM iteration yields

$$v_A^1 = \frac{1}{(v_A^1 + v_B^1 + v_C^1)^{-1} + (v_B^1 + v_C^1 + v_a^1)^{-1} + (v_C^1 + v_A^1)} = 0.2363$$
$$v_B^1 = \frac{2}{(v_A^1 + v_B^1 + v_C^1)^{-1} + (v_B^1 + v_C^1)^{-1} + (v_B^1 + v_C^1 + v_A^1)^{-1}} = 0.5857$$
$$v_C^1 = \frac{1}{(v_A^1 + v_B^1 + v_C^1)^{-1} + (v_B^1 + v_C^1)^{-1} + (v_B^1 + v_C^1 + v_A^1)^{-1} + (v_C^1 + v_A^1)^{-1}} = 0.1779$$

  Continue until the algorithm converges, the parameters for the three labels are:

$$v_A = 0.2192$$

$$v_B = 0.6096$$

$$v_C = 0.1712$$

  These parameters are the ones that maximize the log-likelihood function in equation (3.14).

Now consider other examples which are violated the assumption (Hunter, Assumption 1) mentioned earlier.

Example 3.3.2 *Suppose the three labels to rank are A, B, C and the ranks are:*

$$A \succ B \succ C$$

$$B \succ A \succ C$$

It is clear that label C is always ranked last. This example breaks Hunter's assumption and the P-L model cannot be identified. We shall remove v_C. v_C will not be appeared in the calculation based on MM and therefore v_C has probability zero.

Example 3.3.3 *Suppose the three labels to rank are A, B, C and the ranks are:*

$$A \succ B \succ C$$

$$A \succ C \succ B$$

In this example label A is always ranked first.

- The first MM iteration yields:

$$v_A^1 = \frac{2}{(v_A^0 + v_B^0 + v_C^0)^{-1} + (v_A^0 + v_C^0 + v_B^0)^{-1}} = 0.7143$$
$$v_B^1 = \frac{1}{(v_A^0 + v_B^0 + v_C^0)^{-1} + (v_B^0 + v_C^0)^{-1} + (v_A^0 + v_C^0 + v_B^0)^{-1} + (v_C^1 + v_B^1)^{-1}} = 0.1429$$
$$v_C^1 = \frac{1}{(v_A^0 + v_B^0 + v_C^0)^{-1} + (v_B^0 + v_C^0)^{-1} + (v_A^0 + v_C^0 + v_B^0)^{-1} + (v_C^0 + v_B^0)^{-1}} = 0.1429$$

- The second MM iteration yields:

$$v_A^1 = \frac{2}{(v_A^1 + v_B^1 + v_C^1)^{-1} + (v_A^1 + v_C^1 + v_B^1)^{-1}} = 0.8182$$
$$v_B^1 = \frac{1}{(v_A^1 + v_B^1 + v_C^1)^{-1} + (v_B^1 + v_C^1)^{-1} + (v_A^1 + v_C^1 + v_B^1)^{-1} + (v_C^1 + v_B^1)^{-1}} = 0.0909$$
$$v_C^1 = \frac{1}{(v_A^1 + v_B^1 + v_C^1)^{-1} + (v_B^1 + v_C^1)^{-1} + (v_A^1 + v_C^1 + v_B^1)^{-1} + (v_C^1 + v_B^1)^{-1}} = 0.0909$$

  While we keep running the algorithm, v_A is getting closer to 1, v_B and v_C are getting closer 0. This example also breaks Hunter's assumption.

To prevent the estimates from approaching infinity or probability zero, we should remove a label if it is always ranked first or always ranked last. Then apply the MLE method to the reduced set. An alternative more sophisticated approach would be to consider Bayesian estimation of the parameters. This is however out of our scope.

A ranking refers to the data transformation in which numerical or ordinal values are replaced by their rank when the data are sorted. For example, the numerical data are 0.2, 0.5 and 0.3 are observed, the ranks of these data items in decreasing order would be 2,3 and 1 respectively. Ranks are related to the indexed list of order statistics, which consists of the original dataset rearranged into ascending or decreasing order.

In the case of label ranking, given the estimated parameters v_i^* returned by the MLE. A most probable ranking of this kind of problem can be obtained by sorting the parameters that are associated to labels in decreasing order. For example, three labels to rank are $L1, L2$ and $L3$, the estimated parameters v_i^* for each label are 0.1, 0.6 and 0.2 respectively, the most probable rank of these three labels would be $L2 \succ L3 \succ L1$ meaning $L2$ is ranked first, $L3$ is ranked second and the last rank is $L1$.

3.4 Chapter Summary

In this chapter, we have reviewed some previous works related to label ranking problems, which can be solved using Plackett-Luce model. The model can be interpreted as a vase model interpretation. We have explained the typical way to fit the model by using MLE, motivating some algorithms (Gradient Descent, NR and MM). We have also provided some numerical examples on how to find parameters that maximizes the Plackett-Luce model. In the next chapter, we extend the Plackett-Luce model by having two vases. The proposed model is appreciate model for modelling label ranking problem, in which preferences of users change over time.

Chapter 4

Extending the Plackett-Luce model

The PL1 model in the previous chapter is a method for label ranking which can be interpreted as the vase model. We keep the vase interpretation, and extend this model by having two vases and each vase is allowed to have different proportions of labels. The PL2 model is useful when preferences of users change over time. To understand real world problems, let us consider an example about the English Premier League football. One significant feature of the Premier League in the mid-2000s was the dominance of the "Big Four" clubs: Arsenal, Chelsea, Liverpool and Manchester United. During this decade, they dominated the top four spots, which came with UEFA Champions League qualification.

Suppose that we ask football supporters to make the English football club rankings and there are totally 20 teams in the English Premier League football. Obviously, at the first stage, they rate the Big Four clubs for the top 4 finish than any other clubs. At the second stage, they pay less attention to rank the clubs and give every teams the same priority.

To model such ranking problem, we introduce PL2 model and explain how to estimate parameters of this model in Section 4.1. In Section 4.2, we also propose a hybrid model (PLH), which accomplishes statistical model selection between PL1 and PL2 model. We conclude the chapter in Section 4.3.

4.1 Two-Vases Plackett-Luce Model

4.1.1 Two-Vases Model Interpretation

The PL2 model we propose in this chapter is an attempt to generalize the classical P-L model by allowing two different preference levels over the ranks to be used, respectively for the first and the last part of each ranking. Following the vase interpretation, this model can be regarded as a two-vases model, and each vase has its own proportions of labeled balls. Consider an experiment where depending stage a ball is drawn either from the first vase of labeled balls $V^{(1)} = \{v_1^{(1)}, \cdots, v_m^{(1)}\}$ or from the second vase of labeled balls $V^{(2)} = \{v_1^{(2)}, \cdots, v_m^{(2)}\}$. The number of balls of i^{th} label in each vase is proportional to $v_i^{(1)}$ and $v_i^{(2)}$. Let L be the split point, assume that the first set of parameters $V^{(1)}$ is used for the 1^{th} to L^{th} position of the rank, and the second set of parameters $V^{(2)}$ is used for the positions from L + 1 to m.

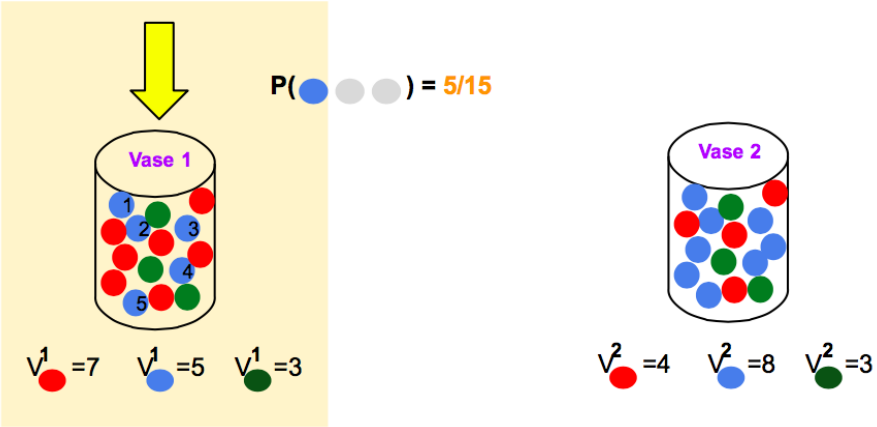
The stages are drawn from the two vases. First, we draw a ball v_1 from the first vase, and the probability of this selection is $p(\pi_1)$. At the second stage, we switch the second vase according

to the L position and draw another ball from this vase. If it is different than v_1, then it is $\pi 2$. If it happens to draw the same labeled ball v_1, put it back and keep on trying until a new label is selected. Continue the same approach until the m^{th} stage where there is only one labeled ball we have not chosen. It is ranked last. Let take a look at some examples to deeply understand the PL2 model:

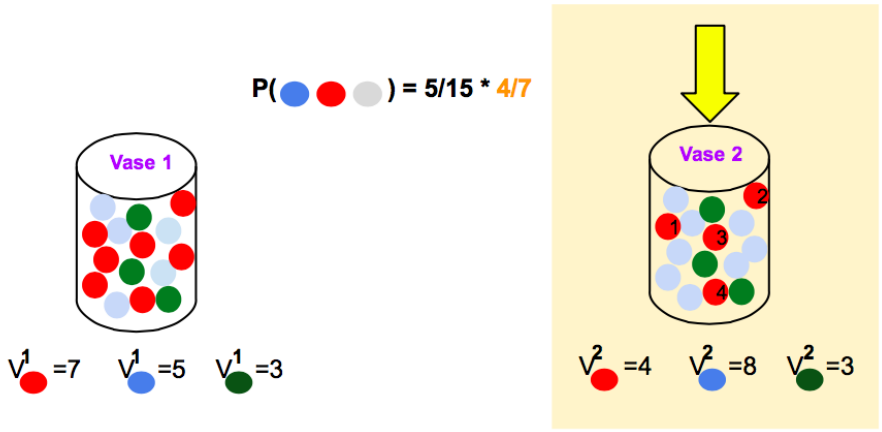
Example 4.1.1 *The two vases consists of infinite labeled color balls {vr,vg,vb}. The proportions of the balls is different in each vase. Assume that the split point L is at the first position. The probability of selecting the balls in an order of blue, red and green can be represented as follows:*

- Each vase has its own proportions of labeled balls.

