ITEMSET SIZE-SENSITIVE INTERESTINGNESS MEASURES
FOR ASSOCIATION RULE MINING AND LINK PREDICTION

by

WALEED A. ALJANDAL

B.S, King Saud University, Riyadh, Saudi Arabia, 2000
M.S, Ball State University, Muncie, IN 2004

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2009

Abstract

Association rule learning is a data mining technique that can capture relationships between pairs of entities in different domains. The goal of this research is to discover factors from data that can improve the precision, recall, and accuracy of association rules found using interestingness measures and frequent itemset mining. Such factors can be calibrated using validation data and applied to rank candidate rules in domain-dependent tasks such as link existence prediction. In addition, I use interestingness measures themselves as numerical features to improve link existence prediction. The focus of this dissertation is on developing and testing an analytical framework for association rule interestingness measures, to make them sensitive to the relative size of itemsets. I survey existing interestingness measures and then introduce adaptive parametric models for normalizing and optimizing these measures, based on the size of itemsets containing a candidate pair of co-occurring entities. The central thesis of this work is that in certain domains, the link strength between entities is related to the rarity of their shared memberships (i.e., the size of itemsets in which they co-occur), and that a data-driven approach can capture such properties by normalizing the quantitative measures used to rank associations. To test this hypothesis under different levels of variability in itemset size, I develop several test bed domains, each containing an association rule mining task and a link existence prediction task. The definitions of itemset membership and link existence in each domain depend on its local semantics. My primary goals are: to capture quantitative aspects of these local semantics in normalization factors for association rule interestingness measures; to represent these factors as quantitative features for link existence prediction, to apply them to significantly improve precision and recall in several real-world domains; and to build an experimental framework for measuring this improvement, using information theory and classification-based validation.

ITEMSET SIZE-SENSITIVE INTERESTINGNESS MEASURES FOR ASSOCIATION RULE MINING AND LINK PREDICTION

by

WALEED A. ALJANDAL

B.S, King Saud University, Riyadh, Saudi Arabia, 2000
M.S, Ball State University, Muncie, IN 2004

A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2009

Approved by:

Major Professor
Dr.William H. Hsu

Copyright

Abstract

Association rule learning is a data mining technique that can capture relationships between pairs of entities in different domains. The goal of this research is to discover factors from data that can improve the precision, recall, and accuracy of association rules found using interestingness measures and frequent itemset mining. Such factors can be calibrated using validation data and applied to rank candidate rules in domain-dependent tasks such as link existence prediction. In addition, I use interestingness measures themselves as numerical features to improve link existence prediction. The focus of this dissertation is on developing and testing an analytical framework for association rule interestingness measures, to make them sensitive to the relative size of itemsets. I survey existing interestingness measures and then introduce adaptive parametric models for normalizing and optimizing these measures, based on the size of itemsets containing a candidate pair of co-occurring entities. The central thesis of this work is that in certain domains, the link strength between entities is related to the rarity of their shared memberships (i.e., the size of itemsets in which they co-occur), and that a data-driven approach can capture such properties by normalizing the quantitative measures used to rank associations. To test this hypothesis under different levels of variability in itemset size, I develop several test bed domains, each containing an association rule mining task and a link existence prediction task. The definitions of itemset membership and link existence in each domain depend on its local semantics. My primary goals are: to capture quantitative aspects of these local semantics in normalization factors for association rule interestingness measures; to represent these factors as quantitative features for link existence prediction, to apply them to significantly improve precision and recall in several real-world domains; and to build an experimental framework for measuring this improvement, using information theory and classification-based validation.

Table of Contents

List of Tables

List of Figures

Acknowledgements

First, all praise and glory are due to Allah for all the bounties granted to me. Without God’s support, guidance and help, this achievement would not be possible. I would like to express my deepest and sincere gratitude to my Major Professor Dr. William Hsu for his valuable guidance, encouragement, and help during my dissertation work. Dr. Hsu has been an excellent, positive and very helpful mentor for me.

My indebted thanks go to my Committee Members, Dr. Doina Caragea, Dr. Torben Amtoft, and Dr. Fayez Husseini for their valuable suggestions, advice and encouragement. I also would like to thank Dr. Satish Chandra for serving as the Outside chair for my final exam. I would like to extend my appreciation to all my colleagues, especially Tim Weninger, Vikas Bahirwani, Jing Xia and Landon Fowles for their cooperation and help.

I sincerely appreciate the feedback of Dr. Jiawei Han and Dr. ChengXiang Zhai, and I am grateful for helpful discussion with them.

I am grateful to my parents for their patience, support and prayers. I would like to thank my wife for her love, sacrifices and continuing support during my graduate study and I would like to express my sincere love to my kids (Wesam, Khalid, and Faris).

Finally, I am thankful to Imam University for providing me with the scholarship and supporting my graduate studies.

Dedication

To my parents, my wife and my kids

CHAPTER 1 - **INTRODUCTION**

The main goal of data mining is to define a process for discovering significant patterns or anomalies in a large volume of data. It has been applied to decision support problems in diverse areas such as medical diagnosis, targeted marketing, bioinformatics, sociology, networking, and information security, making data mining one of the most widely studied topics in intelligent systems. Data mining incorporates theory and practical developments from many older research areas such as databases, machine learning, artificial intelligence, distributed computing, information retrieval, and statistics, and lends an integrative perspective to these research areas. Due to the breadth of both applications and foundational theory in data mining research, it is often divided along methodological lines, into tasks such as classification, clustering, association, decision support, and visualization. Association rule mining is one of these subtopics which have been explored by many research groups. It addresses the problem of discovering relationships between instances that originate from dependence or interaction.

1.1 Association rule mining

Association rule mining is a method for discovering the relationships or correlation between items based on measures that are defined over observed items and proposed relationships. One of the most important aspects of association rule mining is ranking rules by their significance, according to some quantitative measure that expresses their interestingness with respect to a decision support or associative reasoning task.

The concept of association rules was first introduced in a 1993 article (Agrawal, Imielinski, & Swami, 1993) in which the Apriori algorithm was also presented. Since then, association rule mining has become one of the most highly used and studied techniques in data mining. The main principal of this technique involves discovering the efficient relationship and co-occurrence between items in the data. In other words, it discovers and measures quantitative evidence for relationships expressed in the database.

Association rules are expressed in an IF-THEN propositional rule-based format. A classic example of this method is market basket analysis. Consider a simple example:

“customers who buy product *A* often also buy product *B*”. A decision maker such as a shopper or a marketer can access a large volume of historical data from which such rules have been extracted, to more confidently draw conclusions and make decisions that are well-supported by the data.

1.1.1 Formal definition:

Let L = {I_1, I_2, …, I_m} be a set of *m* distinct attributes (items). Let D be a database, where each record (itemset) T has a unique identifier, and contains a set of items such that T $\subseteq$ L. An association rule is an implication of the form X→Y, where both X, Y $\subset$ L, are sets of items called itemsets, and (X ∩ Y = ϕ) where X and Y are two disjointed sets of items. Here, X is called the *antecedent*, and Y the *consequent*. The rule can be described as when we find all items in X within a transaction it is likely the transaction also contains the items in Y (Agrawal, Imielinski, & Swami, 1993). The first step in generating the rules is applying frequent item set algorithms over all possible rules. The rules will then be selected based on thresholds and measures of significance and interestingness.

1.2 Measurement of Association Rules

Generating association rules from a certain dataset will lead to a large number of rules if we do not specify a threshold for each specific measure. In this introduction, I present the two most fundamental association rule interestingness measures, support and confidence, which are the basis of the Apriori algorithm.

1.2.1 Support

Support is a basic measure related to probability and set theory. It is defined as the fraction of transactions in the database which contain all items in a specific rule (Agrawal, Imielinski, & Swami, 1993). This can be written as:

$$\text{Supp}(X \rightarrow Y) = \text{Supp}(X \cup Y) = \frac{|xy|}{|D|}$$

Where |xy| is the number transactions (itemset) which contain both X and Y – i.e., the probability of (x, y) – and |D| represents the total number of transactions (itemset) in the database.

Minimum support threshold are usually specified in generating the association rules which select only the most frequent items in the database.

1.2.2 Confidence

Another measure of the association rules is confidence. This is the strength of the implication of a rule and can be represented as a ratio between the transaction numbers, including X and Y and those including X, which can be written as:

$$\mathrm{Conf}(X \rightarrow Y) = \frac{\mathrm{Supp}\ (X \rightarrow Y)}{\mathrm{Supp}\ (X)} = \frac{|\,xy\,|}{|\,x\,|}$$

Where |x| is the number of transactions (itemset) containing X.

Example of Association rules: market basket analysis, which is analyzing customer buying habits by finding associations between items that customers place in their “shopping baskets”.

An illustrative example follows. Table 1-1 depicts part of a market basket database.

Transaction ID	Items list
1	bread, coffee, tea, candle, BBQ-s
2	BBQ-s, chicken, bread, flower
3	bread, coffee, juice1, juice2
4	BBQ-s, chicken, tea
5	tea, candle, bread, coffee, egg

Table 1-1 Market basket (example)

Transaction-ID presents the single shopping basket which contains a list of items. The first step is to find the frequent items in the dataset with user defined threshold. The next step is to use these items to generate the association rules based on co-occurrences and specific rule measures.

Table 1-2 presents some of the measures and rules which have been generated from the dataset.

Measure	Value
Supp (bread)	3/5 = 0.6
Supp (BBQ-s)	2/5 = 0.4
Supp (tea)	3/5 = 0.6
Supp (bread, coffee)	0.6
Supp (BBQ-s, chicken)	0.4
Supp (tea , candle)	0.4
Rules	
Conf (bread → coffee)	0.75
Conf (BBQ-s → chicken)	0.667
Conf (tea → candle)	0.667

(Supp= Support, Conf= Confidence)

Table 1-2 Rules and measures (example)

1.3 Needs and Objectives

In finding association rules, probability (support) has been used as the main character in the process of finding significant rules. In fact, joint probability (of a subset) is the measure of how likely two or more items are to appear together. This probability can be found in most of the association and correlation measures. Association rules are collected over large numbers of transactions or other collections of items, and because the number of candidate associations grows quadratically in the number of distinct items in the worst case. The need to scale association rule mining up to thousands or millions of transactions and items leads to a need for better interestingness measures. Many interestingness measures have been derived as a form of preference bias, in order to reduce the number of candidate associations that are initially generated. However, interestingness measures have variable effects when applied to different domains; in some such domains, we are able to measure properties of the data. For example, how much variance there is in the sizes of itemsets, to capture some meta-knowledge, such as how significant it is that a pair of items frequently occurs together among triples.

This research can be justified by observing several needs. First, even though statistical measures of interestingness have been derived, which can be systematically applied to generate rules meeting specific requirements, there is still a need to make these measures more sensitive to generic properties of the original data. Furthermore, the refined metrics need to be validated by application to different domains. Second, we can take advantage of existing data (itemset size) behaviors to improve the quality of predictions achieved using a particular interestingness measure - for example, when associations are used as numerical features for classification. This quality is measured in terms of precision *versus* recall (sensitivity *versus* specificity) over tasks such as link analysis in the application domain. Moreover, the new method should be extendable and compatible with any measures that have been discovered or will be discovered. To evaluate an association rule learning system that supports a performance element such as link existence prediction, an evaluation framework and data flow model must be specified.

One of the objectives of this research is to derive a normalization factor that takes into account the itemset size, to better measure the strength of an association in a specified domain. I hypothesize that data-driven calibration of this factor for each domain will improve measures of quality such as precision, recall, F-score, and accuracy. Using this factor improves the sensitivity of interestingness measures, which leads to more significant interesting rules.

Another objective from the evaluation point of view is present two evaluation frameworks. The first one is a framework that can be used to evaluate different measures based on attribute's surprisingness measure (more interesting).

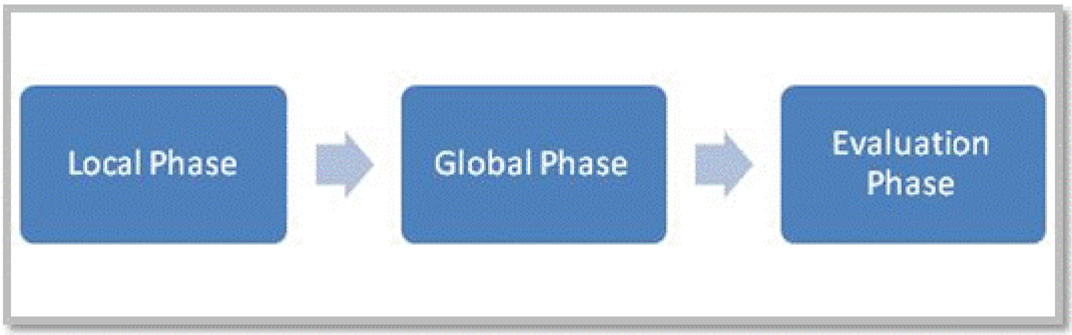


Figure 1-1 Overview of the three phases' evaluation process (information gain)

The main processes can be divided into three phases as shown in Figure 1-1. In the initial local phase, I generate two different order lists of rules based on two different versions of specific measure scores. At the end of this phase, I combine the two lists into one global list. In the subsequent global phase, I construct a global list from the two local lists without any scores (non-ordered) then I apply independent scoring method (Attribute Surprisingness introduced by Freitas (1999)) by using information gain to the global list of rules to make a new order, which I consider as a gold standard. The final phase is the evaluation phase. At this point we have three lists: two locals and one global. By making selective rules from one of the local lists and the global list to get the first comparison result (Result 1), which we can compare with the other comparison result (Result 2) resulting from comparing the other local list with the global list. At the end the final result will show the differences using deferent comparison criteria (in my research I use inversions). I will present the details of this experiment design in Section 5.3.

The other evaluation framework is based on classification methods. The *LiveJournal* data and bioinformatics data sets have been used in this framework, because it has the ground truth of friendship relation and protein-protein interaction. The main objective of classification is to build a model to predict the class of different instances based on given features. This provides a driving problem, a classification technique to be augmented using associational features, and a source of ground truth by which to measure the improvement attained using these features. The framework contains three phases: pre-processing phase, classification phase and comparison phase as shown in Figure 1-2. More details about designing this experiment are given in Section 5.1.2 Evaluation through Classification Methods.

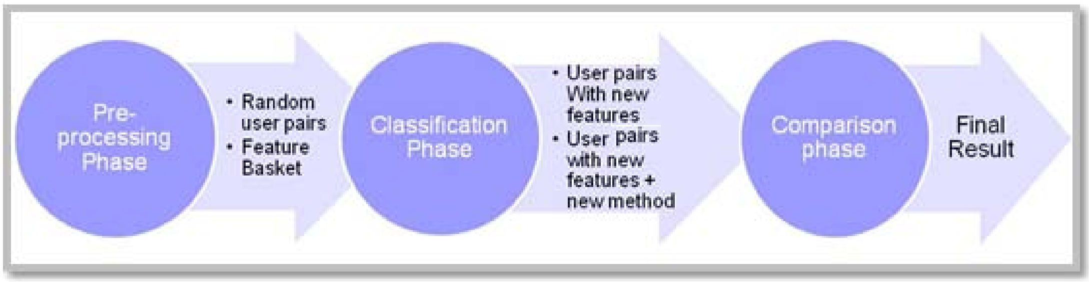


Figure 1-2 Overview of the three phase's evaluation process (classification)

The synopsis of the research is as follows. In CHAPTER 2 - there will be a survey of related work and methodologies of interestingness measures and some concepts related to the measurement theory and link mining. CHAPTER 3 - will be presenting the item-set-size normalization concept and methods and relatedness to domain specific. A survey and related concept of application domains will be presented in CHAPTER 4 - The experiment design includes more details about both evaluation frameworks will be introduced in CHAPTER 5 - . The experiments results will be illustrated and discussed in CHAPTER 6 - includes the interpretations. Conclusion and some possible feature work will be presented in CHAPTER 7 - More experiments materials will be presented in the appendixes.

CHAPTER 2 - **RELATED WORK AND METHODS**

Notation

C_k	set of candidate k-item sets	AttSupp	attribute surprisingness
L_k	set of large k-item sets	InfoGain	information gain
Conf	confidence	TP	true positive
Supp	support	FP	false positive
CD	concept description	FN	false negative
T	single transaction	TN	true negative
L	set of items	D	set of transactions
		N	total number of transactions

In this chapter, I present the problem of deriving interestingness measures for association rule mining and that of link prediction and survey related work on both topics. I then survey several related classification and clustering methods for the supervised and unsupervised inductive learning part of my framework.

2.1 Association rule interestingness measures

The need for interestingness measures originates from limitations of the support and confidence approach. Even though there is a reasonable concept behind the support and confidence approach there are still some cons in using this approach for rule extraction. Brin *et al.* (1997) and Aggarwal *et al.* (1998) address the weaknesses of the support- confidence framework.

The algorithms of support and confidence generate a very large number of rules where many of them are not interesting to the user. Indeed, if the confidence of the rule A→ B is equal to the marginal frequency of B (P (B|A) = P(B)) which indicate that A and B are independent, then the rule A → B adds no new information (e.g. P(A) = 0.7, P(B) = 0.8, P(AB) = 0.56, P(B|A) = 0.8).

Interestingness measures can be divided into two parts subjective and objective. The interestingness measures can be used in three ways; Figure 2-1 shows the frame work of those used (Geng & Hamilton, 2006). First, measures can be used to prune uninteresting patterns during the mining processes to narrow the search space and thus improve mining efficiency. An example of this would be selecting significant rules by setting a threshold for support to filter out the rules which have low support during the mining process (Agrawal & Srikant, 1994). Second, measures can be used to rank patterns according to the order of their interestingness scores. Third, measures can be used during post-processing to select interesting patterns. For example, we can use the chi-square test to select all rules that have significant correlations after the data mining process (Bay & Pazzani, 1999).

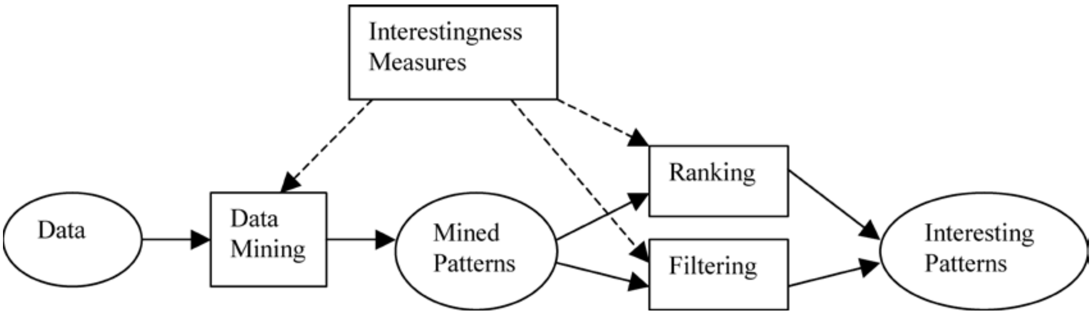


Figure 2-1 Interestingness measures in the data mining process.

The problem of reducing the number of association rules using different measures has been discovered by many researchers to select appropriate measures (subjective or objective) for particular domains and requirements.

2.1.1 Subjective Measures

A number of subjective measures have been proposed. Over all, subjective measures put some facts to generate a smallest possible set of rules which can be more interesting and useful by using some of the user knowledge.

One of the analyzing approaches was proposed (Liu, Hsu, Chen, & Ma, 2000) as a subjective approach that assists the user in finding interesting rules by analyzing the discovered association rules using the user's existing knowledge about the domain.

Liu *et al.* (1999) presented a ranking method for mined patterns according to the user's existing knowledge and general impressions. The main disadvantage is that the user is required to express his/her knowledge in the specifications, which might not be an easy and standard task.

In other research Tuzhilin and Silberschatz (1996) discussed subjective measures depending on two concepts' actionability and unexpectedness and the relation between them. Actionability states that the pattern is interesting if the user can act on it to his advantage. Unexpectedness focuses on the surprising factor for the pattern. Also, it relates to beliefs which can be defined as logical statements. There are two types of beliefs: hard, which the user is not willing to change; and soft, which the user can change if suggested by newly discovered patterns.

2.1.2 Objective Measures

Deriving an objective interestingness measure usually involves estimating some aspect of a candidate rule's structure, analytical performance, and statistical significance with respect to observed itemset data. Compound measures are based on primitive measures grounded in probability density functions, with some – such as the normalization approach described in this paper – based on parametric fusion of these primitive measures, while others are based on more ad hoc rules of combination.

Piatetsky-Shapiro (1991) considered as first proposal using statistical independence of rules as an interestingness measure. More methods have since been proposed using different statistical approaches. Brin, *et al*. (1997) proposed lift and $\chi 2$ (chi-squared) as correlation measures and developed an efficient mining method. Hilderman and Hamilton (2001) and Tan, *et al.* (2002) have comparative studies of different interestingness measures and address the concept of null-transactions. Because the probability of an item appearing in a particular transaction is usually very low, it is desirable that a correlation measure should not be influenced by these transactions which they call *"null-transactions"*, i.e., the transactions that do not contain any of the items in the rule being examined. In another study related to the correlation. Omiecinski (2003), and Lee, *et al.* (2003) found that *all_confidence*, *coherence,* and *cosine* are null-invariant and are thus good measures for mining correlation rules in transaction databases.

After all these studies, Tan *et al.* (2002) discuss the properties of twenty-one objective interestingness measures and analyzes the impacts of Support based pruning and contingency table standardization. This study ends with the conclusion that there is no measure that is consistently better than others in all application domains. However, some of these measures are correlated with each other.

Three measures for capturing relatedness between item pairs are proposed by Shekar and Natarajan (2004). These measures use the concept of function embedding to appropriately weigh the relatedness contributions due to Mutual Interaction, complementary and substitutability between items. At the end they propose interestingness coefficient by combining the three relatedness measures. All the three measures are calculated based on the probability without taking into account the transaction itself (large or small).

Support	$P(AB)$
Confidence/Precision	$P(B\mid A)$
Coverage	$P(A)$
Prevalence	$P(B)$
Recall	$P(A\mid B)$
Specificity	$P(\frac{\neg B}{\neg A})$
Accuracy	$\mathrm{P(AB) + P(\neg\ A \neg\ B)}$
Lift/Interest	$P(B\mid A)/P(B)$ -or- $P(AB)/P(A)P(B)$
Leverage	$P(B\mid A) - P(A)P(B)$
Added Value /Change of Support	$P(B\mid A) - P(B)$
Relative Risk	$P(B\mid A)/P(B\backslash\neg\ A)$
Jaccard	$P(AB)/(P(A) + P(B) - P(AB))$
Certainty Factor	$\frac{P(B\mid A)-P(B)}{1-P(B)}$,
Odds Ratio	$\frac{P(AB)P(\neg A\neg B)}{P(A\neg B)P(\neg BA)}$
Yule's Q	$\frac{P(AB)P(\neg A\neg B)-P(A\neg B)P(\neg AB)}{P(AB)P(\neg A\neg B)+P(A\neg B)P(\neg AB)}$
Yule's Y	$\frac{\sqrt{P(AB)P(\neg A\neg B)}-\sqrt{P(A\neg B)P(\neg AB)}}{\sqrt{P(AB)P(\neg A\neg B)}+\sqrt{P(A\neg B)P(\neg AB)}}$
Klosgen	$\sqrt{P(AB)}\big(P(B\mid A) - P(B)\big), \sqrt{P(AB)}MAX(P(B\mid A) - P(B), P(A\mid B) - P(A))$
Conviction	$\frac{P(A)P(\neg B)}{P(A\neg B)}$
Collective Strength	$\frac{P(AB)+P(\neg B\mid\neg A)}{P(A)P(B)+P(\neg A)*P(\neg B)} * \frac{1-P(A)P(B)-P(\neg A)*P(\neg B)}{1-P(AB)-P(\neg B\mid\neg A)}$
Laplace Correction	$\frac{N(AB)+1}{N(A)+2}$
Gini Index	$P(A) * \{P(B\mid A)^2 + P(\neg\ B\mid A)^2\} + P(\neg\ A) *$ $\{P(B\mid \neg\ A)^2 + P(\neg\ B\mid \neg\ A)^2\} - P(B)^2 - P(\neg\ B)^2$
Normalized Mutual Information	$\sum_i \quad \sum_j \quad P\big(A_i B_j\big) log_2 \frac{P(A_i B_{j)}}{P(A_i)P(B_j)} / \{-\sum \quad _i P(A_i) * log_2 P(A_i)\}$
J Measure	$P(AB)log\left(\frac{P(B\mid A)}{P(B)}\right) + P(A \neg\ B)log\left(\frac{P(\neg B\mid A)}{P(\neg B)}\right)$
One-Way Support	$P(B\mid A) * log_2 \frac{P(AB)}{P(A)P(B)}$
Two-Way Support	$P(AB) * log_2 \frac{P(AB)}{P(A)P(B)}$
Φ-Coefficient (Linear Correlation Coefficient)	$\frac{P(AB)-P(A)P(B)}{\sqrt{P(A)P(B)P(\neg A)P(\neg B)}}$
Piatetsky-Shapiro	$P(AB) - P(A)P(B)$
Cosine	$\frac{P(AB)}{\sqrt{P(A)P(B)}}$
Loevinger	$1 - \frac{P(A)P(\neg B)}{P(A\neg B)}$
Information Gain	$log \frac{P(AB)}{P(A)P(B)}$
Sebag-Schoenauer	$\frac{P(AB)}{P(A\neg B)}$
Least Contradiction	$\frac{P(AB)-P(A\neg B)}{P(B)}$
Odd Multiplier	$\frac{P(AB)P(\neg B)}{P(B)P(A\neg B)}$
Example and Counterexample Rate	$1 - \frac{P(A\neg B)}{P(AB)}$
Zhang	$\frac{P(AB)-P(A)P(B)}{\max(P(AB)P(\neg B),P(B)P(A\neg B)}$

Table 2-1 Probability Based Objective Interestingness Measures

Berberidis, *et al*. (2005) and George, *et al*. (2006) introduce a data mining paradigm, which involves the discovery of contiguous frequent item sets and present level-wise algorithm for finding these item sets, which lead them to generate a two level global support (gsup) the first-level support and local support (lsup), the second-level support. For the evaluation, they introduce new metric (Mutual Exclusion Metric) to evaluate the degree of the mutual exclusion between two items.

Another measure based on information theory (Blanchard, Guillet, Gras, & Briand, 2005) designed a rule interestingness measure, Directed Information Ratio. This measure filters out these rules whose antecedent and consequent are negatively correlated and these that have more counter examples than examples.

The new survey, (Geng & Hamilton, 2006) reviews the interestingness measures for rules and summaries, classifies them from several perspectives and compares their properties. They present thirty-eight probability based objective interestingness measures for association rules Table 2-1. Another paper by Lenca, *et al.* (2007) studied twenty interestingness measures by using 10 data sets. This study is compared to an analysis of formal properties of the measures which make a best choice of user’s needs.

There are some research papers where they tried to improve the measures quality by using some existing information. One of these papers Hilderman, *et al.* (1998) proposed a concept of share-confidence and support which involves the quantity and price of the items in the confident and support computation.

Moreover, in specific application domains some attributes can have very different degrees of interestingness for the user, depending on which attributes occur in the rule antecedent. Thus, in some applications, different attributes might have very different “costs” to be accessed. The typical example is medical diagnosis. For example, some health-related attributes can only be determined by performing a very costly examination. Suppose that the antecedent (“if part”) of a discovered rule *r1* involves the result of an exam *e1* which costs $200, while the antecedent of a discovered rule *r2* involves instead the result of another exam *e2* which costs $20. All other things (including prediction accuracy) being equal, we would rather use rule *r2* for diagnosis. So the cost of the attribute becomes part of interestingness decision. There are some data mining algorithms which take into account attribute costs like what had been described in Ming (1993) and Turney (1995).

From Table 2-1 we can see how the Probability Based objective interestingness measures contain joint probability (co-occurrence) as an important part which may affect measurement value. However, in the calculation of joint probability, items relations in the dataset itemset are treated equally in all interestingness measures without differentiation between one itemset and others. Even if some interestingness measures adopt attributes values to adjust the final result, there is still no change in the value of the joint probability. I seek to make joint probability based measures more sensitive to the relationship between items and each itemset, using itemset size to reflect the real relation between items. This is first step in enriching the semantics of interestingness measures and increasing their usefulness in some domains.

2.2 Frequent pattern concepts and Algorithms

The concept of frequent pattern comes from searching for correlation relationship between instances (such as itemsets, sub-graphs) that may lead to further useful knowledge. The discovering of more interesting associations between items gets attention from many industrial and business decision-makers to assist their decisions (such as cross-marketing, user behavior analysis). On the other hand, discovering interesting associations improves some recommended systems such as friends' recommendation system in social networks.

One of the challenges in this area is efficiently discovering frequent items in a large dataset. Next is discovering the association between interesting frequent items. This section will cover some of the many algorithms used to discover the frequent itemsets which is related to my research.

2.2.1 Apriori Algorithm

The Apriori algorithm is probably the most well-known algorithm in the area of frequent items discovery (Agrawal, Imielinski, & Swami, 1993). The algorithm takes advantage of the property that any subset of a frequent item set must be a frequent item set. If we have (N+1)-item set then we use the (N)-item set (N is number of items in the set) to discover it. Thus, the discovered frequent item sets of the first pass are used to generate the candidate sets of the second pass. Once the candidate 1-item sets are found their supports are counted to discover the frequent 2-itemsets by scanning the database. In the third pass, the frequent 2-itemsets are used to generate candidate 3-item sets. Termination condition, where there are no more new frequent item set, is found in Figure 2-2 . The algorithm contains two steps:

1. Join step:

The first step is to join all frequent items of size k-1 ((k-1)-item set) with themselves to generate candidate K-item set. As a result the new list of k-item sets has been produced.