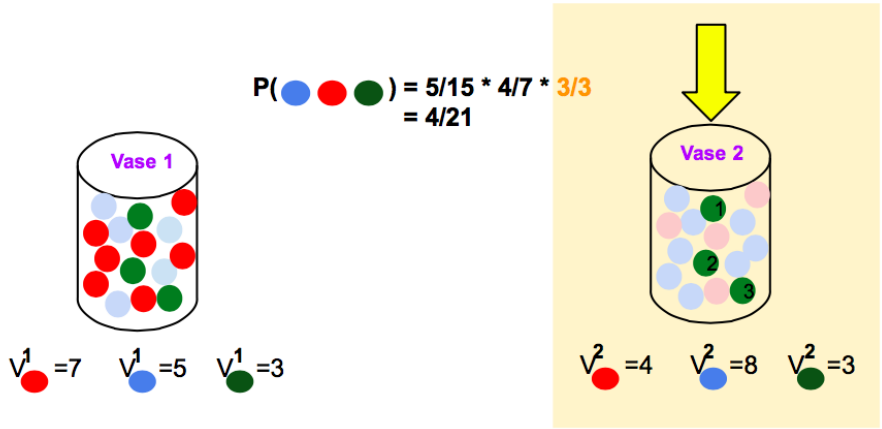
- 1st Stage: We look at the first vase, a blue ball is drawn from the first vase.



- 2nd Stage: We switch to the second vase according to the split point L. A red ball is drawn from the this vase.



- 3rd Stage: We look at the second vase and draw a green ball.



Example 4.1.2 *Suppose we have two vases consisting of infinite labeled balls with known proportions of each label,* $V^{(1)} = \{v_A^{(1)}, v_B^{(1)}, v_C^{(1)}\}$ *and* $V^{(2)} = \{v_A^{(2)}, v_B^{(2)}, v_C^{(2)}\}$*. Assume we draw a ball from the first set* $V^{(1)}$ *at the first stage, and the second set* $V^{(2)}$ *for remaining stages. Continue until a ball of each label has been drawn.*

- Stage 1, $V^{(1)}$ We draw a ball and it is B. The probability of this selection is $\frac{v_B^{(1)}}{v_B^{(1)}+v_A^{(2)}+v_C^{(2)}}$

$$p(\pi|v_B^{(1)}) = \frac{v_B^{(1)}}{v_B^{(1)} + v_A^{(2)} + v_C^{(2)}}$$

- Stage 2, $V^{(2)}$ We switch to the second vase, draw a ball and it is A. The probability of this selection is $\frac{v_A^{(2)}}{v_A^{(2)}+v_C^{(2)}}$.

$$p(\pi|v_B^{(1)}, v_A^{(2)}) = \frac{v_B^{(1)}}{v_B^{(1)} + v_A^{(2)} + v_C^{(2)}} \cdot \frac{v_A^{(2)}}{v_A^{(2)} + v_C^{(2)}}$$

- Stage 3, $V^{(2)}$ The remaining ball is C. The probability of this selection is $\frac{v_C^{(2)}}{v_C^{(2)}}$.

$$p(\pi|v_B^{(1)}, v_A^{(2)}, v_C^{(2)}) = \frac{v_B^{(1)}}{v_B^{(1)} + v_A^{(2)} + v_C^{(2)}} \cdot \frac{v_A^{(2)}}{v_A^{(2)} + v_C^{(2)}} \cdot \frac{v_C^{(2)}}{v_C^{(2)}}$$

Example 4.1.3 *Given the three labels A,B,C and the below ranks. The split position L is first position. $A \succ B$ means label A is preferred over label B.*

$$\underbrace{A}_{1^{st}\,vase} \succ \underbrace{B \succ C}_{2^{nd}\,vase} \quad (first\,rank)$$

$$\underbrace{C}_{1^{st}\,vase} \succ \underbrace{A \succ B}_{2^{nd}\,vase} \quad (second\,rank)$$

We can compute the probability containing all permutations whose 3 labels are A,B and C as follows:

$$P(\pi|v) = \underbrace{\frac{v_A^{(1)}}{v_A^{(1)} + v_B^{(2)} + v_C^{(2)}} \cdot \frac{v_B^{(2)}}{v_B^{(2)} + v_C^{(2)}} \cdot \frac{v_C^{(2)}}{v_C^{(2)}}}_{log-likelihood\,of\,first\,rank} \cdot \underbrace{\frac{v_C^{(1)}}{v_C^{(1)} + v_A^{(2)} + v_B^{(2)}} \cdot \frac{v_A^{(2)}}{v_A^{(2)} + v_B^{(2)}} \cdot \frac{v_B^{(2)}}{v_B^{(2)}}}_{log-likelihood\,of\,second\,rank}$$

Similarly, if the split position L is at the second position between the two vases the probability is:

$$P(\pi|v) = \underbrace{\frac{v_A^{(1)}}{v_A^{(1)} + v_B^{(1)} + v_C^{(2)}} \cdot \frac{v_B^{(1)}}{v_B^{(1)} + v_C^{(2)}} \cdot \frac{v_C^{(2)}}{v_C^{(2)}}}_{log-likelihood\,of\,first\,rank} \cdot \underbrace{\frac{v_C^{(1)}}{v_C^{(1)} + v_A^{(1)} + v_B^{(2)}} \cdot \frac{v_A^{(1)}}{v_A^{(1)} + v_B^{(2)}} \cdot \frac{v_B^{(2)}}{v_B^{(2)}}}_{log-likelihood\,of\,second\,rank}$$

Formally, the probability of obtaining balls for this sequence in the PL2 model, given the balls already chosen is:

$$PL2(\pi|v) = \prod_{i=1}^{m-1} \frac{v_i^{(vase_p)}}{\sum_{j=i}^{m} v_j^{(vase_p)}} \tag{4.1}$$

where

$$vase_p \begin{cases} 1 & \text{if } i, j \leq L \\ 2 & otherwise \end{cases}$$

Note that v_i is the score of the label ranked at the i^{th} position. $vase_p \in \{1, 2\}$ is the integer specifying the score of label i from the first vase or the second vase. L is the position where we switch a stage from the first vase to the second vase. Pictorially, the two vases model where the split position L is at the first position:

$$PL2(\pi|v) = \frac{v_{\pi_1}^{(1)}}{v_{\pi_1}^{(1)} + v_{\pi_2}^{(2)} \ldots + v_{\pi_m}^{(2)}} \cdot \frac{v_{\pi_2}^{(2)}}{v_{\pi_2}^{(2)} + v_{\pi_3}^{(2)} \ldots + v_{\pi_m}^{(2)}} \cdot \ldots \cdot \frac{v_{\pi_{m-1}}^{(2)}}{v_{\pi_{m-1}}^{(2)} + \cdots + v_{\pi_m}^{(2)}} \tag{4.2}$$

$$\underbrace{Vase^{(1)}}_{Stage_1} \rightarrow L_{pos} \rightarrow \underbrace{Vase^{(2)}}_{Stage_2} \rightarrow \cdots \rightarrow \underbrace{Vase^{(2)}}_{Stage_m} \tag{4.3}$$

Similarly, the two vases model where the split position L is at the second position:

$$PL2(\pi|v) = \frac{v^{(1)}_{\pi_1}}{v^{(1)}_{\pi_1} + v^{(1)}_{\pi_2} + \cdots + v^{(2)}_{\pi_m}} \cdot \frac{v^{(1)}_{\pi_2}}{v^{(1)}_{\pi_2} + v^{(1)}_{\pi_3} + \cdots + v^{(2)}_{\pi_m}} \cdot \cdots \cdot \frac{v^{(2)}_{\pi_{m-1}}}{v^{(2)}_{\pi_{m-1}} + \cdots + v^{(2)}_{\pi_m}} \tag{4.4}$$

$$\underbrace{Vase^{(1)}}_{Stage_1} \rightarrow \underbrace{Vase^{(1)}}_{Stage_2} \rightarrow L_{pos} \rightarrow \underbrace{Vase^{(2)}}_{Stage_3} \rightarrow \cdots \rightarrow \underbrace{Vase^{(2)}}_{Stage_m} \tag{4.5}$$

The two vases model are allowed to have different proportions of balls in each vase, but the vase drawn from at each stage depends only on the stage, not on which labels have already been drawn. This model is useful if preferences of users change over time. For example, travellers are care more careful in selecting their top ten travel destinations (the first vase), but after that the travellers less pay less attention. Thus, the destinations are in the second vase may be less important than the first one.

4.1.2 Fitting the Two-Vases Model

In this section we explain how to fit the PL2 model. Generally speaking, the parameters associated to the model can be independently fitted using gradient descent methods (such as Newton-Raphson algorithm). Like many estimating parameters problems involving likelihood function, it is more convenient to work with the natural logarithm of the likelihood function, called the log-likelihood, than it is to work with the likelihood function itself. Let $\theta_i^{(vase_p)} = log(v_i^{(vase_p)})$. The log-likelihood of PL2 model from equation (4.1) is:

$$L_{PL2}(\theta) = \sum_{i=1}^{m-1} log(v_i^{(vase_p)}) - \sum_{i=1}^{m-1} log \sum_{j=1}^{m} v_j^{(vase_p)} \tag{4.6}$$

Let us explain equation (4.6) in more detail. The first term on the right-hand side is the ordering of π mapping independently to label i. Each summation in the second term has components depending on the labels ranked as the k worst, $k = 1, \cdots, m$. For any sample of the rank, there is one case which there are $2^m - 1$ subset. As as result, for any nonempty subset $S' \subset S$ with $k = \#S$, $A(S) = n$ let

$$A(S') = \sum_{i \in S'} \exp(\theta_i^{(vase_p)}) \quad \text{and} \quad B(S') = \#\{\pi^{(j)} | \pi_l^{(j)} \in S', m - k + 1 \leq l \leq m\} \tag{4.7}$$

where $\{\pi^{(1)}, \cdots, \pi^{(n)}\}$ is orderings in a sample. Thus, $B(S')$ is the number of orderings that rank the labels in S at the bottom. The log-likelihood for such sample is then:

$$L_{PL2}(\theta) = n \sum_{i=1}^{m-1} \theta_i^{(vase_p)} - \sum_{S' \subset S} B(S') log(A(S')) \tag{4.8}$$

Then we need to find the maximum of the log-likelihood function, equation (4.8) in an iterative manner. We can apply The Newton-Raphson (NR) algorithm for this purpose. For the maxima, the first derivative $L_{PL2}(v)'$ is zero and the second derivative $L_{PL2}(v)''$ has a negative value. Now let write the first and second derivatives of $L_{PL2}(v)$:

$$\frac{\partial LL_{PL2}}{\partial \theta_i^{(vase_p)}} = n - \sum_{S' \subset S | i \in S'} B(S') \frac{\exp(\theta_i^{(vase_p)})}{A(S')}, \tag{4.9}$$

$$\frac{\partial^2 LL_{PL2}}{\partial^2 \theta_i^{(vase_p)}} = - \sum_{S' \subset S | i \in S'} B(S') \Big[\frac{\exp(\theta_i^{(vase_p)})}{A(S')} - \frac{\exp(2\theta_i^{(vase_p)})}{A(S')^2} \Big], \tag{4.10}$$

and

$$\frac{\partial LL_{PL2}}{\partial \theta_i^{(vase_p)} \partial \theta_j^{(vase_p)}} = - \sum_{S' \subset S | i,j \in S'} A(S') \frac{\exp(\theta_i^{(vase_p)}) \exp(\theta_j^{(vase_p)})}{A(S')^2}, i \neq j, \tag{4.11}$$

Now we can apply the NR algorithm to find the parameters that maximize the log-likelihood function in equation (4.8). One important thing that needs to be taken into account is that the the calculation of the log-likelihood can be infinity. To avoid the estimates from approaching infinity, we should remove an object if it is always ranked first or last, and apply the algorithm to the reduced set. This is consistent with the Assumption 1 mentioned by Hunter.

To obtain the most probable rank we need to consider the parameters from the two vases. Let us consider a simple example.

Example 4.1.4 *Suppose that the parameters obtained from the two vases are* $V^{(1)} = \{v_1^{(1)} = 0.3, v_2^{(1)} = 0.2, v_3^{(1)} = 0.5\}$ *and* $V^{(2)} = \{v_1^{(2)} = 0.2, v_2^{(2)} = 0.5, v_3^{(2)} = 0.3\}$ *respectively. Assume that the split point L is the first position. We can obtain the most probable rank as follows:*

- We first need to sort the parameters of the two vases in decreasing order and therefore the sorted parameters are: $V^{(1)} = \{v_3^{(1)}, v_1^{(1)}, v_2^{(1)}\}$ and $V^{(2)} = \{v_2^{(2)}, v_3^{(2)}, v_1^{(2)}\}$.
- We look at the first vase and pick one parameter with the highest score. It is $v_3^{(1)}$. Let us call this label as o_3.
- We switch to the second vase and pick the remaining parameters in decreasing order, except the one that has been picked from the first vase. The most probable rank is then $o_3 \succ o_2 \succ o_1$.

4.2 Hybrid Model

In this section, we introduce a hybrid model (PLH) which performs statistical model selection between PL1 and PL2 model. A good model selection technique will balance goodness of fit with simplicity. The more complex model will be better able to adapt its shape to fit the data (for example, a third-order polynomial can exactly fit four points), but the addional parameters may not represent anything useful (Maybe those four points are just randomly distributed a bout a straight line). Goodness of fit is generally determined using a likelihood ratio approach, or an approximation of this, leading to a chi-squared test. The complexity is normally measured by counting the number of parameters in the model.

We would like to know whether PL2 model fits significantly better than PL1 model as a predictor for given training data. How can we accomplish this? We adopt the likelihood ratio test (LRT). This test is sometimes described as tests for differences among nested models, because one of the models can be said to be nested within the other (a simple model is nested within

a more complicated model). The null hypothesis for this test is that the smaller model is the true model, a large test statistics indicate that the null hypothesis is false. While all three tests address the same basic question, they are slightly different.

We recall that fitting the PL2 model implies choosing the split point (position of the rank) in which the vase is switched. Given m labels, there are $(m-1)$ possible split points. We try them all and select the one with the highest log-likelihood. We then apply the LRT to select between PL1 and PL2 model.

4.2.1 Model Selection

Suppose the two alternative models are under consideration, PL1 model is simpler then PL2 model. Likelihood ratio test (LRT) is performed by estimating the two models and comparing the fit of one model to the fit of the other. Adding parameters in a model will almost always make the model fit more (i.e., PL2 will have a higher log likelihood), but it is necessary to test whether the observed difference in model fit is statistically significant. LRT does this by comparing the log likelihoods of the two models, if this difference is statistically significant, then less restrictive model (the one with more variables, PL2 model) is said to fit the data significantly better than the more restrictive model. In other words, LRT performs the statistical test by considering if there is evidence of the need to move from PL1 model to PL2 model. If one has the log likelihoods from the models, LRT is fairly easy to calculate. The formula for the LRT statistic is:

$$LR = -2ln(\frac{ll_1}{ll_2}) = 2(ll_2 - ll_1) \tag{4.12}$$

where ll_1 the log-likelihood for the PL1 model and ll_2 the log-likelihood for the PL2 model. The test statistic is chi-squared distribution and the degrees of freedom (dof) is number of parameters estimates which is equivalent to the number of labels to rank. Once we know the LR value (chi-squared distribution), with the number of degrees of freedom we can now use a table or some other method to find the associated $p-value$, indicating that the PL2 model fits significantly (significantly level $= \alpha$) better than the PL1 model.

Algorithm 2: Model Selection

Input: ll_1, ll_2, dof, α
Output: h : rejection decision, if $h = 1$ then select $PL2$

$LR \leftarrow 2 * |ll_2 - ll_1|$
$P \leftarrow chisquare(LR, dof)$
; // chisquare() is chi-squared distribution with dof degree of freedom
if $P < \alpha$ **then**
  $h = 1$
else
  $h = 0$

4.3 Chapter Summary

In this chapter, we have introduced the PL2 model for label ranking. PL2 is useful to model label ranking in which the ratio of the preferences change depending on the position of the

rank (e.g., example of football supporters we discussed at the beginning of this chapter). We have shown how to obtain estimated parameters and the most probable rank for the PL2 model. We have also introduced the PLH model, which performs statistical tests to select the best fit model between PL1 and PL2 model.

Chapter 5

Testing the Parameters Estimation

In this chapter, we verify that the parameter estimation is working properly by generating synthetic data from a vase distribution with known parameters, and then try to infer the parameters.

This chapter is organized as follows: Section 5.1 introduces artificial datasets generated for testing the parameters estimation obtained by PL1 and PL2 model. In Section 5.2, we discuss some approaches (MAE, Kendall's tau and Spearman's rank) to evaluate the performance between estimated and actual parameters. We describe the process to select the best fit model between PL1 and PL2 in Section 5.3. Experiments generating artificial datasets from the 1-vase model and the 2-vases model are presented in Section 5.4 and 5.5 respectively. We discuss the time complexity of finding parameters that fit the models in Section 5.6. Section 5.7 is the summary of this chapter.

5.1 Artificial Dataset

The datasets are created in two different ways using 1-vase or 2-vases model. The 1-vase artificial dataset is generated from one vase distribution and the 2-vases artificial dataset is generated from two vases distribution. We are unaware of previous results regarding the parameter estimation of the PL2 model.

5.1.1 Creating 1-Vase Artificial Dataset

Consider a vase V containing the random positive numbers $v_i, i = 1, ..., m$ where v_i corresponds to label i. The larger the parameter, the more preferred is the label. The $vi's$ are proportional to the probability that label i is ranked first. Without loss of generality, normalize the v_i so that they sum up to 1.

To build a rank, at each stage we are drawing a label from the vase. Label i is drawn with probability v_i. We randomly draw the first label which gives the first element of the rank o_1. At the second stage, draw another label. If it is different from the already drawn o_1, it continues the second position of the rank (o_2). The third stage - keep drawing until obtaining a label distinct from o_1 and o_2, and assign it to o_3. The remaining stages follow, until the m^{th} stage, where there is only one label left. It is ranked last.

Algorithm 3: Creating 1-Vase Artificial Dataset

Input: $nLabels$: number of labels, $nInstances$: number of instances
Output: $ranks$: ranks which the size of $nInstances * nLabels$ matrix

```
init ranks
init v
for i : 1 to nInstances do
    o ← v
    o_cum ← cumulativeSum(v)
    for j : 1 to nLabels − 1 do
        draw ← random(0, 1)
        index ← which(o_cum >= draw)
        ranks[i, j] ← index[1]
        index_new ← setdiff(which(v > 0), ranks[i, ])
        if length(index_new) == 1 then
            ranks[i, nLabels] ← index_new
        else
            o_new ← o/sum(o[index_new])
            o_new[ranks[i, 1 : j]] ← 0
            o ← o_new
            o_cum ← cumulativeSum(o)
            o_cum[ranks[i, 1 : j]] ← 0
```

5.1.2 Creating 2-Vases Artificial Dataset

Imagine we have two different vases $V^{(1)}$ and $V^{(2)}$. For each vase, we draw the parameters $v_i^{(1)}, v_i^{(2)}, i = 1, ..., m$ where $v_i^{(1)}$ and $v_i^{(2)}$ are the parameters of label i in each vase. The larger the parameter, the more preferred is the label. The $v_i^{(1)}$ and $v_i^{(2)}$ are proportional to the probability that label i is ranked first in the first and in the second vase respectively. Let L be the split point, where Assume that the first set of parameters $V^{(1)}$ is used to produce the 1^{th} to L^{th} position of the rank, and the second set of parameters $V^{(2)}$ is used to generate the positions from $L+1$ to m.

To build a rank, at each stage we are drawing a label from the two vases. Suppose that the split position is at the second position of the rank between the first vase and second vase ($L = 2$). At the first two stages we sample from the first vase and we draw the first two labels of the rank. The third stage, we switch to the second vase and we sample the remaining positions of the rank. The remaining stages follow, until the m^{th} stage where there is only one label remaining.

Algorithm 4: Creating 2-Vases Artificial Dataset

Input: $nLabels$: number of labels, $nInstances$: number of instances, L: split point
Output: $ranks$: ranks which the size of $nInstances * nLabels$ matrix

```
1  init ranks
2  init v1
3  init v2
4  for i : 1 to nInstances do
5      index ← matrix(0, 1, nLabels)
6      o1 ← v1
7      o2 ← v2
8      o1_cum ← cumulativeSum(v1)
9      o2_cum ← cumulativeSum(v2)
10     for j : 1 to nLabels − 1 do
11         draw ← random(0, 1)
12         if j > L then
13             index = which(o2_cum >= draw)
14             v = v2
           else
15             index = which(o1_cum >= draw)
16             v = v1
17         ranks[i, j] ← index[1]
18         index_new ← setdiff(which(v > 0), ranks[i, ])
19         if length(index_new) == 1 then
20             ranks[i, nLabels] ← index_new
           else
21             o1_new ← o1/sum(o1[index_new])
22             o1_new[ranks[i, 1 : j]] ← 0
23             o1 ← o1_new
24             o1_cum ← cumulativeSum(o1)
25             o1_cum[ranks[i, 1 : j]] ← 0
26             o2_new ← o2/sum(o2[index_new])
27             o2_new[ranks[i, 1 : j]] ← 0
28             o2 ← o2_new
29             o2_cum ← cumulativeSum(o2)
30             o2_cum[ranks[i, 1 : j]] ← 0
```

5.2 Evaluating Error and Measuring Rank Correlation Coefficient between Estimated Parameters and Known Parameters

The mean absolute error (MAE) is used here to evaluate how close the estimated parameters are to the known parameters.