2. Prune step:

This step come from the Apriori property which states if an item set is not frequent, then all its supersets are absolutely not a frequent set. Therefore we can prune all Candidate k- item sets by checking whether all its (k-l)-item sets subsets are frequent or not. If we find any member of (k-1)-item sets is not we can prune its superset from a new list.

```
-------------------------------------------------------------------
Method: apriori_gen() [ (Agrawal & Srikant, 1994)]
Input: set of all large (k-1)-item sets $L_{k-1}$
Output: A superset of the set of all large k-item sets
// Join step
$I_i$ = Items i
    insert into $C_k$
    Select $p.I_1$, $p.I_2$, ……. , $p.I_{k-1}$, $q.I_{k-1}$
    From $L_{k-1}$ is p, $L_{k-1}$ is q
    Where $p.I_1 = q.I_1$ and …… and $p.I_{k-2} = q.I_{k-2}$ and $p.I_{k-1} < q.I_{k-1}$.
// Pruning step
For all item sets $c \in C_k$ do
    For all (k-1)-subsets s of c do
        If ($s \notin L_{k-1}$) then
            delete c from $C_k$
--------------------------------------------------------------------
```

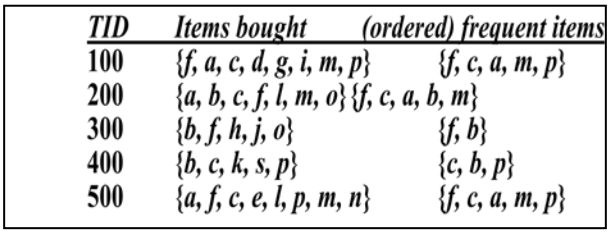
Figure 2-2 Apriori Algorithm Pseudo code

The main disadvantage of the Apriori algorithm is running time, because the algorithm needs to scan the database for every processed level. The performance of the algorithm will be unacceptable when the database size is large, however, there are many algorithms that have been proposed to solve this problem and improve the performance of the processing time of finding the frequent items.

2.2.2 FP-growth Algorithm

Han *et al.* (2000) proposed the FP-growth algorithm for mining the complete set of frequent patterns. This algorithm is based on a new frequent pattern tree (FP-tree) structure, which is a prefix tree for storing necessary information about frequent patterns. Only frequent 1-item(s) are stored in each node of the tree. The FP-tree is applied to restrict generation of a large number of candidate sets. This concept eliminates the multi-scan inefficiency the Apriori algorithm. FP-growth is adapted to the pattern growth approach to avoid scanning the database for every level of frequency and handles a large number of candidate sets. The algorithm begins with frequent 1-items which are kept in the FP-tree to perform recursive mining. The search technique is a partitioning-based divide-and-conquer method to increase the running time efficiency.

In Figure 2-3 an example of transaction table of market basket where each row represents a single market basket with the list of items that have been bought (Han, Pei, & Yin, 2000).



TID	*Items bought*	*(ordered) frequent items*
100	{*f*, *a*, *c*, *d*, *g*, *i*, *m*, *p*}	{*f*, *c*, *a*, *m*, *p*}
200	{*a*, *b*, *c*, *f*, *l*, *m*, *o*}	{*f*, *c*, *a*, *b*, *m*}
300	{*b*, *f*, *h*, *j*, *o*}	{*f*, *b*}
400	{*b*, *c*, *k*, *s*, *p*}	{*c*, *b*, *p*}
500	{*a*, *f*, *c*, *e*, *l*, *p*, *m*, *n*}	{*f*, *c*, *a*, *m*, *p*}

Figure 2-3 Transactions table

The processes of the algorithm are arganized into these stages :

1. Scan DB once, find frequent 1-itemset (single item pattern)
2. Order frequent items in frequency descending order
3. Scan DB again, construct FP-tree, as in Figure 2-4
4. Traverse the FP-tree by following the link of each frequent item
5. Accumulate all of the transformed prefix paths of that item to form a conditional pattern base
6. Construct the FP-tree for the frequent items of the pattern base

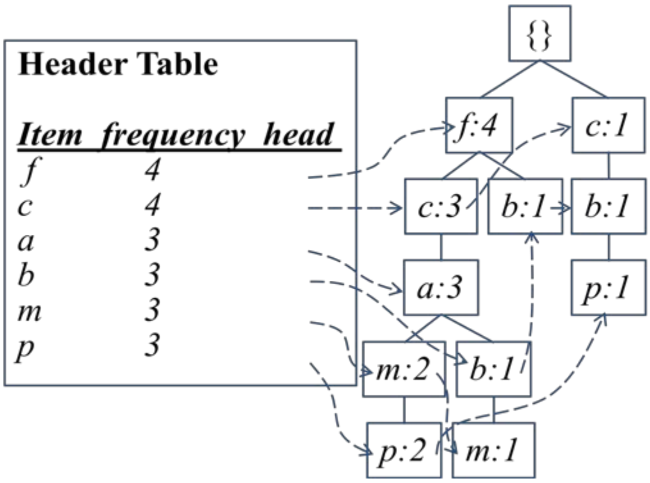


Figure 2-4 Example of FP-Tree (Han, Pei, & Yin, 2000)

2.3 Association rules interestingness measures

Discovering correlation relationship from frequent items is the next step of generating the association rules. From this step, there are too many discovered rules some of them are redundant or not interesting. Early in Chapter 1, I introduced two main measures (support and confidence) and Agrawal, *et al.* (1993) suggests using a threshold on confidence to reduce the number of discovered rule. The main problem of raising the threshold of these two measures is missing some interesting rules, so there is a need for more interesting measures which can rank the rules based on their interestingness. However, the effect of interestingness measures can be different from domain to domain.

2.3.1 Interestingness Measures

There are many interestingness measures, but I will describe some measures that I am going to use in this research.

2.3.1.1 Lift

Lift measures show the relationship between two or more items when they occur together more often than expected, if they were statistically independent. Introduced by Brin, *et al.* (1997).

Lift = confidence / expected confidence

Let X and Y are two different set of items then lift can be defend as:

$$\text{Lift } (X \rightarrow Y) = \frac{\text{Conf } (X \rightarrow Y)}{\text{supp } (Y)} = \frac{\text{Conf } (Y \rightarrow X)}{\text{supp } (X)} = \frac{P(X \cup Y)}{(P(X)\ P(Y)}$$

The expected confidence is identical to the support of the rule head. It is assumed in the definition of the expected confidence that there is no statistical relationship between the rule body and the rule head. This means that the occurrence of the rule body does not influence the probability for the occurrence of the rule head and vice versa. The lift is a measure for the deviation of the rule from the model of statistic independency of the rule body and rule head. The lift is a value between 0 and infinity:

- A lift value greater than 1 indicates that the rule body and the rule head appear more often together than expected. This means that the occurrence of the rule body has a positive effect on the occurrence of the rule head.
- A lift smaller than 1 indicates that the rule body and the rule head appear less often together than expected. This means that the occurrence of the rule body has a negative effect on the occurrence of the rule head.
- A lift value near 1 indicates that the rule body and the rule head appear almost as often together as expected. This means that the occurrence of the rule body has almost no effect on the occurrence of the rule head (Brin, Motwani, Ullman, & Tsur, 1997).

2.3.1.2 Conviction

Conviction is one of interestingness measures proposed by Sergey, *et al.* (1997). It was developed as an alternative measure to confidence, which does not capture direction of associations effectively, and is defined as:

$$\text{Conviction } (A \rightarrow B) = \frac{P(A)P(\neg B)}{P(A \neg B)}$$

The logical justification for this measure is that $A \rightarrow B$ can be rewritten as $\neg (A \wedge \neg B)$ now by observing $A \wedge \neg B$ we can see how different from independence and invert the ratio to take care of the outside negation. The value of conviction equals 1 when and if the target item is completely unrelated.

2.3.1.3 Leverage

The leverage measure was introduced by Piatetsky-Shapiro (1991), It measures the expected dependence between items. According to the paper that introduced leverage, leverage and lift measure are similar, except that leverage measures the difference between the probability of co-occurrence of A and B as the independent probabilities of each of A and B. defined as:

$$\text{Leverage (A→B)} = P(B|A) - P(A)P(B)$$

2.3.1.4 Other Interestingness Measures

There are many interestingness measures that I listed in Table 2-1 and I am going to introduce the ones that I use in some of my experiments such as:

1. Match

   Defined as:

$$\text{Match (A→B)} = \frac{P(AB) - P(A) * P(B)}{P(A) * (1 - P(A)}$$

2. Accuracy

   Defined as:

$$\text{Accuracy (A→B)} = P(AB) + P(\neg A \neg B)$$

Each measure has a different way of handling the co-occurrences and the independences.

2.3.2 Information-Theoretic measure (Attribute surprisingness)

AttSurp or attribute surprisingness, (Freitas, 1999) is a term for measuring rule surprisingness. This measure is a reciprocal function of average information gain, an information theoretic measure based on condition entropy. The information gain of each attribute is defined as the class entropy minus the class entropy given the value of the predicting attribute. This is similar to the idea in decision tree-based classification that attributes, with high information gain, are good predictors of class. These attributes are also considered individually, i.e. one at a time. On the other hand, a user who has background knowledge about his/her domain application knows what the best predictors (attributes) for this domain are. Thus, rules containing these attributes would tend to have lower degrees of surprisingness (interestingness) for the user. When an attribute appears in rules with low information gain, which the user did not expect, the user will be surprised and therefore the rules will be interesting. The AttSurp is defined as:

$$\text{AttSupp} = \frac{1}{\left(\frac{\sum_{1}^{k} InfoGain\ \ (A_i)}{K}\right)}$$

Where InfoGain (*Ai*) is the information gain of the ith attribute occurring in the rule and *k* is the number of attributes (more detailed information in CHAPTER 5 -

2.4 Classification

Classification is a data mining task that often applies machine learning-specifically, inductive learning. The main objective of classification is to build a model to predict the class of different instances based on given features. The classifier starts with using training set which has correct answers (class label attribute) then creating a model by running the algorithm on the training data. This model will be used to predict the class of the rest of data. Finally, it tests the model and measure the performance.

Formally, the problem can be stated as follows:

Given training data $\{(x_1, y_1), (x_2, y_2) \dots . (x_n, y_n)\}$ *produce a classifier* $h: x \rightarrow y$ *that maps an object* $x \in X$ *to its classification label* $y \in Y$*.*

Many classification learning methods have been proposed. Research in this area has led to scalable and efficient algorithm for classification. However, the classifiers have been built based on different structural models (tree-based classifier, rule-based classifiers, etc.).

2.4.1 Best-First Decision Tree Classifier (tree based)

One widely-used method in classification is the induction of decision trees introduced by Quinlan (1986). A decision tree is a flow-chart-like structure consisting of internal nodes, leaf nodes, and branches. Each internal node represents a decision, or test, on a data attribute, and each outgoing branch corresponds to a possible outcome of the test. Decision trees can be represented as sets of IF-THEN rules (Mitchell, 1997). There are many algorithms based on decision trees.

Standard decision tree learners such as C4.5 expand nodes in depth-first order (Quinlan J. R., 1993), while in best-first decision trees learning first introduced by Haijian (2007); the “best” node is expanded first. The “best” node is the node whose split leads to maximum reduction of impurity among all nodes available for splitting. The algorithm of the best-first tree can be summarized as follows: First, select an attribute to place at the root node and make some branches for this attribute based on some criteria. Then, split the training instances into subsets, one for each branch extending from the root node. Then, this step is repeated for a chosen branch, using only those instances that actually reach it. In each step we choose the “best” subset among all subsets that are available for expansions. This constructing process continues until all nodes are pure or a specific number of expansions are reached (Haijian, 2007). The best-first method always chooses the node for expansion whose corresponding best split provides the best information gain or Gini gain among all unexpanded nodes in the tree.

2.4.2 Random Forest Classifier (tree-based)

Random forest is a classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees. The algorithm for inducing a random forest was developed by Breiman (2001)

The Random Forests algorithm grows many classification trees. Than when we classify a new object from an input vector, apply each tree in the forest to the input vector. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (overall the trees in the forest) (Breiman, Random Forests, 2001). Each tree is grown as follows:

1. If the number of cases in the training set is *N*, sample *N* cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree.
2. If there are *M* input variables, a number *m*<<*M* is specified such that at each node, *m* variables are selected at random out of the *M* and the best split on this *m* is used to split the node.
3. Each tree is grown to the largest extent possible. There is no pruning.

2.4.3 OneR Classifier (rule based)

OneR stands for “One Rule” and is a simple algorithm proposed by Holte (1993) which induces classification rules based on the value of a single attribute. The OneR algorithm chooses the most informative single attribute and bases the rule on this attribute alone which is the rule with the smallest error rate. To create a rule for an attribute, the most frequent class for each attribute value must be determined. This class is simply the class that appears most often for that attribute value (Holte, 1993). In other words, OneR selects the rule with the lowest error rate. The error rate is the number of training data instances in which the class of an attribute value does not agree with the binding for that attribute value in the rule. Random selection will be used if there are two or more rules that have the same error rate. The basic idea of the algorithm is as follow:

```
For each attribute a, form a rule as follows:
  For each value v from the domain of a,
    Select the set of instances where a has value v.
    Let c be the most frequent class in that set.
    Add the following clause to the rule for a:
      if a has value v then the class is c
  Calculate the classification accuracy of this rule.
Use the rule with the highest classification accuracy.
```

The interested reader is referred to Holte (1993) for more details.

2.4.4 IB1Classifier (instance-based)

IBK (Aha & Kibler, 1991) uses the k-nearest neighbor approach for classification where the class of a test item is derived from the training instances that are most similar to it. Instance based learning is often referred to as “lazy” learning because it stores all training examples in the memory during the learning process.