- When we generate ranks from 1-vase artificial dataset, we estimate the parameters of

both the PL1 model and PL2 model. Assume that the estimated parameters of PL1 are $\hat{v} = \{\hat{v}_1, \hat{v}_2, \hat{v}_3\}$. The actual parameters are $v = \{v_1, v_2, v_3\}$. The MAE is the average error of the absolute errors between $\hat{v}$ and v.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{v}_i - v_i| \tag{5.1}$$

where the absolute error is $|\hat{v}_i - v_i|$, $\hat{v}_i$ is the estimate and v_i is the actual.

- When we generate ranks from 2-vases artificial dataset, we estimate the parameters of the first vase and the second vase of PL2 model. Eventually, we average the MAE of the two vases.

$$MAE_{vase1} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{v}_i^{(1)} - v_i^{(1)}\right| \tag{5.2}$$

$$MAE_{vase2} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{v}_i^{(2)} - v_i^{(2)}\right| \tag{5.3}$$

where the absolute error of the first vase and the second vase is $\left|\hat{v}_i^{(1)} - v_i^{(1)}\right|$ and $\left|\hat{v}_i^{(2)} - v_i^{(2)}\right|$ respectively.

A rank correlation coefficient assess the degree of similarity between two sets of rankings, and can be used to measure the significance of the relation between them. Kendall's tau and Spearman's, we mentioned in section 2.4, are the two popular rank correlation statistics used to measure the correlation between the two ranks.

5.3 Model Selection Between PL1 and PL2

We perform a likelihood ratio test to evaluate if there is evidence of the need for moving from a simple model (PL1) to a more complicated one (PL2). We construct the test statistic using the ratio of the log-likelihood evaluated at the PL2 model parameter estimates and the PL1 model parameter estimates. The rejection decision of the likelihood ratio test conducted at significance level $\alpha = 0.05$. We define the log-likelihood of the PL1 model as $ll1$ and the log-likelihood of the PL2 model as $ll2$. The likelihood-ratio statistic is:

$$LR = 2ll_1 - 2ll_2 \tag{5.4}$$

The probability distribution of the test statistic can be approximated by a chi-square distribution with $df_1 df_2$ degrees of freedom, where df_1 and df_2 are the degrees of freedom of models PL1 and PL2 respectively. In our case, $df_1 = m-1$ and $df_2 = 2(m-1)$ where m is the number of labels to rank. Thus, the degrees of freedom is $m-1$. Once we obtain the deviation of the above equation, we find the associated $p-value$ for $m-1$ degree of freedom, which tells us whether the PL2 model fits significantly better than the PL1 model.

5.4 Experiments Generating Artificial Data from the 1-Vase Model

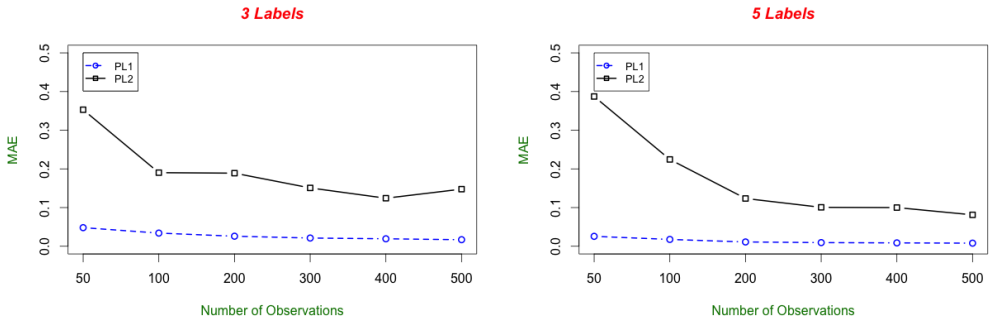
We run 100 experiments and report the result regarding MAE, Kendall's tau, Spearman's rank and log-likelihood. For each experiment, dataset is drawn from known parameters $V = \{v_1, ..., v_m\}$,

where m is the number of labels (3 and 5). Each dataset includes training set with the sample size of 50, 100, 200, 300, 400 and 500 observations, and the test set with the sample size of 250 observations.

For both PL1 and PL2 model, we obtain the most probable rank by sorting the parameters in decreasing order. We measure Kendall's tau and Spearman's rank correlation between the most probable rank retrieved from the two models and the ranks of the instances characterizing the test set. We also compute the log-likelihood of the model prediction on the test set.

5.4.1 MAE and Rank Correlation Coefficient on PL1 Model and PL2 Model

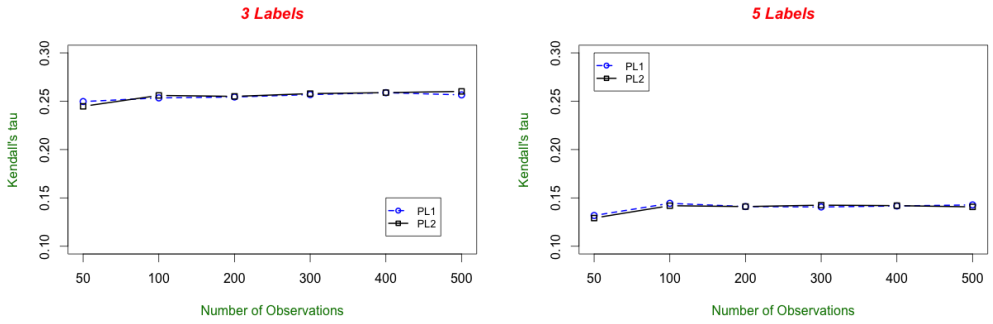
Figure 5.1a and 5.1b show MAE of PL1 and PL2 parameters estimate. MAE decreases as the number of observations increases, as expected, PL1 model has less error because it requires to fit less parameters. The MAE of both models remain stable when the number of observations is large (e.g., more than 400 observations).



(a) 3 labels (b) 5 labels

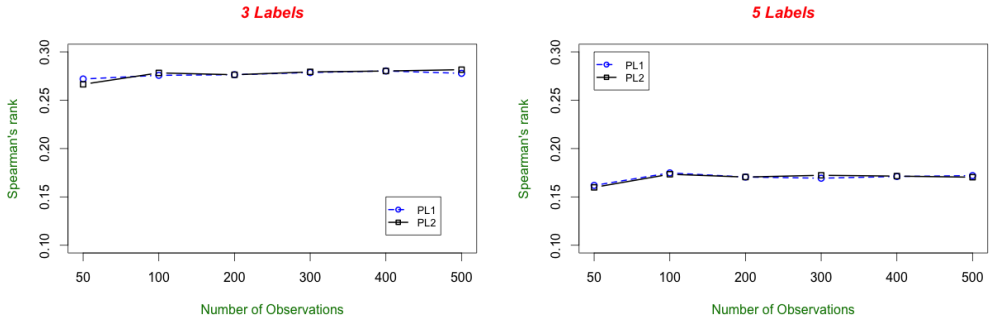
Figure 5.1. MAE between PL1 and PL2 model parameters estimation evaluated on 1-vase artificial dataset, each point is the mean over 100 experiments.

Figure 5.2 and 5.3 show the performances of PL1 and PL2 model in terms of Kendall's tau and Spearman's rank correlation respectively. In general the performances of the two models are comparable. The performances gradually increase with the sample size and stay at the same level when the observations are large.



(a) 3 labels (b) 5 labels

Figure 5.2. Kendall's tau between PL1 and PL2 model parameters estimation evaluated on 1-vase artificial dataset, each point is the mean over 100 experiments.



(a) 3 labels (b) 5 labels

Figure 5.3. Spearman's rank between PL1 and PL2 model parameters estimation evaluated on 1-vase artificial dataset, each point is the mean over 100 experiments.

To sum up, the performance of PL1 and PL2 is comparable in terms of both Kendall's tau and Spearman's rank correlation.

5.4.2 Testing the Calibration of Model Selection

We test the probability of choosing PL2 model over PL1 model when the data are generated using the PL1 model. PL2 should be wrongly chosen instead of PL1 about 5% of the times, since the test is performed setting $\alpha = 0.05$.

Table 5.1 shows the number of times PLH model has selected PL2 model over PL1 model according to the outcome of the likelihood ratio test at the significance level $\alpha = 0.05$. For each case, the number is the result over 100 experiments on different datasets. As expected, PL2 has been chosen by roughly 5% of the times, regardless the sample size.

Table 5.1. The number of times PLH model has selected PL2 model over PL1 model according to the outcome of the likelihood ratio test at the significance levels alpha (= 0.05) . For each case, the number is the result based on 100 generated datasets.

#Sample Size	3 labels #times	4 labels #times	5 labels #times
50	3	9	8
100	3	5	4
200	2	4	7
300	3	5	4
400	2	4	4
500	3	4	2
Avg.#times	2.67	5.17	4.89

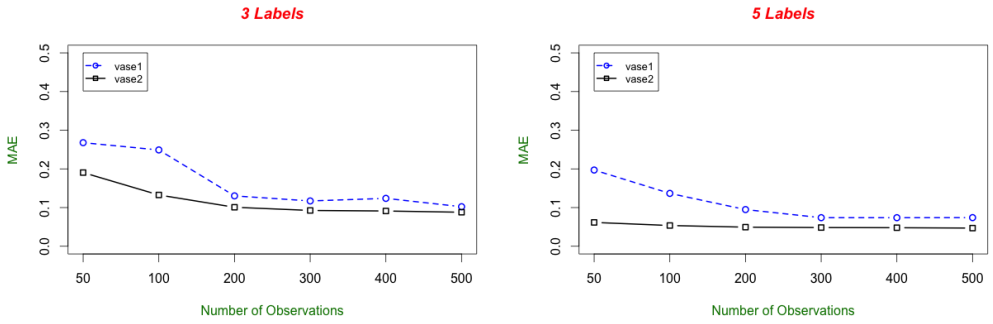
5.5 Experiments Generating Artificial Data from the 2-Vases Model

Given the 2-vase artificial dataset, we run PL1 and PL2 model on the dataset and analyse the performance indicators in relation to MAE, Kendall's tau, Spearman's rank and log-likelihood. The setting of the experiments are the same as we have demonstrated in the 1-vase artificial dataset case.

5.5.1 MAE and Rank Correlation Coefficient on PL1 Model and PL2 Model

Note that we do not access MAE of estimated parameters from PL1, instead we evaluate MAE of known parameters and estimated parameters derived from PL2 model. Figure 5.4 shows MAE of estimated parameters from the first vase and the second vase produced by PL2 model. We performed 100 experiments and reported the average of the result. The number of observations are from 50 to 500 observations and the labels to rank are 3 and 5. The split point L between the first vase and the second vase is set as the first position for the 3 labels case and the second position for the 5 labels case. Clearly, the estimates become more accurate for larger observations. MAE decreases significantly from 50 observations to 200 observations but remains stable from 300 observations onwards.