IB1 is identical to the Nearest Neighbors algorithm except that it normalizes its attributes ranges and processes instances incrementally. According to k-Nearest Neighbors, each instance is treated as a point in n-dimensional space where n is the number of features that describe the instance. When a new instance is classified, the algorithm looks for the k most similar instances (Nearest Neighbors) in the set of training examples. The similarity is based on the distance in the k-dimensional space between instances. The distance is computed as a Euclidean distance:

$$\Delta(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Where x_i and y_i refer to value of ith feature of x and y instance. In the case where two or more labels are equally frequent the label that is first seen in the training set is chosen. Table 2-2 IB1 shows the pseudocode of the IB1 algorithm (Aha & Kibler, 1991).

```
--------------------------------------------------
CD ← D
For each $x \in$ Training Set do
    1. for each $y \in$ CD do
           Sim[$y$] ← similarity($x$, $y$)
    2. $y_{max}$ ← some $y \in$ CD with maximal Sim[$y$]
    3. If class($x$) = classs($y_{max}$) then
              Classification ← correct
       else
              Classification ← Incorrect
    4. CD ← CD ∪ ($x$)
(CD : Concept Description)
-----------------------------------------------------------
```

Table 2-2 IB1 Algorithm

2.5 Link Mining

Link mining is an interdisciplinary data mining technique that lies at the interface between other areas such as link analysis, hypertext and web mining, relational learning and inductive logic programming, and graph mining (Getoor & Diehl, 2005). Some of the earliest work on link mining as an application of machine learning grew out of topical tracks on link analysis at research meeting such as the AAAI Fall Symposium on AI and Link Analysis (1998). For categorizing the link mining tasks there is taxonomy of common link mining tasks described by Getoor (Table 2-3).

In this research, my work is more related to the link prediction task. In this area, the existence of a link between two or more instances can be predicted based on properties, such as shared relational features, that are not commonly available in some domains. The main challenge is using suitable data properties and techniques to improve the link prediction capability. Therefore, different techniques are proposed to address this problem. In addition, the graph structure induced by links between entities can be considered as an important property in some domain which can improve the link predication. Popescul and Ungar (2003) used statistic learning to build robust model from noise data and relational database and applied it to the citation prediction in the domain of scientific publications. In another research, Taskar, *et al.* (2003) applied a relational Markov network framework in the graph links to predict link existence in the domain of web pages such as the advisor-advisee link between a professor and a student, and in a social network domain.

1. Object-Related Tasks (a) Link-Based Object Ranking (b) Link-Based Object Classification (c) Object Clustering (Group Detection) (d) Object Identification (Entity Resolution)
2. Link-Related Tasks (a) Link Prediction
3. Graph-Related Tasks (a) Sub-graph Discovery (b) Graph Classification (c) Generative Models for Graphs

Table 2-3 A taxonomy of common link mining tasks

However, there are some properties that have not yet been deeply explored within a link prediction model. One of my objectives is to utilize unused properties to expose unobserved associations that can improve link prediction in applicable domains.

2.5.1 Link prediction in social network

A social network is a collection of associations between individuals (e.g., people) or organizations that can be graphically represented. Links in this graph are based on one or more specific types of interdependency, such as values, visions, ideas, financial exchange, friendship, kinship, dislike, conflict or trade. The term was first coined by

Professor J. A. Barnes in the 1950s (in: Class and Committees in a Norwegian Island Parish, “Human Relations”). The information that a social network provides about each individual can be used to build prediction models for a recommended link or missing link for example. Social network services such as *MySpace* and *Facebook* allow users to create a profile which contains many aspects related to the user (e.g. lists of interests, communities, schools, and links to friends). Some services, such as Google’s OrKut, are community-centric; others, such as the video blogging service *YouTube* and the photo service *Flickr*, are related to social media. In other kinds of services such as Six Apart’s *LiveJournal* and *Vox*, they are organized around text-and-image weblogs.

Some studies such as Hsu, *et al.* (2007) use a friend’s network of LiveJournal to predict friendship based on graph features. Other studies such as thoes by Liben-Nowell and Kleinberg (2003) and Popescul and Ungar (2003), define certain linkage measures to estimate the existence probability of a potential future link. In order to use the co-occurrent property through association rules, Schmitz, *et al.* (2006) propose a method of using association rules in Folksonomies[1] as a recommended system (such as tags, users, or resources). In my research, I use association measures (based on users co-occurence) of some users’ properities as link prediction features (Aljandal, *et al.* (2008).

2.5.2 Link prediction in bioinformatics

Bioinformatics is a new field of science resulting from combining different disciplines: biology, computer science, and information technology. The primary goal of this field is to understand the biological processes using different techniques in computer and information science. Understaning the associations, structures and patterns in the huge amount of biological data are the most important tasks in this field (Chen, 2005).

Link prediction methods can be used to provide an expectation of unknown relations which come from the massive amount of data related to gene expression, known regulatory relationships, RNA, protein sequences, and and protein interaction. Association rule mining has been used in this area for discovering associations between different concepts with different structures. In order to discover association

[1] A folksonomy is a collaborative tagging system allows users to assign (arbitrary) tags to resources.

rules, researchers have investigated some special algorithms to handle bioinformatics dataset for discovering frequent patterns (Pan, Cong, Tung, Yang, & Zaki, 2003). Cong, *et al.* (2004) introduces different algorithms to mine frequent closed patterns and propose new one. *GenMiner* is an implementation of a special association rule generater from genomic data which uses an algorithm called NorDi which is more efficent than the Apriori approch as shown in (Martinez, Pasquier, & Pasquier, 2008). Other studies by Jiang and Gruenwald (2005) propose a new data structure, BSC-tree and FIS-tree, to prepare the gene exprestion data for the association mining step.

The structure of association rules can be adobted to meet representtational requarments in the area of bioinformatics. Hoan, Satou and Ho (2004) study the association between gene regulator network and transcriptional regulatory modules by using association rules with the form *factorset* → *geneset* . In other research, Creighton and Hanash (2003) use association rules of the form *Gene1* → *Gene2* with only support and confdence measures, they restrict their rule generator with some criteria to discover rules with one gene on the left-hand side (LHS) of each rule and seven or more on the right-hand side (RHS). Similer to previous rule structure, McIntosh and Chawla (2005) and McIntosh and Chawla (2007) extract interesting gene relationships from microarray data by using association rule with support and confidence measures with using MAXCONF method for generating rules. MAXCONF is not related to a rule's measures, but is related to the rule's structure where consequence of discovered rules are not subset of other rules. Discovering relations between genes motifs and cell types disccused in (Thakkar, Ruiz, & Ryder, 2007) using association rules of the form *Motif1 → cell type*. For example: *M8 && M10 →neural [Supp =0.27, Conf = 0067,]* where *M8* and *M10* are motifs and *nural* is cell type.

The problem of protein-to-protein interaction modeling has given rise to several research studies for predicting positive interactions and other related properties. Hao, *et al.* (2004) discussed how to identify distinstive patterns to extract protein-protein interactions from biological literatures using dynamic programming algorithms. In interaction prediction, Deng, Sun and Chen (2003) consider two methods: the neighborhood-counting method and the chi-square method. They use protein-protein interaction network to predict protein function. Another approch using a common-neighbor-based model and a Bayesian framework to predict protein function is proposed by Lin, *et al.* (2006). Mixture-of-Feature-Experts method has been used in

(Qi, Klein-seetharaman, & Bar-joseph, 2007) to predict protein-protein interaction where they combine a set of features as a mixture of experts.

Previous work on applying association rules techniques to protein-protein interaction has primarily been devoted to building predictive rules of identifying function regions pairs engaged in protein-protein interactions (Hung & Chiu, 2007), (Oyama, Kitano, Satou, & Ito, 2000), (Oyama- Takuya, 2002). Recently, new frequent pattern identification techniques specific to protein networks have been proposed by Turanalp and Can (2008) that were used to find patterns for predicting protein-protein interaction, specifically recurring functional interaction patterns. Kotlyar and Jurisica (2006) integrated association mining approach to integrate several diverse types of evidence. Features of primary structure and associated physicochemical properties were used by Oyama, Yoshida, *et al.* (2003) and gene expression profiles, as features, were considered by Oyama- Takuya (2002) with large number of protein network and difference among them. The concept of deferential association rule mining was introduced by Bock, *et al.* (2001) and Besemann, *et al.* (2004).

In my research, I consider the protein-protein interaction network and use numerical features to predict protein-protein interaction from only the parent-child relationships. However, using further information such as gene expression and other protein features, we can improve the prediction module.

2.6 The K-means clustering method

K-means clustering is an algorithm to classify or to group a number of objects based on attributes/features into K number of group. K is a positive integer and the grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid (the center).

The K-means algorithm is:

1) Decide on the value of K.
2) Start off with K arbitrary centers. They may be chosen randomly, or as the centroids of arbitrary starting partitions of the case set.
3) Consider each case in sequence and find the center to which the case is closest. Assign the case to that cluster. Recalculate the center of the new and old clusters as the centroids of the points in the cluster.
4) Repeat until the clusters are stable.

5) Repeat for different initial centers. Choose the best clustering, in terms of minimum within cluster sum of squares.

In this research I use the K-means clustering in the evaluation phase of the information gain framework section to demonstrate a significant measures' values distribution improvement by visualizing the result.

The relation measurement has been involved in many data mining techniques such as association rule and classification. Capturing a real association between items is the primary goal in previously described methods. However, interestingness measures can adapt any properties to improve prediction quality. The concept of Itemset size and its implication can drive further relation information.

2.7 Concept Hierarchy (Ontology)

A concept hierarchy or ontology is an explicit description (similar to the formal specification of a program) of the concepts and relationships that exist in a domain (Gruber, 1994). Ontologies can be seen as metadata that are used to provide a better understanding of the data. In my experiments, I use a dataset from *LiveJournal,* social network service that provides personal and social information about its members-publicly by default. In social networks, ontologies can provide a crisp semantic organization of the knowledge available in the domain. In particular the interest ontology can be used to make explicit the relationships between various interests, thus helping the process of understanding the data. They can also be used to improve the predictive power of the classification algorithms, which otherwise base their decisions only on the statistical information in the data.

Previous work by Hsu, *et al.* (2006) has shown that the accuracy of predicting friendship relationships in a social network is very low if common interests are used as features and no graph features are available (Hsu, King, Paradesi, Weninger, & Pydimarri, 2006). However, the accuracy can be improved if an interest ontology is exploited (Bahirwani, Aljandal, Hsu, & Caragea, 2008) when constructing features using association rule measures.

The hierarchical structure of the ontology enables us to view the connections between interests at different levels of abstraction. For instance, a user mentions "computers" to be one of his interests and other user lists "laptops" in her interest list. In the absence of interest's ontology, the algorithm for finding association rules considers “computers” and “laptops” to be different interests, obviously a semantically incorrect assumption. By analyzing the connection between these two interests in the interest ontology, the knowledge that “computers” and “laptops” refers to the same concept can be inferred and incorporated in computing association rule measures. Thus, I explore the use of the ontologies when constructing features using association rule measures and investigate the improvement in performance of various classifiers that use the resulting features to predict friends.

CHAPTER 3 - **ITEMSET SIZE: CONCEPTS AND ADAPTIVITY**

In this chapter I will present the concept of itemset size-sensitivity and a specific method for achieving it. Then I will describe the needs and techniques of using this concept in interestingness measures with respect to domains adobtivity.

<u>Notation</u>

C	Constant	R_i	normalization factor
n_i	itemset size	q	number of target items
m	constant (e.g., minimum itemset size)	$\hat{p}$	propose probability

----

3.1 Itemset size-adaptive interestingness measures

In a preliminary study, I investigated the problem of tuning and selecting among interestingness measures for association rules. I derived a parametric normalization factor for such measures that address imbalanced itemset sizes, and show how it can be generalized across many previously derived measures. Itemset size is a property that has not been involved in any of interestingness measures yet.

3.1.1 Need for sensitivity to itemset size

Discovering and predicting relations between items in a set of transactions (each one denoted by an itemset) is a technical objective driven by prior background knowledge. Some data properties are directly observable such as co-occurrence of items, others can be hidden behind the data behaviors. Itemset size is one of these properties that can give additional deep information about associations between items, which can turn expose more hidden relations.

In some real-life data, joint probability may not reflect the deep relationship between items if there is significant variability in the number of items in each itemset. This property will be more effective in some domains, where itemset size reflects the intrinsic properties of a domain and is not affected by other characteristics of the domain. One limitation of existing binary measures of rule interestingness is that they do not account for the relative size of the itemset to which each candidate pair of associated subsets (X, Y) belongs. Moreover, there are some hidden associations related to the appearance of candidates in small groups. Therefore, giving some attention and weight to these small groups may lead us to a different perspective on relationship between items. This kind of data behavior can be seen, for example, in domains such as social network where each user's record consists of features such as interests, communities, schools attended, etc. In particular, one of these feature sets is a user's list of interests, each of which corresponds to a list of interest holders. Some interests such as "DNA replication" have low membership; whether this is because the interests are less popular or more specialized, it often suggests a more significant association between users naming them than between those who have interests, such as "Music," "Art" or "Games" in common. In general, an extremely large number of interest holders tends to correspond to a more tenuous link. The size of the itemsets produces further information that can be used to increase the sensitivity of interestingness measures applicable to a candidate association.

3.1.2 Itemset size in statistical probability

The size of itemsets has been used in statistical probability, specifically in randomization theory. Therefore, I am going to discuss the problem of estimating the property of relation existence in a random data set and how Itemset size can affect the result.

Let L = $\{x_1, x_2, \ldots, x_k\}$ be a set of k distinct instances (items). Let D be a data set that consist of itemsets T_i $1 \leq i \leq M$ each of which a set of instances such that $T_i \subseteq L$

Let T_1 and T_2 $\in$ D (itemsets) with size M and N respectively where M > N

$T_1 = \{x_1, x_2 \ldots x_M\}$

$T_2 = \{ x_a, x_b \ldots x_N\}$

Where each of $x_i \in L$ and $1< i < M$ and $a, b \in [1, N]$

If there is a p where $1< p < M$ and on instance $x_p \in T_1 \cap T_2$ such that:

$$\exists\, x_p \rightarrow x_q \in E \quad \text{where E is a set of existing links and } x_q \in L$$

Then the prior probability of finding x_q in $T_1 = {}^{1}\!/_{M}$ (without any further information) and finding x_q in $T_2 = {}^{1}\!/_{N}$ (without any further information)

As a result, the probability that we find x_q in T_2 is higher than the probability of finding x_q in T_1 because M > N. However, this assumption is derived from probability theory without taking into account any further information about the instances (items) just to show how itemset size can be involved in probability.

3.1.3 Normalization task

Normalization knowledge has been used to discover the correlation between a single numeric feature and multiple intermediate concepts. This concept will make a difference to the results' order, which improves the measure quality. Normalization knowledge reduces unrelated correlations making axis-parallel division in the instance space more useful (Steven, 1996).

The main concept of the current objective measures is based on the probability. When Hilderman, *et al.* (1998) proposed a concept of share-confidence and support, they believed that accounting for the quantity and price of the items in the computational of confident and support would improve measurement quality. In my research, the itemset size property can be a part of association measures to improve association capture in applicable domains.

A normalization step is used to sensitize association measures to the popularity of a specific property, which is measured by the sizes of itemsets. Intuitively, it is more significant for two candidate instances to share rare properties than popular ones, a property which gives itemset size a particular semantic significance in different application domains.