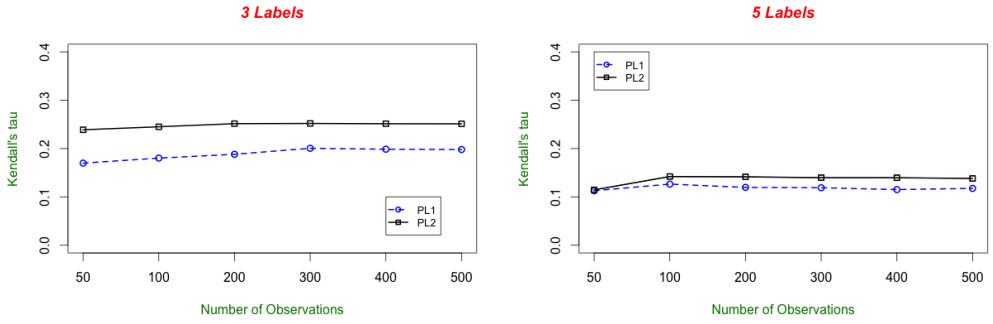
In general, the second vase has constantly lower error than the first vase. This is due to the fact that the split point allows the second vase to contain larger part of the observed rank and thus the model parameters are learned better. For 3 labels case, the split point is the first position between the first and the second vase. This means there is only one parameter to be estimated in the first vase and it makes the estimates less accurate compared to the second vase with a pair of parameters. Similarity, for 4 and 5 labels case, the split point is fixed at the second position between the first and the second vase. Estimating parameters for the first vase is less certain than the second vase. However, the estimates of both vases tend to meet at a point, where the number of observations is large enough. i.e., from 300 observations onwards.



(a) 3 labels (b) 5 labels

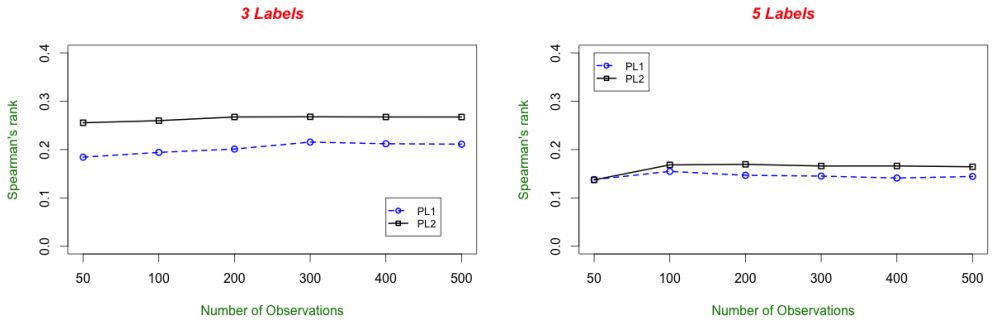
Figure 5.4. MAE between PL1 and PL2 model parameters estimation evaluated on 2-vase artificial dataset, each point is the mean over 100 experiments.

Figure 5.5 and 5.6 show the performances of PL1 and PL2 concerning Kendall's tau and Spearman's rank respectively. The performance indicators slowly increase for small numbers of observations (from 50 to 200 observations) and remains stable for larger observations. Obviously, PL2 has better performances as it is the true model for the generated dataset.



(a) 3 labels (b) 5 labels

Figure 5.5. Kendall's tau between PL1 and PL2 model parameters estimation evaluated on 2-vase artificial dataset, each point is the mean over 100 experiments.



(a) 3 labels (b) 5 labels

Figure 5.6. Spearman's rank between PL1 and PL2 model parameters estimation evaluated on 2-vase artificial dataset, each point is the mean over 100 experiments.

Summing up, if the actual data are generated by PL1, fitting PL1 and PL2 yields (in our experiments) similar predictive performance. If they are generated by PL2, fitting the PL2 model yields higher performance than the PL1 model.

5.5.2 Testing the Calibration of Model Selection

In this section, we perform the same statistic test as it has done in section 5.5.2. The test is performed setting $\alpha = 0.05$. However, the data are generated using the PL2 model instead of PL1 model. Thus, PL1 should be wrongly chosen instead of PL2 about 5% of the times. In other words, PL2 should be correctly selected roughly 95% of the times.

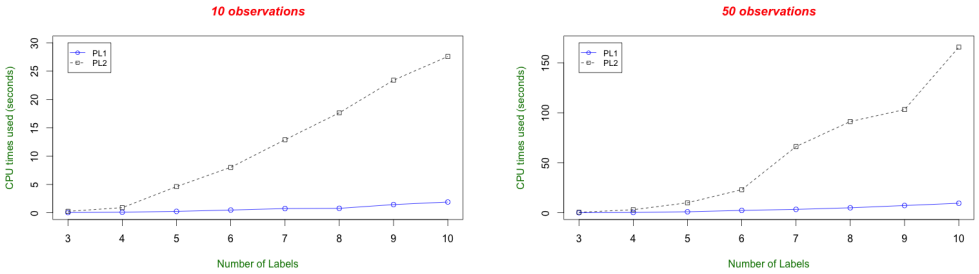
Table 5.2 shows the number of times PLH model has selected PL2 model over PL1 model according to the outcome of the likelihood ratio test at the significance levels $\alpha = 0.05$. For each case, the number is the result based on 100 experiments on different datasets. PLH model has selected PL2 model about 95% of the times.

Table 5.2. The number of times PLH model has selected PL2 model over PL1 model according to the outcome of the likelihood ratio test at the significance levels alpha (= 0.05) . For each case, the number is the result based on 100 generated datasets.

#Sample Size	3 labels #times	4 labels #times	5 labels #times
50	73	90	8
100	82	93	4
200	92	96	7
300	93	100	4
400	94	100	4
500	97	100	2
Avg.#times	88.5	96.5	99.67

5.6 Time Complexity

We demonstrate the performance of PL1 and PL2 model in finding the maximum likelihood estimate for the PL1 and PL2 model. The two models use Newton-Raphson algorithm for the maximum likelihood estimation. The sample size consisted of 10 and 50 observations, for each sample size we consider between 3 and 10 labels. R command (system.time) was used to find the execution time of the algorithm. The learning time is roughly linear in the number of labels for model PL1. PL2 has roughly quadratic complexity since the computational time goes as $O(n * n)$. PL2 has to run twice the fitting algorithm (2 vases) for each possible split point. Given m labels there are $(m-1)$ possible split points. The plots are roughly in line with these considerations.



(a) 10 observations (b) 50 observations

Figure 5.7. The performance of PL1 and PL2 model in finding the maximum likelihood estimator for the Plackett-Luce model. x-axis is the number of labels and y-axis is the CPU times used in seconds.

5.7 Chapter Summary

In this chapter, we have shown that PL2 model is capable of estimating parameters, given data generated from 1-vase and 2-vases model distribution. The performance indicators (MAE, Kendall's tau and Spearman's rank) have proved that PL2 model is comparable with PL1 model for estimating parameters of 1-vase artificial datasets. For 2-vases artificial datasets, fitting PL2 model results higher performance than PL1 model. The time complexity of estimating parameters is roughly linear in the number of labels for model PL1, whilst PL2 has roughly quadratic complexity since the computational time goes as $O(n^2)$.

Chapter 6

Experiments and Results

In this chapter, we compare PL1, PL2 and PLM on different datasets (see Section 6.1 and Section 6.2). We present an evaluation of the instance-based label ranking framework as introduced in Chapter 2. We define the instance-based label ranking with one vase, two vases and hybrid model as IB-PL1, IB-PL2 and IB-PLH respectively. For the instance-based label ranking, the neighborhood size $k \in 5m, 10m$ is selected through cross validation on the training set, where m is the number of labels to rank. We use the Euclidean distance after normalizing the features as a distance measure in the instance space. This chapter ends with conclusions in Section 6.3.

6.1 Ranking Nascar Racing Drivers

The Nascar dataset (2002 US Nascar season) was a case study of fitting the P-L model performed by Hunter (2004). The dataset consisted of 36 races in which a total of 87 different drivers participated, and each race involved only 43 drivers. Some drivers attended only part of the races. We thus consider the reduced dataset consisting of 27 drivers who entered all 36 races.

Table 6.1 shows the top and bottom 10 drivers as ordered by average place, as well as their rank assigned by both PL1 and PL2 model. Regarding PL2 model in which the lag point moves from the first to the second vase, we selected the one that maximizes the log-likelihood. The split position at the seventh position is the one with the highest log-likelihood meaning the ranks after the seventh one may be much more uniform than the first seven ranks. There are some clear differences between the two models. PL1 model places Mark Martin in the first - this certainly ties in with his very high average place whilst PL2 model puts Tony Stewart in the first place. The top 7 drivers are placed in different orders, this is perhaps to be as expected, each vase in PL2 has its own set of proportions of the ranks, and the parameters from the first vase (first rank to seventh rank) are very high compared to the parameters from the second vase. Notice that after the split position, neither PL1 or PL2 are equivalent to simply ordering by average place. Likewise, toward the bottom of the table the two models place Mike Skinner and Ward Burton at the very bottom.

To compare the fit of the two models, we compute the statistic test based on the likelihood ratio. PL1 has 27 parameters (number of drivers) and a log-likelihood of -2279.5. PL2 has 54 parameters (twice PL1 number of parameters) and a log-likelihood of -2198.797. Then the probability of this difference is that of chi-squared value of $2 * (2279.5 - 2198.797) = 161.42$ with $m - 1 = 27 - 1 = 26$ degrees of freedom. The deviation of the chi-squared value exceeds

the critical value and thus the PL1 is rejected and PL2 is selected. In other words, PL2 model is more suitable for the Nascar dataset than PL1 model.

Generally, PL2 model is useful when we intend to have different proportions of the set of ranks, but the vase drawn from at each stage depends only on the stage, not on which labels have already been drawn (Marden 1995). This model is suitable if users' opinions change over time. For example, users are asked to rate their favorite movies from 100 movies, it might be that the users pay more attention to rank their top 10, but after that the users are less careful. Then the rank after the tenth one may be much more uniform than the first 10.

Table 6.1. Rankings for top and bottom ten 2002 Nascar drivers, as given by average place. The parameters have been normalized to sum to 1 for both PL1 and PL2 so that they are comparable.

Drivers	Avg.place	PL1 rank	PL1 parameters	PL2 rank	PL2 parameters
Mark Martin	9.2778	1	0.0604	7	0.0759
Tony Stewart	9.4167	4	0.0516	1	0.0816
Jimmie Johnson	10.1944	2	0.0555	6	0.0762
Kurt Busch	10.5278	9	0.0437	3	0.0782
Jeff Gordon	10.5556	5	0.0479	2	0.0803
Rusty Wallance	10.6389	3	0.0549	8	0.0352
Ryan Newman	10.7222	8	0.0437	4	0.0781
Matt Kenseth	11.30556	7	0.0451	5	0.0778
Dale Jarrett	11.5	10	0.0434	21	0.0204
Ricky Rudd	12.25	12	0.0421	13	0.0259
.					
.					
.					
Dave Blaney	16.0556	16	0.0357	9	0.0298
Robby Gordon	16.0556	19	0.0319	17	0.0244
Kyle Petty	16.6389	20	0.0305	12	0.0263
Ward Burton	16.9167	27	0.0144	27	0.0073
Terry Labonte	17.1111	22	0.0241	23	0.0186
Elliott Sadler	17.1667	23	0.0222	25	0.0163
Jeremy Mayfield	17.6389	21	0.0254	20	0.0206
John Andretti	18.3611	24	0.0213	22	0.0196
Ken Schrader	19.3611	25	0.0201	24	0.0181
Mike Skinner	20.2222	26	0.0165	26	0.0146

6.2 Label Ranking Datasets

6.2.1 Datasets

In this section, we consider different datasets from the KEBI[1] Data Repository hosted by the Philipps University of Marburg. These datasets, which commonly used for label ranking problems, are presented in Table 6.2. The selected datasets are from UCI repository and Statlog

[1] https://www.uni-marburg.de/fb12/kebi/research/repository/labelrankingdata

collection are amended and transformed to label ranking datasets, which have been done in two different approaches. For classification datasets, a naive Bayes classifier is used to train the complete dataset and for each instance, all the labels are ordered with respect to the predicted class probabilities. In the case of tied data, the labels with lower index are ranked first. For regression datasets, predicate number of numerical attributes is removed from the set of predictors, and each of them is considered as a label. A complete ranking is obtained by standardizing the attributes and ordering it with respect to the size.

Table 6.2. Datasets and their properties (the type refers to C:classification datasets or R:regression datasets).

Dataset	type	#inst.	#attr.	#labels
authorship	C	841	70	4
calhousing	R	20640	4	4
cpu-small	R	8192	6	5
elevators	R	16599	9	9
fried	R	40769	9	5
glass	C	214	9	6
housing	R	506	6	6
iris	C	150	4	3
segment	C	2310	18	7
stock	R	950	5	5
vehicle	C	846	18	4
vowel	C	528	18	11
wine	C	178	13	3

6.2.2 Results of Instance-Based Label Ranking Approach

We perform 10 runs of 10-folds cross validation on the various datasets comparing the instance-based one-vase model (IB-PL1), instance-based two-vases model (IB-PL2) and instance-based hybrid model (IB-PLH). The IB-PLH is the model which implements statistical model selection between the IB-PL1 and IB-PL2 model. The IB-PL2 is preferred over the IB-PL1 if the difference of the log-likelihood is significant enough. To implement cross-validaion, the dataset is first sorted according to their class labels and one random column is added. Then the dataset is sorted again with respect to the random column, and the sorted dataset is divided into 10 parts leaving one portion for the test data and the remaining set for the train data. This will give us the cross validation randomness and satisfy that the classes are equal stratification in the different folds. The Euclidean distance (after standardized the attributes) is used as a distance measure for both IB-PL1 and IB-PL2. Given number of n labels to rank, the neighborhood size K are $5n$ and $10n$. The performances are indicated in terms of Kendall's tau, Spearman's rank and log-likelihood.

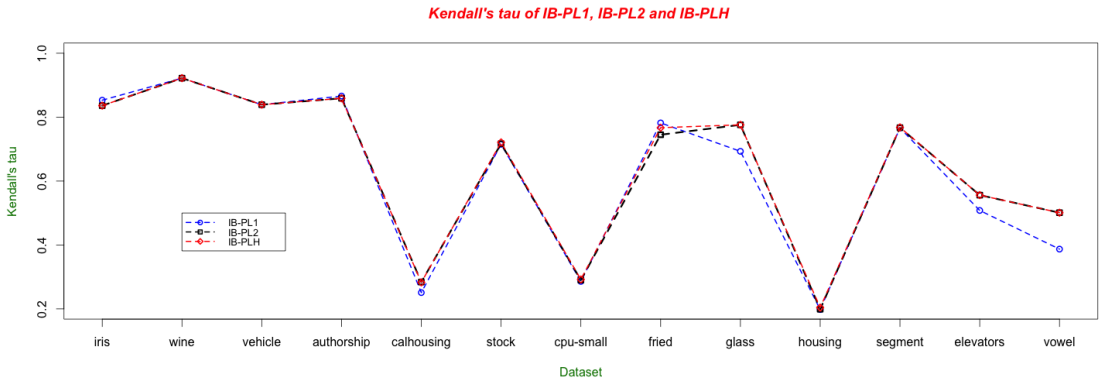


Figure 6.1. Kendall's tau of IB-PL, IB-PL2 and IB-PLH on each dataset.

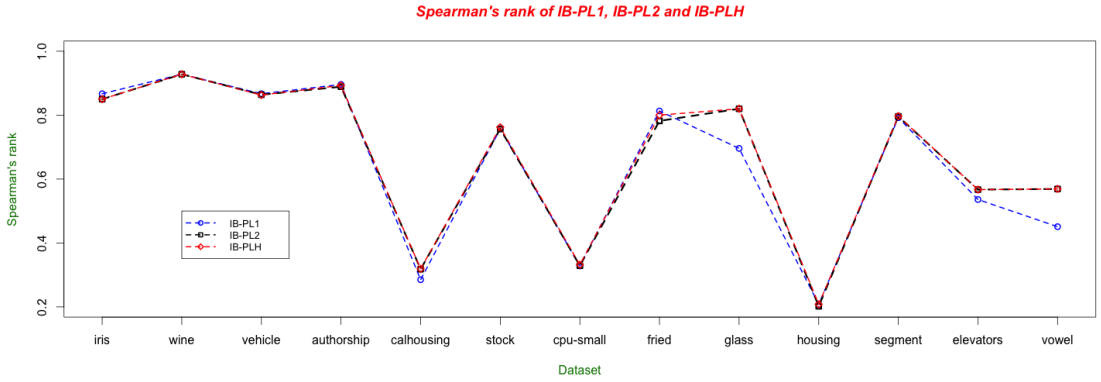


Figure 6.2. Spearman's rank performances of IB-PL, IB-PL2 and IB-PLH on each dataset.

Table 6.3. Performance of the label ranking methods in terms of Kendall's tau (in brackets the rank). On each dataset the best-performing model is given rank 1. Higher ranks (1 is the highest) correspond to better models.

Dataset	IB-PL1	IB-PL2	IB-PLH
iris	0.853(1)	0.836(2)	0.836(2)
wine	0.922(1)	0.922(1)	0.922(1)
vehicle	0.839(1)	0.839(1)	0.839(1)
authorship	0.866(1)	0.859(3)	0.863(2)
calhousing	0.249(3)	0.282(2)	0.283(1)
stock	0.715(3)	0.717(2)	0.721(1)
cpu-small	0.286(3)	0.290(2)	0.292(1)
fried	0.782(1)	0.745(3)	0.767(2)
glass	0.693(2)	0.776(1)	0.776(1)
housing	0.198(3)	0.199(2)	0.204(1)
segment	0.766(3)	0.767(2)	0.768(1)
elevators	0.508(2)	0.556(1)	0.556(1)
vowel	0.387(3)	0.501(1)	0.501(1)
Avg.Rank	2.271	1.909	1.272

Table 6.4. Performance of the label ranking methods in terms of Spearman's rank (in brackets the rank). On each dataset the best-performing model is given rank 1. Higher ranks (1 is the highest) correspond to better models.

Dataset	IB-PL1	IB-PL2	IB-PLH
iris	0.867(1)	0.850(2)	0.850(2)
wine	0.928(1)	0.928(1)	0.928(1)
vehicle	0.867(1)	0.864(2)	0.864(2)
authorship	0.896(1)	0.889(3)	0.893(2)
calhousing	0.285(3)	0.318(2)	0.319(1)
stock	0.757(2)	0.757(2)	0.762(1)
cpu-small	0.328(3)	0.329(2)	0.332(1)
fried	0.813(1)	0.782(3)	0.800(2)
glass	0.696(2)	0.820(1)	0.820(1)
housing	0.208(1)	0.202(2)	0.208(1)
segment	0.792(3)	0.796(2)	0.797(1)
elevators	0.536(2)	0.567(1)	0.567(1)
vowel	0.451(2)	0.569(1)	0.569(1)
Avg.Rank	1.833	1.917	1.333

Table 6.3 and 6.4 shows that IB-PLH model has the best performance concerning Kendall's tau and Spearman's rank as we can notice the Avg. Rank. It is clear that combining the IB-PL1 and IB-PL2 through the likelihood ratio test is better than both IB-PL1 and IB-PL2 alone. In the case of tied data which occurs often in wine dataset, the three models report the same performance indicators. The indicators are slightly different when we tried running the experiments with other number of neighborhoods (not K=5n nor K=10n).

Table 6.5 shows that IB-PL2 produces better performance than IB-PL1 for dataset with large number of labels to rank (e.g number of labels is greater than four). The average log-likelihood shows how fit each model is for each dataset.

Figure 6.3. shows the difference between LL1 and LL2 as a function of the number of labels. Let LL1 denotes the log-likelihood of IB-PL1 and LL2 denotes the log-likelihood of IB-PL2. The result demonstrates that the difference between the log-likelihood of the two models consistently grows as the number of labels increases. However, we observe the slightly difference in the fried dataset and once we looked at this dataset in detail, we notice many duplicated rank order embedded in the selected K-neighborhoods instances meaning this is tied data and thus this dataset does not meet our assumption. Table 3 has also shown the outcome that is relevant to the intuition.

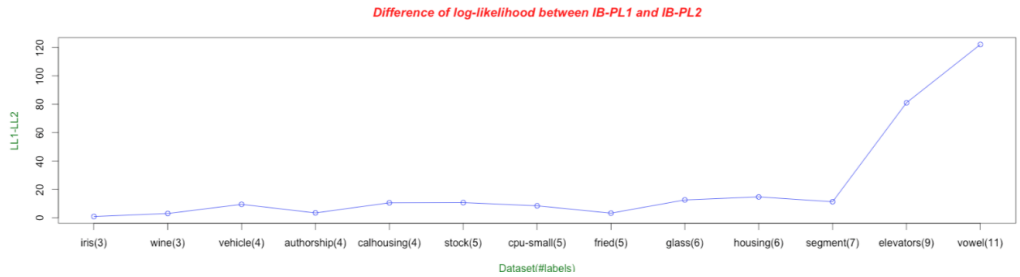


Figure 6.3. The difference of log-likelihood between IB-PL1 and IB-PL2 as the function of number of labels.