The main idea about using Itemset size in association measures originates from modifying the joint probability to reflect the significance of sharing common properties. The process starts from computing a normalized factor for each itemset based on their size. Then I compute the joint probability of any target items and, instead of counting the number of itemset that contain the target items, we add up the normalized factor of itemsets that contain the target items. Therefore, the degree of involvement of each itemset is related to its size; the smaller their size the greater their significance in the resulting joint probability estimates. This allows us to modify the joint probability and hence the association rule interestingness measures by substituting on itemset size-sensitive joint probability. Therefore, we measure the interestingness using a degree of importance of each itemsets based on their size.

3.2 Validation-Based Approaches to Normalization

Joint probability has been used as an important part in the interestingness measures. In data mining applications, this may not reflect the true meaning of the relationship between some items if there is a large variance among the number of items in each itemset. Frequent items in small-sized itemsets may be more informative about the relationship between their constituent items than the large ones. We now consider how to extend the interestingness measures to take into account the size of the itemset to construct a new concept of size-sensitive probability and drive a parametric function that can be used in any measures (Aljandal, *et al.* 2008). In addition, we can automatically tune these functions by using various optimization approaches.

3.2.1 Parametric Functions

In this section I describe the normalization function that I derived from the size's relation between itemset sizes.

Let m be a constant such as the minimum given itemset size (we can also use a trim-mean[2]), so for each itemset with size n_i there is a real number $C_i \geq 1$ such that:

$$m^{c_i} = n_i$$

$$m = \sqrt[c_i]{n_i} = n_i^{\frac{1}{c_i}} \therefore m \propto C^{-1}$$

Let $R_i = 1 / C_i$. Then:

$$m = n_i^{R_i}, \qquad 0 < R_i \leq 1 \tag{1}$$

$$R_i = \frac{\log m}{\log n_i} \quad \rightarrow \quad R_i = \log_{n_i} m \tag{2}$$

R represents a relational factor that describes the relationship between m and the size n_i of each itemset. Moreover, the value of R_i will become more efficient if we involve the number of target items in the equation. For example, if the target items are (x_1, x_2, x_3) the value of R_i for itemset K should be slightly larger than the value of R_i

[2] Trim-mean is the average which can be obtained by trimming the largest and the smallest cretin percentage (this percentage can vary) of the numbers in a series and then calculating the arithmetic mean for the remaining numbers.

of the same itemset when the target items are (x$_1$, x$_2$). Therefore, the q^{th} root is used to adjust the value of R_i based on the number of target items q in $X \cup Y$. Then:

$$R_{i_q} = \sqrt[q]{\log_{n_i} m} \qquad (3)$$

If we consider R_{i_q} in calculating the joint probability, we can define an item set (size)-sensitive joint probability. Let $L \equiv \{x_1, x_2 \dots x_k\}$ be the set of items. Let D be a set of transactions ($|D| = N$), where each transaction T is a set of items such that $T \subseteq L$. Then:

$$\begin{aligned}\hat{p}(x_1, x_2, \dots, x_q) &\triangleq \frac{1}{N}\sum_{i=1}^{N} R_{i_q} \\ &\triangleq \frac{1}{N}\sum_{i=1}^{N} \sqrt[q]{\log_{n_i} m} \qquad (4)\end{aligned}$$

The lower bound of this equation is achieved where the number of items q in one itemset is 2, which is also the smallest possible itemset size. In the case where another constant has been used (such as trim mean) we will consider all itemset sizes less than or equal to the constant have a value similar to the value of the constant.

3.2.2 The Curve of Itemset Size-sensitive Normalization Function:

Based on Equation 3, Figure 3-1 illustrates the curve of the normalization factor under three assumptions of the size of the target items. When the size of itemset $n_i = 5$ the value of $R_{i_q} = 1$ (the max for R_{i_q}) which is exactly equal to the normal value when we compute the normal joint probability. Moreover, R_{i_q} is positively correlated with (and sublinearly proportional to) the size of target itemsets. We can see from Figure 3-1 that the value of R_{i_q} increased with the increasing of the number of target items q which makes R_{i_q} approach 1. This provides a logical explanation of the relation and interpretation of the itemset size and the target itemset size in the new normalization factor equation.

Let m = 5 and $5 \leq n_i \leq 100$

$$R_{i_2} = \sqrt[2]{\mathrm{Log}_{n_i} m}$$

$$R_{i_3} = \sqrt[3]{\mathrm{Log}_{n_i} m}$$

$$R_{i_4} = \sqrt[4]{\mathrm{Log}_{n_i} m}$$

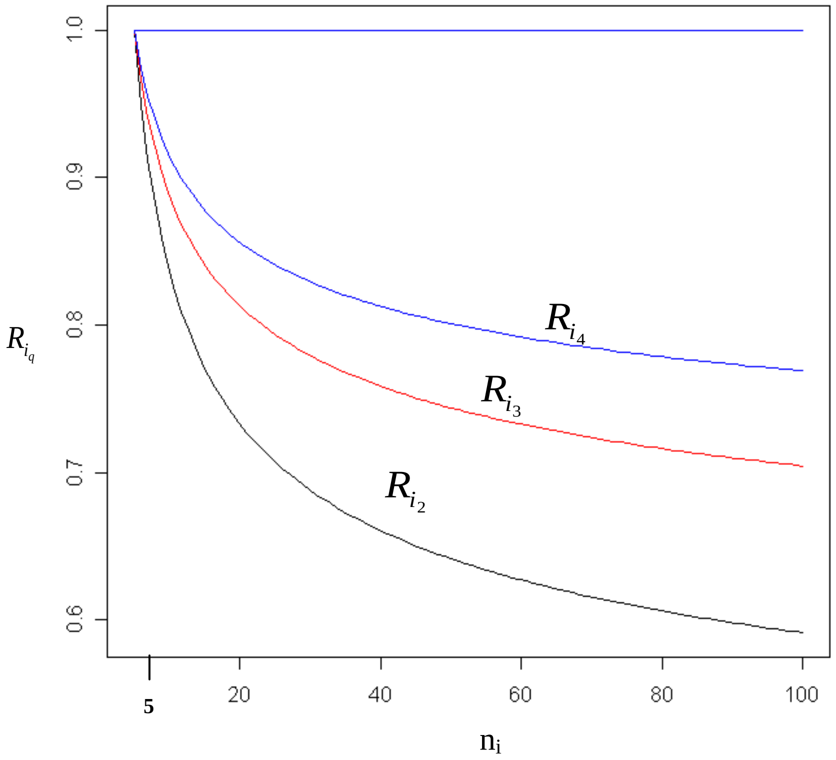


Figure 3-1 The curve of the normalization function

3.2.2.1 Example 1:

In this example I present a small data set of a social network in Table 3-1 to demonstrate the effect of the new normalization factor (R_{i_q}) and explain how this factor changes the association rules interestingness measure.

In this example the smallest itemset size is 2 (*m* =2) and the target itemset is also 2 (*q* =2). The first step is to compute the normalization factor for each row of data (basket) based on Equation 3 and the constant given.

We will consider only some of the frequent items :

P(User20) = 0.8

P(User2) = 0.6

P(User9) = 0.6

#	Items list	Size (n)	R_i
1	User20, User10, User5, User8, User11, User9, User12, User15, User18, User22	10	0.549
2	User2, User4, User20	3	0.794
3	User20, User10 ,User11, User9, User12, User21, User16, jucie3	8	0.577
4	User2, User4	2	1
5	User14, User9, User2, User20, User16, User10	5	0.656

Table 3-1 Market basket (example 1)

The next step is compute the proposed joint probability based on Equation 4 where the target items = 2. Then we calculate the support and confidence of target items based on joint probability (without R) and the proposed joint probability   (with R).

Probability	Without R	With R
$\hat{p}$ (User20, User10)	0.6	0.3564
$\hat{p}$ (User2 ,User4)	0.4	0.3588
$\hat{p}$ (User9 , User12)	0.4	0.2552
Rules		
Conf(User20 → User10)	0.75 (1)	0.446 (2)
Conf (User2,→ User4)	0.667 (2)	0.598 (1)
Conf (User9 → User12)	0.667 (2)	0.375 (3)

Table 3-2 Rules and measures (example 1)

$\hat{p}$ = probability (Support), Conf = Confidence. ( ): order number

From the Table 3-2 we can see how the new definition of size-sensitive joint probability affects the order of the confidence for three association rules. The value of Conf(User20 → User10) has higher probability without considering the new normalization factor (R), but when we use the new size-sensitive joint probability the

$\hat{p}$ (User2, User4) reduced because User20 and User10 appear in a large itemset size compared with the appearance of User2 and User4 which gives them more indication and a higher size-sensitive joint probability.

3.2.2.2 Example 2:

The second social network data set Table 3-3 illustrates the effect of a new normalization factor when there is no distinguished different in the size of itemset (*n*).

Now *m* = 3 for computing the normalization factor. The same steps we follow as the example.

#	Items list	Size (n)	R_i
1	User20, User11, User9, User3, User7	5	0.826
2	User7, User14, User20, User8	4	0.890
3	User20, User11, User22, User16	4	0.890
4	User7, User14, User9	3	1
5	User9, User3, User20, User11, User23	5	0.826

Table 3-3 Market basket (example 2)

Probability	Without R_i	With R_i
$\hat{p}$ (User20, User11)	0.6	0.5084
$\hat{p}$ (User7, User14)	0.4	0.3780
$\hat{p}$ (User9 , User3)	0.4	0.3304
Rules		
Conf(User20 → User11)	0.75 (1)	0.636 (1)
Conf(User7,→ User14)	0.667 (2)	0.630 (2)
Conf(User9 → User3)	0.667 (2)	0.551 (3)

Table 3-4 Rules and measures (example 2)

From Table 3-4, we can see that the new normalization factor will not make a big difference in the order of the three rules because there is no big distinguishable difference between the sizes of itemset. However, the sensitivity of the defined joint probability makes little difference in the order of the three rules.

From the previous small examples we can draw a clear picture about the effect of the itemset size-sensitive joint probability. In the next chapter I present a number of experiments designed to demonstrate this effect in real datasets in different domains.

3.2.3 Optimization-based approaches

In the previous section, I present a parametric function for Itemset size as normalization factor. However, changing the threshold (parameter "*m*") in parametric function (Equation 4) can change the performance measures. In this section I show the affects that will occur when we use different constant values for parameter *m*.

In parametric function optimization test, I use a *Livejournal* dataset that contains ~6000 user pairs with 10 association measures (numerical measures) for both users' common interests and common communities. More details about this dataset in Section 5.3.3 but I added one more measure, a *Φ*-coefficient. I produced 119 different classification files that contain normalized measures where each file uses the normalization factor with *m* = {2, 3, 4 ….. 120}. The first test is using Conviction of common community as a friendship prediction feature. From Table 3-5 we can see the frequency of communities' membership.

Community members	Frequency
1	26677
2	7623
3	3238
4	1784
5	1100
6	781
7-8	841
9-10	500
11-20	781
21-30	191
31-40	81
41-50	31
51-60	24
61-70	17
71-80	5
81-90	5
91-100	1
101-110	2
111-120	4
>120	9

Table 3-5 The frequency of communities membership size

The total number of communities is more than 43,000 and the maximum number of users in one community is 731users (only one community has this number of users). The minimum membership's size is one user which is about 26,677 communities as shown in Table 3-5. Therefore, I will ignore the communities that have only one user because they do not affect the association measures (there is no users' co-occurrence in these communities). Figure 3-2 shows a histogram of Community membership size. The majority of communities have at least two users as a member. Based on the Itemset size normalization factor concept, if we specify a proper threshold, we will reduce the effect of co-occurrences that happened in communities with a size larger than the threshold (parameter *m*).

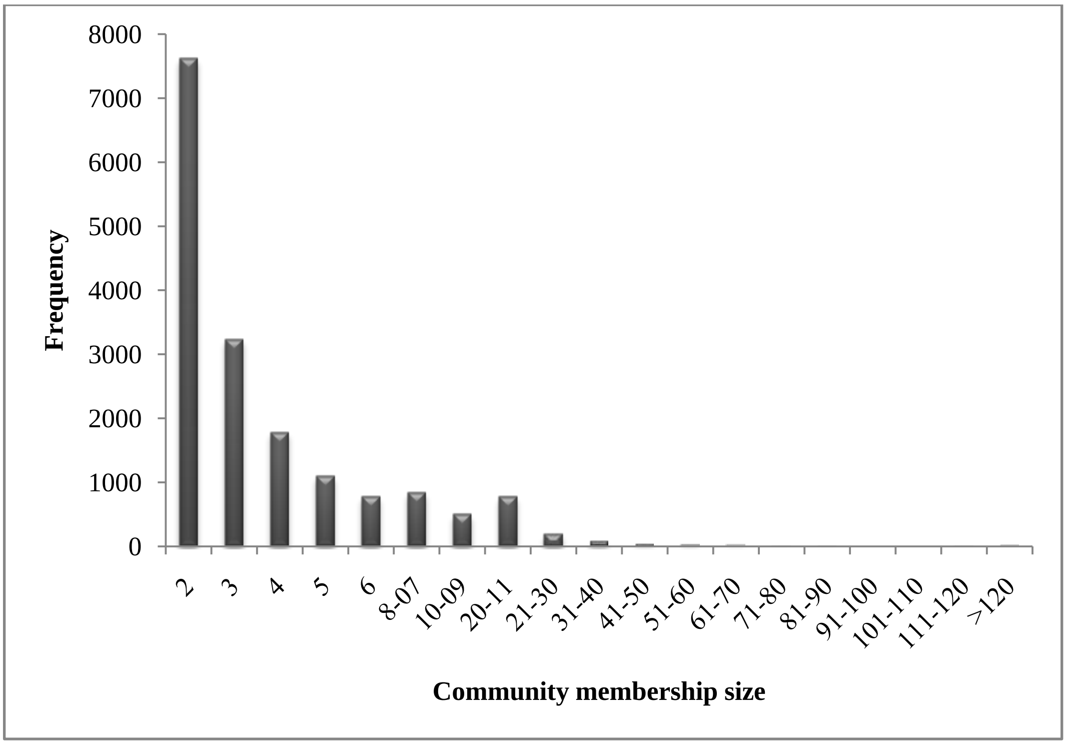


Figure 3-2 Community membership size histogram

Figure 3-3 shows the AUC result of J48 with 10-fold cross validation for each of the first 88 files for normalized and unnormalized conviction measure. The value of AUC of unnormalized conviction was drowning as a line. We can observe from Figure 3-3 that AUC of normalized conviction has a different degree of improvement based on chosen *m.* In addition, the mean of communities' membership size is 2.45 and the standard deviation equals 8.2. Thus, we can concentrate on selecting a proper *m* from the area around the mean with a range close to the standard deviation; this should provide a high probability that we find an optimal threshold (*m*).

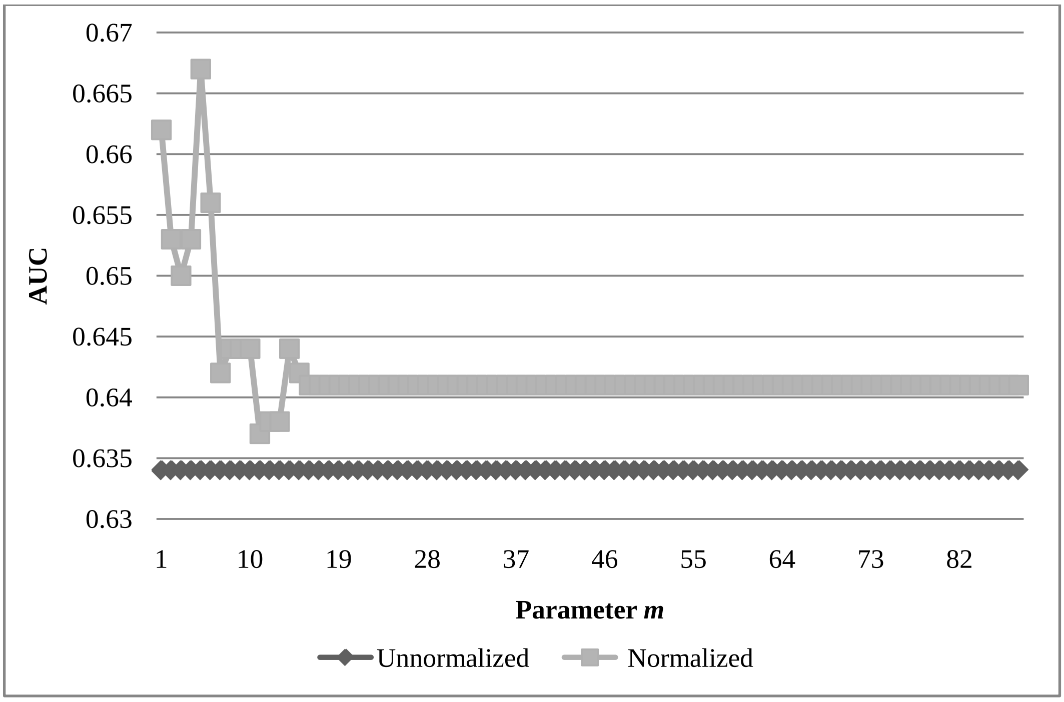


Figure 3-3 AUC's result for each file using Conviction of common communities

On the other hand, the AUC value of using normalized conviction measure will get close to the value of unnormalized conviction measures when parameter *m* reaches the maximum community membership size.

In another test, using Φ-coefficient (Linear Correlation Coefficient) measure which is defined as:

$$\Phi\text{-coefficient} = \frac{\mathrm{P(AB)-P(A)P(B)}}{\sqrt{\mathrm{P(A)P(B)P(\rightarrow A)P(\rightarrow B)}}}$$

With the J48 decision tree inducer and using the same dataset, I obtained AUC as shown in Figure 3-4, which yields the same trends relative to the choice of *m*, but with different strengths.

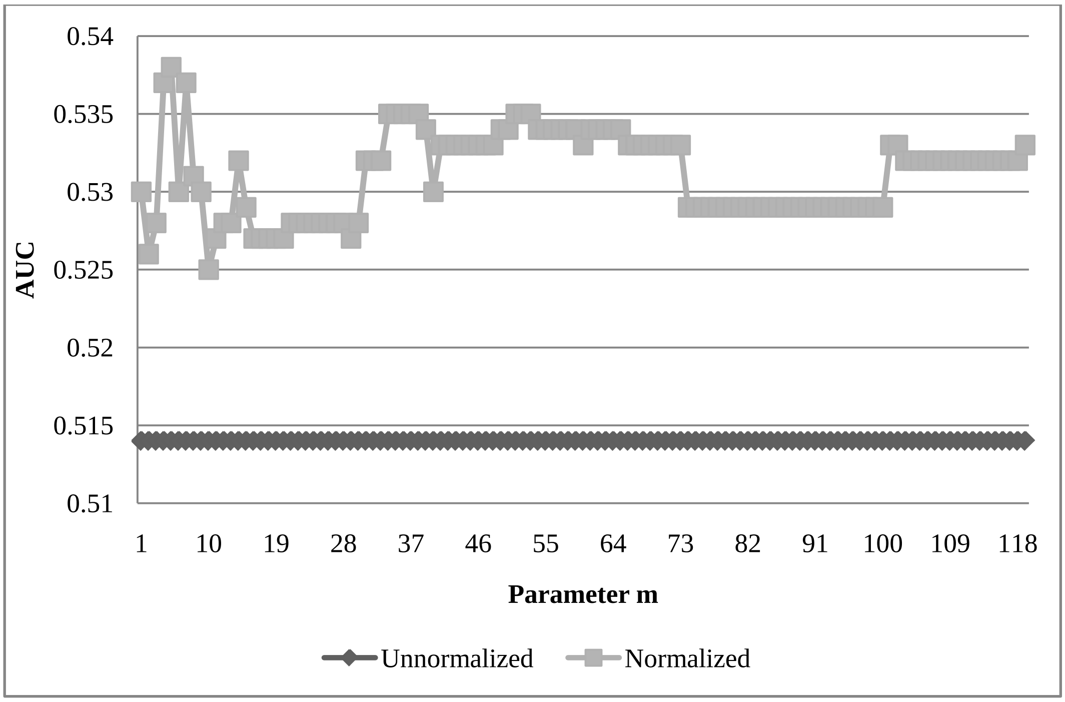


Figure 3-4 AUC's result for each file using Φ-Coefficient of common communities

Optimizing the parametric function using a proper threshold can be obtained if we concentrate in the size of small itemsets. Taking the average or the mean of the small itemsets are easy ways to get closer to the optimal result. However, we can involve parameter *m* in the process of features selection using a genetic algorithm (GA).

3.3 Role of association rules in link mining

The main objective of the association rule is discovering the associations between instances which can be measured and filtered later on. Capturing a relation is similar to finding a link between instances. Association measures are descriptive statistics computed over rules of the form $u \rightarrow v$. This allows us to apply algorithms for association rule mining based on calculation of frequent itemsets (co-occurrence), which, by analogy with market basket analysis, denote sets of instances who share a specific property. Some research focuses on such property, such as Ganiz, *et al.* (2006) and Shanfeng, *et al.* (2005), to improve link prediction.

Using association rule mining concepts and measures in link mining can be summarized as follow: if two instances X, Y are co-occurring in many cases (*P(X, Y)* is high enough to consider), we can predict that there is a link between X and Y and the significance of this link can be measured by using some association measures. Getoor *et al.* (2001) observed that there are often correlations between the attributes of entities and the relations in which they participate in. For example, in a social network, people who have the same hobbies are more likely to be friends. Therefore, we can use association rule measures as numerical features for building a link prediction module. In addition, each association rule measure captures one or more desiderata of a data mining system: novelty (surprisingness), validity (precision, recall, and accuracy), expected utility, and comprehensibility (semantic value).

3.4 Domain-specific properties and semantics

Many objective association rule measures are proposed to meet different users needs based on domain characteristics. However, there are many domains that have different properties and semantics. This is one of the reasons that some objective measures can not be consistently better than others in all application domains (Tan, Kumar, & Srivastava, 2002).

The concept of membership relation in a specific domain can be affected by external exogenous variables. Increasingly the number of effects which can not be captured as data will lead to a weak prediction result in this domain. For example, in market basket customer shopping habits and other personal effaces are either hard or impassible to capture. In the case where there are many effects, data properties will not have a stable impact which can be taken into account in a mining process to improve prediction measures quality.

We can categorize domains based on some characteristics:

Non Autonomous

In this category the instance does not have any control for being in a specific group. For example an item in a market basket has external instance which is a customer who controls the shopping trip. However, having an external instance gain extra causes an item to become a member of a specific basket. This kind of data will have some weakness in the mining process if we concentrate in data properties. A well-known example of this category is market basket data.

Semi-Autonomous

An instance of this category has some control for being in a specific group which is sometimes controlled by the neutral. Therefore in general, the number of effects in this category will be less than the first category. Dataset examples of this category are Bio-informatics, medical information (disease dataset), and demographics dataset.

Autonomous

This category contains domains where the instance has more control for being a member of specific group. In this category data properties will not be disturbed by that much of external effects like the first category. For example, in social network datasets, a user decides to be a member of a particular community. Example of data sets this category are social networks and criminological link analysis.

In my research I use three different domains to cover the three categories:

- Market baskets
- Protein-protein interaction
- Social networks

CHAPTER 4 - **APPLICATION DOMAINS**

In this chapter I present some aspects related to application domain and how it is related to my research.

There are several studies about the domain-orientation in software engineering such as Domain-Specific Modeling (DSM) (Cook, 2006) and Domain-Specific Information Retrieval (IR) (Kang, Lin, Zhou, & Guo, 2007), but the main concept of these studies lie in the domain topic or domain specification. In the area of data mining, monitoring data behavior is one of the most important goals. Therefore, in association rule mining we need to provide more details about domain characteristics and their effects on data properties to take advantage of usefulness of these properties.

Before we go further, I will define the general concept of the data. I use the words “instance” and “group” to indicate the item and basket in market basket dataset for example. The main goal of association rule measures is to capture the association between instances by using their appearance in each group. However, an instance’s memberships are not always independent from other effects. Therefore, looking behind the circumstances that make two or more instances share a membership of one or more groups gives an explanation about their data behavior.

Increasing the number of exogenous variables which can not be captured as data will weaken some properties effects on link prediction. One example is market basket analysis, which reflects customer shopping habits and other personal property which are either hard or impassible to capture such as: quick shopping trip and specific shopping trip for BBQ. In the case where there are many effects, data properties will not have a stable impact that can be taken into account of the mining process to improve prediction measures quality.

In the case of the itemset size-sensitive property, as implemented using validation-based normalization of interestingness measures, the effect fails to improve the precision and recall of association measures in some domains. As I list in Table 4-1, there are exogenous (potentially latent, i.e., hidden) variables existing in some domains that affect both itemset size and link existence, hence the reflection of link strength in interestingness measures.

	Domain	Exogenous variable	Effect
1	Market basket	Shopping mode	Mode determines co-purchases analysis (trip type) as well as number of items
2	Click stream	Search mode	Mode determines search arguments and other choices
3	Advising	Temporal Relation spread out over time	Concept drift Advisor/advisee topics can differ (even intentionally)
4	Co-authorship	Discipline	Some have more authors per paper
		Inter disciplinarity	More authors for more diverse topics
		Hidden relationship	Funding and co-worker relation
		Historical context	People who wrote together before are likely to do so again

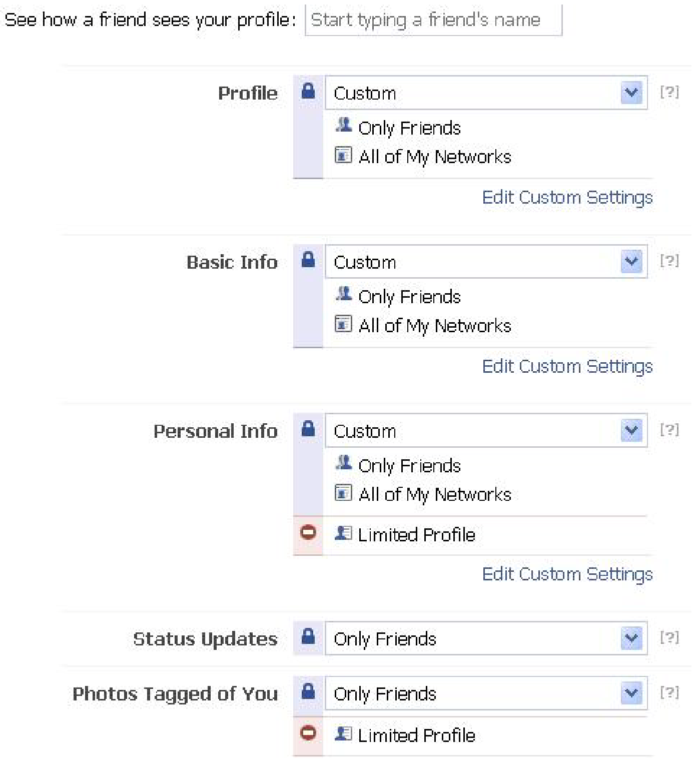
Table 4-1 Exogenous variables in various domain effects

In the next sections I will describe some domains and their properties.

4.1 Social Networks

Most social networking services include friend-listing mechanisms that allow users to link to others, indicating friends and associates. Friendship networks do not necessarily entail that these users know one another, but are means of expressing and controlling trust, particularly accessing private content. In blogging services such as SUP’s *LiveJournal* or *Xanga*, this content centers on text but comprises several media, including: interactive quizzes, voice posts, embedded images, and video hosted by other services such as *YouTube*. In personal photograph-centric social networks such as News Corporation’s *MySpace*, *Facebook*, Google’s *Orkut*, and Yahoo’s *Flickr*, links can be annotated (“How do you know this person?”) and friends can be prioritized (“top friends” lists) or granted privileges as shown in Figure 4-1.

Some vertical social networks such as *LinkedIn*, *Classmates.com*, and *MyFamily.com* specialize in certain types of links, such as those between colleagues, previous employers and employees, classmates, and relatives. As in vertical search and vertical portal applications, this specialization determines many aspects of the data model, data integration, and user knowledge elicitation tasks.



See how a friend sees your profile: Start typing a friend's name

Profile: Custom [?]
Only Friends
All of My Networks
Edit Custom Settings

Basic Info: Custom [?]
Only Friends
All of My Networks
Edit Custom Settings

Personal Info: Custom [?]
Only Friends
All of My Networks
Limited Profile
Edit Custom Settings

Status Updates: Only Friends [?]

Photos Tagged of You: Only Friends [?]
Limited Profile

Figure 4-1 Facebook's access control lists for user profile components. © 2008 Facebook, Inc.

For example, *LinkedIn*'s friend invitation process requires users to specify their relationship to the invited friend, an optional or post-hoc step in many other social networks.

Friendship links can be undirected, as in *Facebook* and *LinkedIn* (requiring reciprocation, also known as confirmation, to confer access privileges) or directed, as in *LiveJournal* (not necessarily requiring reciprocation).

In my research I use *LiveJournal* dataset where the links present a friendship relation between users. For the itemsets I use two user's properties: users' interests and communities' membership.

4.2 Protein-Protein Interaction

There are different kinds of information related to protein that can be taken into account in studying the interaction between proteins. However, the information about protein-protein interactions are important for many biological functions and diseases. For example, the number of features can be collected from interactions between yeasts and features that characterize each protein involved in the interaction. In a research study by Oyama, *et al.* (2000) produced more than two thousand features from six different types of protein features. In addtion, protein interaction network can be a source of information to build a prediction module for unknown interactions. For example, the paper by Schwikowski (2000) and others related to the PPI Networks in Rice Blast Fungus (He, Zhang, Chen, Zhang, & Peng, 2008) are useful for investigating the cellular functions of genes.