Table 6.5. Average log-likelihood of both methods and the H-value returned from the likelihood ratio test. If H-value is equal to 1, the IB-PL2 is selected.

Dataset	IB-PL1	IB-PL2	H-value
iris	-10.870	-10.039	0
wine	-11.982	-9.031	0
vehicle	-36.339	-26.931	1
authorship	-13.848	-12.620	0
calhousing	-55.321	-44.823	1
stock	-39.024	-28.388	1
cpu-small	-93.667	-85.289	1
fried	-64.517	-61.312	0
glass	-50.108	-37.584	1
housing	-128.437	-113.799	1
segment	-66.212	-54.971	1
elevators	-388.995	-307.964	1
vowel	-793.615	-671.5	1

Table 6.6. The number of K-neighbors (K=5n) produces the best performance for IB-PL2 on each dataset. n is number of labels to rank. The last column the selected model according to the result of likelihood ratio test.

Dataset	IB-PL1	IB-PL2	Selected Model
iris	5n	5n	IB-PL1
wine	10n	10n	IB-PL1
vehicle	5n	5n	IB-PL2
authorship	10n	5n	IB-PL1
calhousing	10n	5n	IB-PL2
stock	5n	5n	IB-PL2
cpu-small	5n	5n	IB-PL2
fried	10n	10n	IB-PL1
glass	10n	5n	IB-PL2
housing	5n	5n	IB-PL2
segment	5n	5n	IB-PL2
elevators	5n	5n	IB-PL2
vowel	5n	5n	IB-PL2

As expected, using different number of K neighbors leads to different results. The interesting point is the IB-PL2 always perform well with a smaller number of K, where k is equal to 5n. IB-PL2 tends to fit well with small number of observations as larger observations may be cause the overfitting problem. Thus, choosing a proper number of K affects significant performances for the models. In most cases, K=5n is suggested to avoid the overfitting issue.

6.3 Chapter Summary

We have shown the powerful of PL2 model by running it on real-world datasets (Nascar ranking dataset and KEBI label ranking datasets). We have developed PLH model which implements statistical model selection to choose between PL1 and PL2. We have shown that PLH model is persistently better performing than both PL1 and PL2 model. The big improvement over the traditional PL1 model is observable when dataset includes large numbers of labels.

Chapter 7

PLRank: An R Package for Label Ranking

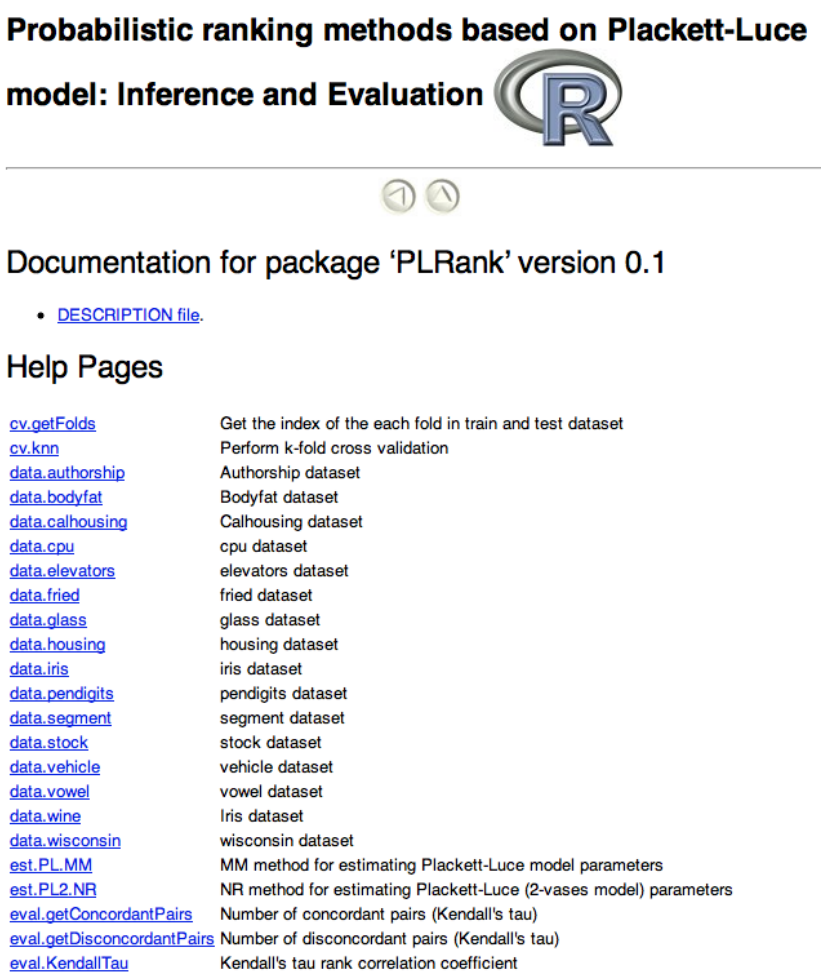
7.1 PLRank overview

In the this chapter, we describe the R package PLRank[1] which is a collection of general purpose functions that provide a flexible set of tools for applying a wide range of label ranking methods based on Plackett-Luce model. We have extended an existing method (pl.r) of fitting Plackett-Luce model from PMR[2] package. The PLRank package includes the new algorithms we developed in Chapter 4, the evaluation methods and datasets used in Chapter 5 and 6 of this thesis. Label ranking datasets in the PLRank consist of training examples of a target function that has multiple binary target variables. In other words, each item of a label ranking dataset is annotated by many labels (classes). This is the nature of many real world problems such as web page categorization, music categorization, direct marketing, and etc. The typical usage scenario of PLRank would involve a machine learning researcher performing an empirical evaluation of one or more label ranking learning algorithms, based on one or more label ranking datasets, and a machine learning practitioner building a label ranking model using a training dataset and then applying it to a new (unlabeled) dataset, in order to obtain predictions.

Currently the PLRank package includes algorithms for performing major label ranking tasks and sample of datasets:

- Label ranking datasets, KEBI[3] Data Repository hosted by the Philipps University of Marburg. The properties of the datasets are mentioned in Table 6.2.
- Artificial ranking datasets. The functions to generate ranking datasets.
- Maximum likelihood estimation methods to estimate the parameters of PL1 and PL2 model. Two MLE methods are available, MM and NR.
- Evaluation methods for label ranking problems. Calculate a variety of evaluation measures through k-folds cross validation.

[1] https://github.com/toppu/PLRank
[2] http://cran.r-project.org/web/packages/pmr/index.html
[3] https://www.uni-marburg.de/fb12/kebi/research/repository/labelrankingdata



Probabilistic ranking methods based on Plackett-Luce model: Inference and Evaluation

Documentation for package 'PLRank' version 0.1

- DESCRIPTION file.

Help Pages

cv.getFolds	Get the index of the each fold in train and test dataset
cv.knn	Perform k-fold cross validation
data.authorship	Authorship dataset
data.bodyfat	Bodyfat dataset
data.calhousing	Calhousing dataset
data.cpu	cpu dataset
data.elevators	elevators dataset
data.fried	fried dataset
data.glass	glass dataset
data.housing	housing dataset
data.iris	iris dataset
data.pendigits	pendigits dataset
data.segment	segment dataset
data.stock	stock dataset
data.vehicle	vehicle dataset
data.vowel	vowel dataset
data.wine	Iris dataset
data.wisconsin	wisconsin dataset
est.PL.MM	MM method for estimating Plackett-Luce model parameters
est.PL2.NR	NR method for estimating Plackett-Luce (2-vases model) parameters
eval.getConcordantPairs	Number of concordant pairs (Kendall's tau)
eval.getDisconcordantPairs	Number of disconcordant pairs (Kendall's tau)
eval.KendallTau	Kendall's tau rank correlation coefficient

Figure 7.1. Screenshot of PLRank.

7.2 Using PLRank

Many examples concerning optimization tasks are provided in this section. In particular, we will present the optimization of well-known benchmark mathematical functions and label ranking problems in general. Hereafter, we assume that the PLRank package is already installed and loaded in the current R session, for example by entering the following command:

```
R> library("PLRank")
```

7.2.1 Function Generating 1-Vase Artificial Dataset

We start by creating an artificail dataset from PL1 model. This is useful for performing inference tasks for ranking problems. Figure 7.2, $\$rank$ is the rankings randomly drawn from the known parameters $\$para$

```
#Usage
genRank1v(nLabels, nObs)

#Arguments
nLabels number of labels to rank
nObs number of observations

#Examples
Create artificial dataset with 4 labels to rank and 10 instances
R> lables = 4
R> observations = 10
R> genRank1v(lables, observations)
```

```
$para
[1] 0.30365754 0.45304143 0.14735316 0.09594787

$rank
      [,1] [,2] [,3] [,4]
 [1,]    2    4    1    3
 [2,]    2    4    1    3
 [3,]    2    1    4    3
 [4,]    2    1    4    3
 [5,]    3    1    2    4
 [6,]    1    2    3    4
 [7,]    1    2    4    3
 [8,]    1    3    4    2
 [9,]    2    3    4    1
[10,]    2    1    3    4
```

Figure 7.2. Sample of 1-vase artificial dataset.

7.2.2 Function Generating 2-vase Artificial Dataset

Creating an artificial dataset from PL2 model.Figure 7.3, $\$rank$ is the rankings randomly drawn from the known parameters $\$para1$ and $\$para2$, where the split point is defined at the second position.

```
#Usage
genRank2v(nLabels, nObs, L)

#Arguments
nLabels number of labels to rank
nObs number of observations
L split position between the first vase and the second vase

#Examples
Create artificial dataset with 4 labels to rank and 10 instances
R> lables = 4
R> observations = 10
```

```
R> L = 2
R> genRank2v(lables, observations, L)
```

```
$para1
[1] 0.1420982 0.2250002 0.4075334 0.2253682

$para2
[1] 0.16347150 0.02948138 0.26514733 0.54189979

$rank
      [,1] [,2] [,3] [,4]
 [1,]    1    3    4    2
 [2,]    3    4    1    2
 [3,]    3    2    4    1
 [4,]    2    3    1    4
 [5,]    1    3    4    2
 [6,]    4    2    3    1
 [7,]    1    2    4    3
 [8,]    4    3    1    2
 [9,]    4    3    1    2
[10,]    3    4    1    2
```

Figure 7.3. Sample of 2-vase artificial dataset.