There are many data resources as catalogs experimentally determined interactions between proteins such as:

- The Database of Interacting Proteins (DIP) (UCLA, 2008) catalogs experimentally determined interactions between proteins. It combines information from a variety of sources to create a single, consistent set of protein-protein interactions.
- Yeast Interacting Proteins Database (Kanazawa-University, 2001) A yeast protein interactome with a Genetic Network Visualization System
- Domain Annotated Protein-protein Interaction Database DAPID (BioXGEM-Lab, 2006) database of domain-annotated protein interactions inferred from three-dimensional (3D) interacting domains of protein complexes in the Protein Data Bank (PDB)

In my research, I use a structural property of protein-protein interaction to predict unknown (hidden) positive interaction. The main concept is building incomplete positive protein-protein interaction network from a dataset provided by Ben-Hur and Noble (2005) and construct some associational features (numerical features) to predict the complete network. Therefore, the link of this domain is where protein *A* positively interacts with protein *B* as (A→ B). For itemset, I use the protein Parent-Child relation property in positive protein interaction network that I use to construct numerical features. The complete design of the experiment presented is in Section 5.3.5.

4.3 Other domains surveyed

There are other domain that I surveyed as possible domains for further investigation. Table 4-2 shows these domains and possible itemset that can be used to construct the numerical features, and also the possible link property that a domain can have. Some of the aspects need to be considered to work in these domains such as:

ground truth availability, dataset size and prepare data properties that can be used as a prediction base.

Domain	Itemset	Link
Movies	Type of Actors Type of actors and decade Appeared-With other actors	Appeared-With actors Appeared-With actors "Knows" -OR- Personal relation
Math Genealogy	Thesis type and year of graduation Thesis type with ontology	Advisor-of Advisor-of
Epidemiology	Diseases expertise	co-occurrence by elicitation
Citations	Bibliographies (cited by others)	A cites B
Spatial Event	Events within a radius (tagged)	Attested -OR- co-references
Temporal Event	Events with in interval (tagged)	Attested -OR- co-references
Blog community	Cluster of blog entries	Co-members at communities Interests Party affiliation position on issue
Protein-protein	Domains and other features Interaction chain sequence (PPI)	Interaction Interaction
Collaboration	Co-author list	Collaboration as recorded in Erdos number Project (Grossman, 2007)
Text and named entities	Document (named entities mentioned in bag of words)	Page links

Table 4-2 Domain property for future work

CHAPTER 5 - **EXPERIMENT DESIGN AND IMPLEMENTATION**

In this chapter, I give a detailed explanation of my overall evaluation approach, followed by the derivation of evaluation measures that I use and, at the end of the chapter, my experiments design.

5.1 Evaluation Approach

Two evaluation frameworks are used to evaluate my approach. The first one is evaluation using information gain and rule selection, used for a dataset where ground truth is not available. The second one is evaluation through classification methods which is used for datasets that have a source of ground truth.

5.1.1 Evaluation through Information Gain and Rules Selection

In this section I describe the design of my first experiment framework, which consists of three phases as shown in Figure 5-1: Local Phase, Global Phase and Evaluation Phase followed by dataset description. This framework is designed for domains that do not have a ground truth for a comparison. However, the gold standard that I consider can be rebuilt by using any independent measures.

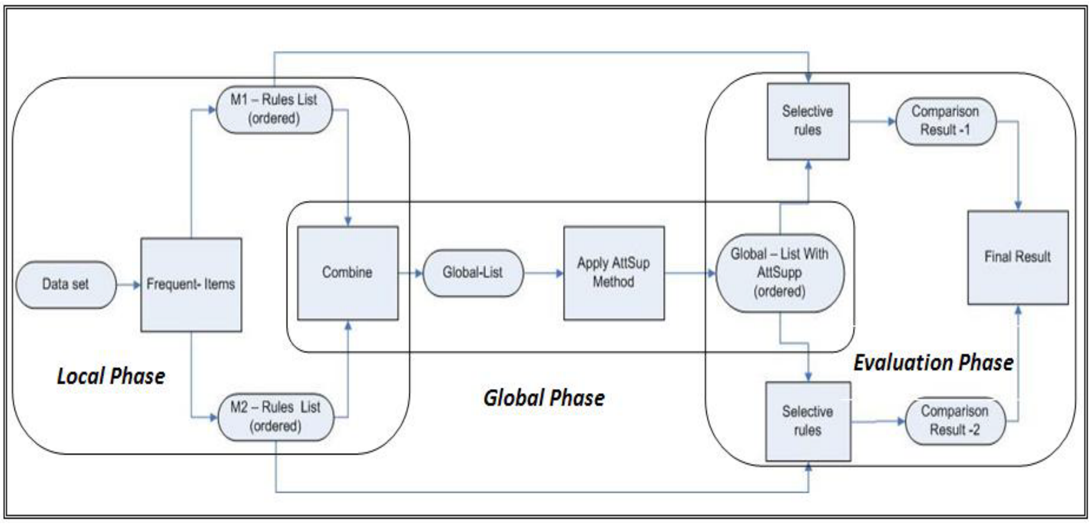


Figure 5-1 Three phases' evaluation process (using information gain)

5.1.1.1 Local Phase

In this phase I generate two list of rules based on two versions of specific association rules measured as shown in Figure 5-2.

Local Phase

The first local list:

```
-----------------------------------------------------------------------------------------------
Input: Dataset D
Output: Local list ARL-1                                // with rank
-------------------------------
Let ARL-1 = Generate association rules (D , 0.2)    // with minimum support ≥ 0.2
For each rule X in ARL-1
      Compute lift (X)
Sort(ARL-1 )                                            // based on life measure
Rank(ARL-1)
End
-----------------------------------------------------------------------------------------------
```

The second local list:

```
-----------------------------------------------------------------------------------------------
Input: Dataset D
Output: Local list ARL-2                                // with rank
-------------------------------

For each itemset T in D
       Compute the normalization factor-R ( T )
     Let ARL-2 = Generate association rules with-R (D , 0.2)     / / with minimum
                                                                      support ≥ 0.2
For each rule X in ARL-2                                //  R is the normalization factor
            Compute lift with-R (X)
Sort(ARL-2 )                                         //  based on life measure
Rank(ARL-2)
End
```

Figure 5-2 Local phase processes pseudo code

The first list is going to be generated by Magnum Opus[3] with filtering of the redundant rules with minimum support = 0.2 and ordering the result by one of the interestingness measures (Lift). The output is a list of rules ordered by the Lift measure (from high to low), as discussed in Chapter 3. The next step is to substitute the lift value with the order number (Rank) for each distinct value (1 is the highest).

[3] A pattern discovery Software from RuleQuest data mining tools (http://www.rulequest.com/)

The second list will be generated by using the new normalization factor (R) which will need to be computed from the market basket. After that I use the same interestingness measures (Lift) but this time with the normalization factor (R), which is using itemset size-sensitive joint probability instead of the normal joint probability. Then I order the result by modified lift measure (with R) which is also going to be substituted by the order number (Rank).

The result from this phase is two local ranked lists of rules. For the next phase I choose the union of the two local lists as generating criteria of the Global list.

5.1.1.2 Global Phase

Global Ranking	Attribute Surprising
Candidates Criteria	Union

Table 5-1 Global List criteria

```
Global Phase

Producing the Global list:
-----------------------------------------------------------------------------------------------
Input: ALR-1, ARL-2
Output: Global list ARL-G                                   // with rank
-------------------------------------

Let ARL-G ← ARL-1 ∪ ARL-2
For each rule X in ARL-G
      Compute the attribute surprising ( X )
Sort(ARL-G)                                        // sorted by attribute
                                                   surprising (LHS and ALL)[4].
Rank(ARL-G)
End
-----------------------------------------------------------------------------------------------
```

Figure 5-3 Global phase processes pseudo code

[4] The attributes located in the left hand side (LHS) are used for first-level sorting and all rule attributes as second-level

I will refer to the standard list that will be generated in this phase as *global* list. The main reason for creating this list is because some domains lack a ground truth such as a market basket domain. There are many criteria for generating the global list from two local lists, but in this research I am taking the union of the two local lists, as shown in Table 5-1, which have been generated in the Local Phase. This is one of the safest ways to generate a non-ordered global list and avoiding adding a penalty of a missing rule in the comparison phase.

From the previous steps I generate two lists of rules ALR-1, ARL-2 using two methods and each list has its own order (Ranking). Let ALR-1 $=\{r_1,r_2\ldots.r_n\}$ and ARL-2 $=\{r_1,r_2\ldots.r_m\}$ where n and m >1. The global list can be produced as ARL-G = $\bigcup_{i=1}^{k} r_i$ where k = m + n - z and z = | ARL-1 ∩ ARL-2 | which is removing all the redundant rules. The result will be a list of rules without any order (or values) as shown in Figure 5-3.

For ranking method, I am going to use the Attribute surprisingness (chapter 3) as standard ranking measure for the Global list:

$$\text{AttSupp} = \frac{1}{\left(\frac{\sum_1^k InfoGain\ \ (A_i)}{K}\right)}$$

Where InfoGain (A_i) is the information gain of the i^th^ attribute occurring in the rule and *k* is the number of attributes.

The first step in computing the attribute surprisingness is calculating the information gain of each attribute in the rule lists. As requiring parameter for computing the information gain, I classify the date set (market basket) based on the number of items in each instance (itemset). Let X_i be the i^th^ itemset, then the classification of X_i is as follows:

$$\text{C}(\text{X}_\text{i}) = \begin{cases} True\ , & |S_i| \geq 2 \\ False, & |S_i| < 2 \end{cases}$$

Where | S_i| is the size of the i^th^ itemset. After this step we can compute the information gain of each attribute as:

$$Gain(S, A) = Entropy(S) - \ Entropy(S|A)$$

which is:

$$Gain(S,A) = Entropy(S) - \sum_{v\epsilon\ values\ (A)} {|S_v|}/{|S|}\, Entropy(S_v)$$

Where $Entropy(S)$ is defined as

$$Entropy(S) = \sum_{i=1}^{c} -P \ \lg P_i$$

Where S is the set of example, P_i is the proportion of S belonging to class i, *Values(A)* is the set of possible values for attribute *A*, and S_v is the subset of *S* where attribute *A* have value *v* (Mitchell, 1997).

From the previous steps I produce the *global* list that is going to be ordered (Ranked) based on the Attribute surprisingness of the left hand side items of the rule (LHS) as first level and for over all items as second level ordering. I am using two levels of ordering to make the global list more accurate because there are many rules with similar LHS items.

5.1.1.3 Evaluation Phase

Compression Candidates	Number of Inversions	Rank Clustering
Top N	Yes	♦
ALL	Yes	Yes

Yes: Available ♦: beyond scope of dissertation

Table 5-2 Evaluation criteria

At this point we already have two local lists of rules with ranking and one global list of rules which has been ranked by Attribute surprisingness. The goal of this phase is to compare the ranking of the two local lists by using the ranking of the global list as a canonical list, as shown in Figure 5-4. There are many compression criteria and in the evaluation process we need to apply independent criteria that can only be affected by rule ranking, as shown in Table 5-2. In this research I am considering inversion of the rules ranking for all rules and top N rules as comparison criteria.

```
Input : ARL-1, ARL-2, ARL-G
Output:  INV-1, INV-2                         //  Total number of inversions
 -------------------------------------            for the two local lists
Let INV-1 = Compute inversion (ARL-1,ARL-G)
Let INV-2 = Compute inversion (ARL-2,ARL-G)
End
    ------------------------------------------
        Compute inversion:
              Input: (Local, Global)
              Output: N                        // the total number of inversions
          -------------------------------
                 N ←0
           For each rule X in Local
                 Find X in Global
                     N ← N + | rank of X in local – rank of X in Global |
              End
----------------------------------------------------------------------------------------------
```

Figure 5-4 Pseudocode for evaluation phase processes

I considered several other criteria that will be omitted from now on. Example include computing edit distance and inversions of the rule indexes to compare candidate, because after we get the local lists there are some rules which have the same measure value (for example: Lift). This issue makes the order of rules dependent on the position. However, these positions are in turn, based on the randomization of rule learning system (Weka[5] or Magnum Opus). Therefore, using rule indexes in the comparison process with a global rules indexes list it will lead to an insufficient result.

[5] Weka is a suite of machine learning software written in Java, developed at the University of Waikato. (Witten & Frank, 2005)

Consequently, the result from this evaluation phase will give an explanation of how one of the local lists is more surprising than another by being more similar in ranking order to global list.

5.1.2 Evaluation through Classification Methods

In this section I present the framework evaluation method using the classification measures as comparison measures to show the effect of the new method for the prediction of link existence.

The requirement of supervised learning (classification) leads us to domains that have ground truth property such as the *LiveJournal* data set. In addition, I use the classification result as evaluation measures. In this section I describe the framework steps using the *LiveJournal* as an example dataset. However, this framework is applicable to any domain for which ground truth data is available.

The main step of this framework is to add new features related to interestingness measures (numerical features) with using the original joint probability and itemset size-sensitive joint probability, which includes the new normalization factor, as a friendship prediction feature.

In this section I give more details about the experimental design phases as shown Figure 5-5 : the pre-processing phase, the classification phase, and the comparison phase.

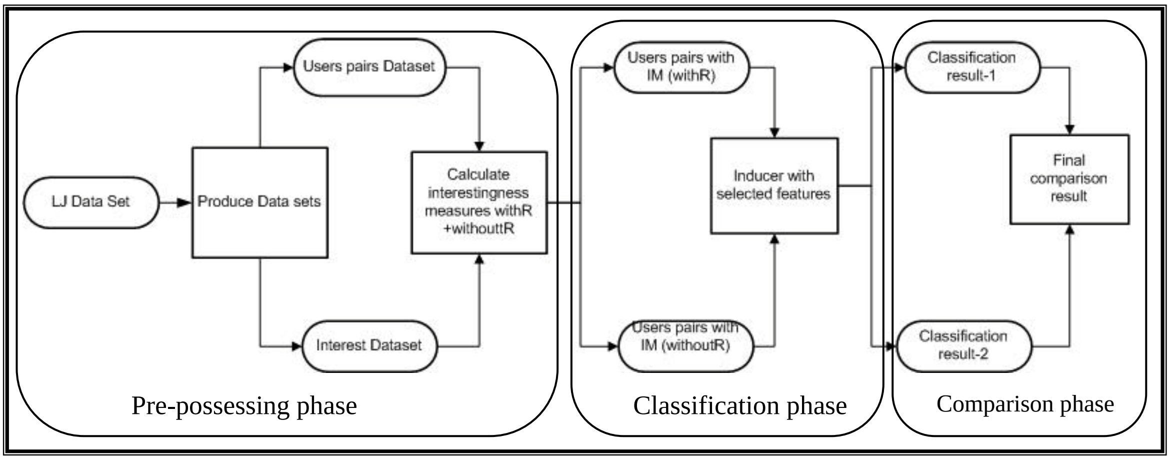


Figure 5-5 Three phases' evaluation process (using Classification)

5.1.2.1 Pre-processing Phase

In this phase I prepare the datasets that I am going to use for the next phase and compute some of the association rule interestingness measures (Support, Confidence, Lift, etc) for each instance-pair that I am going to produce as numerical features.