7.2.3 MM Method for Estimating One-Vase Plackett-Luce Model Parameters

This function uses the MM (minorization-maximization) algorithms to fit PL1 model. The method was rewritten from the original Matlab code (plackmm) provided by [Hunter, 2004]. The method was modified by adding a stop criteria to prevent the estimates from approaching infinity. Figure 7.4 shows MM method for estimating Plackett-Luce model parameters, given a set of label rankings $\{L1, L2, L3\}$ and the ranks:

$$L1 > L2 > L3$$

$$L3 > L1 > L2$$

```
#Usage
est.PL.MM(dset)

#Arguments
dset ranking dataset

# Examples
Create 2 ranks and 3 labels
R> ranks = matrix(0,2,3) # rows:number of ranks, column:number of labels
R> ranks[1,] = c(1,2,3) # L1>L2>L3
R> ranks[2,] = c(3,1,2) # L3>L1>L2
R> est.PL.MM(ranks)
```

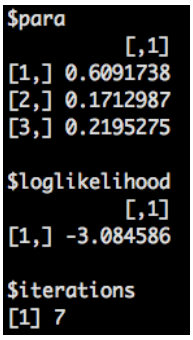


Figure 7.4. Parameters estimation for Plackett-Luce model using MM method.

From the estimated parameters, we can conclude that label 1 is ranked first as it is ranked the first place once and the second place one. While label 2 and label 3, it is likely that label 3 has more probability to rank before label 3.

7.2.4 NR Method for Estimating Two-Vases Plackett-Luce Model Parameters

This function uses the NR algorithm to fit PL2 model. The function extends the original method fitting the one-vase Plackett-Luce model from PMR package by adding the two-vases model interpretation. The work was contributed by Shuai Fu[4] from the Swiss AI lab IDSIA[5].

```
#Usage
est.PL2.NR(dset, L)

#Arguments
dset ranking dataset
L split point between the first vase and the second vase

#Value
Estimated parameters and log-likelihood
Create 2 ranks and 3 labels
R> ranks = matrix(0,2,3) # rows:number of ranks, column:number of labels
R> ranks[1,] = c(1,2,3) # L1>L2>L3
R> ranks[2,] = c(3,1,2) # L3>L1>L2
R> est.PL2.NR(ranks,1)
```

7.2.5 Evaluating Predictive Models On Label Ranking Dataset

This section illustrates the use of the *cv.knn*() function on two real datasets. The first one, Iris, is a well-known dataset in ranking study, which consists of measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for each flowers from each of 3 species (3 labels).

[4] http://people.idsia.ch/ shuai.fu
[5] http://www.idsia.ch

The second one, Eurovision, presented in Jacques and Biernacki (2012), consists of the votes of European countries during the Euro- vision Song Contest. Both datasets are available in the Rankcluster package, as well as the three other datasets analysed and described in Jacques and Biernacki (2012): APA, quiz, sports.

Label ranking dataset consists of two main parts, the features and ranks. The latter forms a set of preference labels. At the moment, each instance only contains a complete ranking. Particularly, labels are assembled to a labek ranking by using the ">" sign. For example, L1 > L2 means that label 1 is preferred over label 2. It is not allowed to state one single label inside of an instance and thus there have to be at least two labels separated by the ">" character. The label ranking dataset will be automatically avaialble in the R environment once the PLRank package is loaded into the R session. For example, we can simply call the iris dataset by trying $R > data.iris$ in the R console. Figure 7.5 shows the iris dataset, which has been amended and transformed to a label ranking dataset.

	V1	V2	V3	V4	V5
1	-5.55556e-01	0.2500000	-0.8644070	-9.16667e-01	L1>L2>L3
2	-6.66667e-01	-0.1666670	-0.8644070	-9.16667e-01	L1>L2>L3
3	-7.77778e-01	0.0000000	-0.8983050	-9.16667e-01	L1>L2>L3
4	-8.33333e-01	-0.0833334	-0.8305080	-9.16667e-01	L1>L2>L3
5	-6.11111e-01	0.3333330	-0.8644070	-9.16667e-01	L1>L2>L3
6	-3.88889e-01	0.5833330	-0.7627120	-7.50000e-01	L1>L2>L3
7	-8.33333e-01	0.1666670	-0.8644070	-8.33333e-01	L1>L2>L3
8	-6.11111e-01	0.1666670	-0.8305080	-9.16667e-01	L1>L2>L3
9	-9.44444e-01	-0.2500000	-0.8644070	-9.16667e-01	L1>L2>L3
10	-6.66667e-01	-0.0833334	-0.8305080	-1.00000e+00	L1>L2>L3

Figure 7.5. Sample of label ranking dataset. There are 4 features and 3 labels $\{L1, L2, L3\}$

Learning the parameters of a prediction function and testing iton the same data is a methodological mistake: a model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet unseen data. This situation is called overfitting. To avoid this, it is common pratice when performing the label ranking learning to hold out part of the available data as a test set. We will work with a couple of datasets and demonstrate which model (PL1, PL2 or PLH) is the best fit for each dataset.

```
#Usage
cv.knn(dset, k = 10, nFolds = 10, nRuns = 10, alpha = 0.05)

#Arguments
dsetlabel ranking dataset
k the number of nearest neighbors to search. The default value is set to 10
nFolds  the number of folds for the cross validation
nRuns the number of runs for the cross validation
alpha the default value is set to 0.05

#Examples
Perform the 10 runs of 10-folds cross validation on the Iris dataset
dset = data.iris
cv.knn(dset, k=15, nFolds=10, nRuns=10, alpha=0.05)
```

	IBPL1	IBPL2	IBPLH
Kendall's tau	0.9296296	0.9222222	0.9222222
Spearman's rho	0.9361111	0.9277778	0.9277778
Log Likelihood	-19.2044862	-14.4274801	-14.7108358

Figure 7.6. 10 runs of 10-folds cross validation on Iris dataset.

Figure 7.6 shows the result of 10-folds cross validation on the Iris label ranking dataset. We can conclude that IBPL1 has the best performance in terms of Kendall's tau and Spearman's rank correlation coefficient.

7.3 Chapter Summary

In this chapter, we have presented the PLRank R-package for analysing label ranking data. The package provides parameters estimation through MM and NR algorithm fitting the PL model both with one vase and two vases interpretation. Evaluation methods for the predictive model such as likelihood ratio test, Kendall's tau, Spearman's rank and k-flods cross validation is also available for users to perform the statistical test.

Chapter 8

Conclusions

8.1 Summary

We have developed an extension of the Placekett-Luce model. It relaxes by having two vases while the traditional model has only one vase. We have shown that this can accurately learn parameters from data generated from a known vase distribution. We have shown the scalability of PL2 model by running it on real-world datasets, the performance of the mixtures of the PL1 and PL2 model. PLH model implements statistical model selection to choose between PL1 and PL2. It is constantly better performing than both PL1 and PL2. The improvement over the traditional PL1 model is especially strong when dataset includes larger numbers of labels to rank. We have implemented R package (PLRank) for label ranking methods based on Plackett-Luce model.

8.2 Limitations

There are a few limitations to the work presented in this thesis. The first is the limited size of the dataset used in the experiment. Some datasets are represented with a small number of observations and a large number of items to rank, or each individual observation may rank only a few of the total labels. This makes difficulty for MLE to estimate parameters on this types of datasets. In some cases, the MLE cannot even find the optimal solution and lead to the infinity estimates such as datasets, where a label is always ranked first or last.

Another identified limitation is the PL2 model cannot handle incomplete ranks. PL1 model generally deals with incomplete ranks in which the data should be missing at random. For the PL2 model, the data are instead missing not at random and this would start to prove difficult to handle incomplete ranks.

8.3 Future Work

Future work involves extending the Plackett-Luce model to have multiple vases and perhaps learn the interaction between the labels. Since a later chosen label cannot be independent of that chosen before. A method of finding the best split position between the first vase and second vase should be more specific (i.e., top three ranks or the split position from one to three often

leads to the maximum likelihood estimation) instead of going through all possible positions and select the one that maximizes the log-likelihood. This can improve time used in computation, especially for the large size of datasets in terms of number of observations and number of labels to rank.

We observe that prediction performance can be influenced by the use of a mixture model. Improvement of model selection approach is likely to benefit the overall performance of the mixtures of PL1 model and PL2 model. In the likelihood ratio test, this is perhaps can be done by selecting the proper significant levels alpha instead of one fixed number for all datasets. We plan to further investigate along this direction.

A great deal additional research needs to be conducted with respect to combining label ranking methods and models. Alternative parameter estimation methods are needed. For example, Bayesian inference for ranking models can be applied to the Plackett-Luce model whenever parameters are not possible to be estimated by MLE.

Bibliography

Steven Beggs, Scott Cardell, and Jerry Hausman. Assessing the potential demand for electric cars. *Journal of econometrics*, 17(1):1–19, 1981.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, pages 324–345, 1952.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 89–96, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: 10.1145/1102351.1102363. URL http://doi.acm.org/10.1145/1102351.1102363.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

Weiwei Cheng and Eyke Hüllermeier. Instance-based label ranking using the mallows model. In *ECCBR Workshops*, pages 143–157, 2008.

Weiwei Cheng and Eyke Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009.

Weiwei Cheng and Krzysztof Dembczyn ÌĄski Hüllermeier, Eyke HüÌĹllermeier. Label ranking methods based on the plackett-luce model. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 215–222, 2010.

HE Daniels. Rank correlation and population models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 12(2):171–191, 1950.

Thomas Gärtner and Shankar Vembu. Label ranking algorithms: A survey. In Eyke Hüllermeier Johannes Fürnkranz, editor, *Preference Learning*. Springer–Verlag, 2010.

Isobel Claire Gormley and Thomas Brendan Murphy. Exploring irish election data: A mixture modelling approach. Technical report, Technical Report 05/08). Department of Statistics, Trinity College Dublin, Dublin 2, Ireland, 2005.

Eyke Hüllermeier, Johannes Furnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008. ISSN 0004-3702.

David Robert Hunter. Mm algorithms for generalized bradley-terry models. *Annals of Statistics*, pages 384–406, 2004.

Robert Duncan Luce. *Individual Choice Behavior a Theoretical Analysis*. john Wiley and Sons, 1959.

Colin Lingwood Mallows. Non-null ranking models. i. *Biometrika*, pages 114–130, 1957.

John Marden. *Analyzing and modeling rank data*. CRC Press, 1996.

Robin Lewis Plackett. The analysis of permutations. *Applied Statistics*, pages 193–202, 1975.

FW Scholz. Maximum likelihood estimation. *Encyclopedia of statistical sciences*, 1985.

Arthur Russell Silverberg. *Statistical models for q-permutations*. PhD thesis, [Sl: sn], 1980.

Ashok Sinyh. Metric methods for analyzing partially ranked data. *Technometrics*, 29(3):385–385, 1987.

Louis Leon Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

Ypma Tjalling. Historical development of the newton-raphson method. *SIAM Review*, 37(4): 531–551, 1995. doi: 10.1137/1037125. URL http://dx.doi.org/10.1137/1037125.

Philip LH Yu, WM Wan, and H Lee. Analyzing ranking data using decision tree. *Proceedings of ECML PKDD 2008*, pages 139–156, 2008.