From the row data that we generate, a class presents the actual relation or link (e.g., friend or not). The second dataset will be a basket of one of the instance's characteristics such as communities and interests in the *LiveJournal* dataset. This basket presents the common properties between instance-pairs that are going to be a key to predicting the actual relation (link) between instances. The next step is using the basket (the second dataset) to produce interestingness measures features (numerical features) for common properties with two versions (with and without the new normalization factor).

5.1.2.2 Classification Phase

The inputs to this phase are two datasets consisting of instance pair, with and without the new normalization factor. In this phase I am going to use some of the classification methods (chapter 3.3) with selected features to show how the new normalization factor improves classification measures by using WEKA (Witten & Frank, 2005).

The output of this phase is the classification result for each inducer based on selected features.

5.1.2.3 Comparison Phase

The comparison is based on the classification result of the two classification files that has been produced using the original measures and normalized measures. In this phase the results of the comparisons are represented as tables and visualization graphs. Comparison criteria are based on the classification measures:

1. Accuracy
2. Recall
3. Precession
4. F-measure
5. ROC - AUC (for some experiment)

5.2 Evaluation Measures

In the classification tasks, the terms true positives, true negatives, false positives and false negatives are used to compare the given classification of an item (the class label assigned to the item by a classifier) with the desired correct classification (the class the item actually belongs to).

5.2.1 Precision and Recall

Precision and recall are two widely-used measures for evaluating the quality of classification results. These measures were first used to measure the information retrieval (IR) system (Cleverdon, Mills, & Keen, 1966). Recall is defined as the ratio of correct positive predictions made by the system and the total number of positive examples. Precision is defined as the ratio of correct positive predictions made by the system and the total number of positive predictions made by the system. As shown in Table 5-3, the meaning of the terms true positives, true negatives, false positives and false negatives has been used to compare the given classification of an instance (the class given by the classifier) with the actual classification (the ground truth).

		Actual result	
		-	+
Predicting result	-	TP (true positive)	FP (false positive)
	+	FN (false negative)	TN (true negative)

Table 5-3 Confusion Matrix

Formally, the definition of the Precision and Recall are follows:

$$Precision = \frac{TP}{TP+FP} \quad , \quad Recall = \frac{TP}{TP+FN}$$

5.2.2 The F-Measure

The F-measure, first introduced by Van (1979), which combines precision and recall measures, with equal importance, into a single parameter for optimization. F-measure is the weighted average of the precision and recall measures (harmonic mean), and is defined as follows:

$$F = \frac{2*(Precision\ *Recall\ )}{(Precision\ +Recall\ )}$$

5.2.3 The Accuracy

Accuracy is the ratio of the number of correct classified examples to the total number of examples.

Formally, accuracy can be defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

5.2.4 The ROC–AUC

The Receiver Operating Characteristic (ROC) curve (Fawcett, 2006) is a two-dimensional measure of classification performance that shows the tradeoff between the true positive rate and false positive rate. A ROC graph is a plot with false positive rate on the X axis and true positive rate on the Y axis, and the Area Under the Curve (AUC) ranges between 0 and 1, where 1 is a perfect score and 0 is a minimum score value.

5.3 Experiment Design

In this section I describe the design of each experiment and the datasets that I am using. The first experiment follows the first framework evaluation in Section 5.1.1 and the rest of the experiments follow the second framework described in Section 5.1.2.

5.3.1 Rules validation and selection

In this experiment, I generate the rules and validate them based on evaluation approach through Information Gain and rules Selection. Information about dataset and rules generating is presented in AR: Association rule

Table 5-4. Statistical information about these data sets is given at the end of this section.

There is diversity in choosing the datasets from the number of transactions, number of items and the limitations. Therefore, the evaluation processes will come across different data features.

The first dataset comes from Magnum Opus software as a shopping basket, containing 1000 transactions. This dataset represents the shopping baskets of anonymous stores. Each shopping basket contains sets of items totaling 16 different items. Each record (transaction) in the data set contains information about one single shopping basket. First column contain the basket-ID followed by boolean variables (0, 1) which distinguish if the item is in the basket (1) or not (0).

No	Topic	Source	Construction Method	Items	Transaction	Number of AR	limitation
1	Market Basket	Magnum opus	Original	16	1000	6151	Supp >0.001
2	Artifici al	Magnum opus	Hand-modified	13	10	28	Supp ≥ 0.2
3	Artifici al	Magnum opus	Hand-modified	10	10	44	Supp ≥ 0.2
4	Market Basket	Brijs's Dissertation	Modified Sample	303	1000	6505	Supp > 0.02

AR: Association rule

Table 5-4 Data Sets for first experiment

Figure 5-6, shows the distribution of the total number of items in each shopping basket. In addition, there is some statistical information represented in Table 5-5, which is about the number of items in each basket.

Max	8
Min	1
Average	2.58
STD	1.463986

Table 5-5 Statistics information for number of items in single basket (1st)

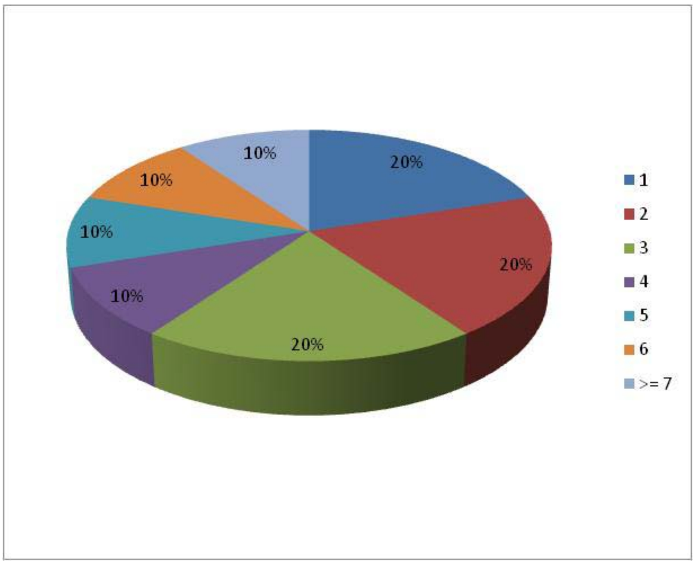


Figure 5-6 Distribution of the number of items in single basket (1st)

The second dataset is hand-modified from the Magnum Opus tutorial dataset which contains 10 transactions representing the shopping baskets. Each shopping basket contains a set of 13 different items. Each record in the data set contains information about one single shopping basket. The first column contains the basket-ID followed by boolean variables that medicated whether the item is in the basket (1) or not (0).

Figure 5-7, shows the distribution of the total number of items in each shopping basket. In addition, there is statistics are reported in Table 5-6, for the number of items in each basket.

Max	7
Min	1
Average	3.4
STD	2.065591

Table 5-6 Statistics information for number of items in single basket(2nd)

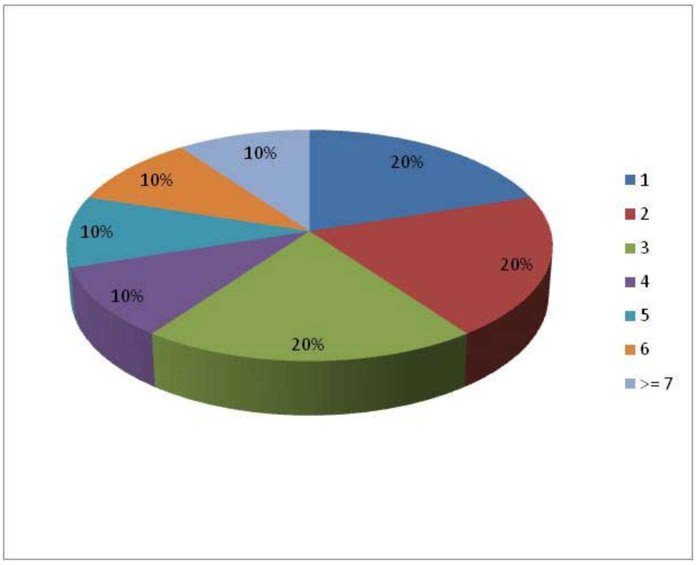


Figure 5-7 Distribution of the number of items in single basket(2nd)

The third dataset is hand-modified from the Magnum Opus tutorial dataset which contains 10 transactions representing the shopping baskets. Each shopping basket contains a set of 10 different items. Each record (transaction) in the data set contains information about one single shopping basket. The first column contains the basket-ID followed by variables that medicated whether the item is in the basket (1) or not (0).

Figure 5-8 shows the distribution of the total number of items in each shopping basket. In addition, there is statistics are reported in Table 5-7, for the number of items in each basket.

Max	7
Min	1
Average	3.2
STD	2.098

Table 5-7 Statistics information for number of items in single basket(3ed)

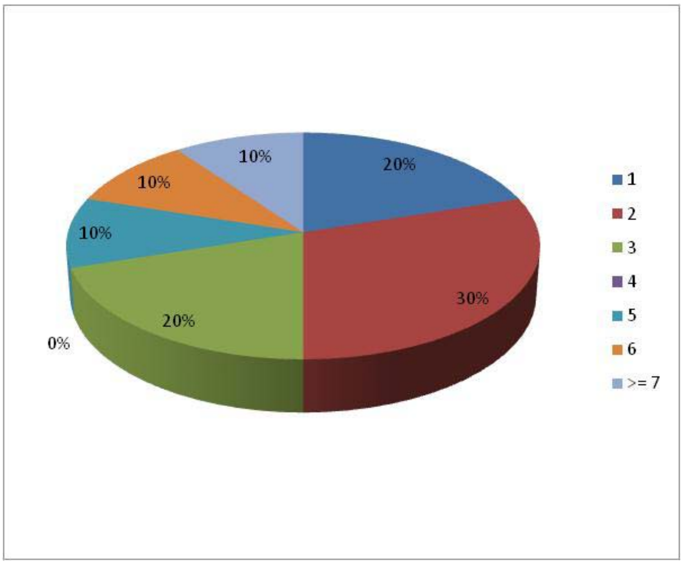


Figure 5-8 Distribution of the number of items in single basket(3ed)

The fourth dataset, which is a modified sample from Brijs, *et al.* (1999) dataset, contains 1000 transactions. This dataset represents the shopping baskets data from an anonymous Belgian retail store. Each shopping basket contains a set of 303 different items. Each record (transaction) in the data set contains information about one single shopping basket. The first column contains the basket-ID followed by boolean variables that medicated whether the item is in the basket (1) or not (0).

Figure 5-9 shows the distribution of the total number of items in each shopping basket. In addition, there is statistics are reported in Table 5-8, for the number of items in each basket.

Max	303
Min	0
Average	7.72
STD	20.11509

Table 5-8 Statistics information for number of items in single basket (4th)

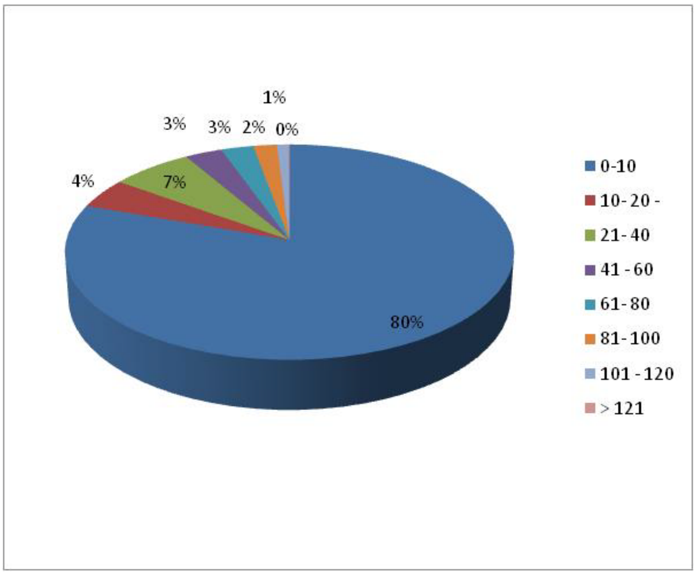


Figure 5-9 Distribution of the number of items in single basket (4th)

5.3.2 Predicting Friendship Relation using User Common Interests

The first experimental task for this framework is predicting friendship relation in social network based on numerical features. The dataset that I use is the *LiveJournal* dataset which is used in Hsu, *et al*. (2007) for link prediction based on graph features. From this study, the authors found that using mutual interests alone results in very poor prediction accuracy. Uncategorized interests in *LiveJournal* (each user indicates his/her own interests) increase the weakness of the mutual interests feature because of misspellings, or the addition of stop words such as “the” or “of,” or by adding symbols such as “underscore.” However, by using the new normalization factor we can improve prediction results over previously poor results as shown in claim-1 where E is a link existing (friendship).

Claim (1):

From the *LiveJournal* dataset we can construct feature baskets $\boldsymbol{B_i}$ such that

$$B_i \;\; 1 \le i \; \le n_\tau \quad where\ \tau \; \in \{interests, communties, schools \ldots.\}$$

$u \overset{c_j}{\rightarrow} v \quad For\ high\ c_j \quad given\ \{B_i\} \Rightarrow \quad (u,v) \in E\ with\ high\ probability$

where c_j is association rule measure for some B_i

E is a set of connected user pairs (u, v) represent the actual friendship relation.

For classification framework we need two datasets. The first dataset contains random user pairs with graph features and have two versions of numerical features one with original measures and another one with normalized measures. The numerical features will be computed using the second dataset as I introduced in the framework. The user pair’s dataset contain 10,000 user-pair where the number of actual friends is ≈2.2%. These files contain the following attributes:

1. User-Id(v)
2. User-Id(u)
3. In degree of u: popularity of the user
4. In degree of v: popularity of the candidate
5. Out degree of u: number of other friends besides the candidate; saturation of friends list
6. Out-degree of v: number of existing friends of the candidate besides the user; correlates loosely with likelihood of a reciprocal link
7. “Forward deleted distance“: minimum alternative distance from u to v in the graph without the edge (u, v).
8. Backward distance from v to u in the graph.
9. Number of common property: the total number of property that both v and u have.
10. Support of common property: the support of the rule v → u.
11. Confidence of (v, u) common properties: the confidence of the rule v → u.
12. Confidence of (u, v) common properties: the confidence of the rule u → v.
13. Lift measure of common properties: the Lift of the rule v↦ u (same as u→v).

The second dataset is a basket of users that share specific properties (ex: basket of users common interests) this will be like the market basket and will contain the following:

1. Property -Id: the name of each specific property (in case of interests these are keywords such as Art, Game, Movieetc.)
2. User-Id: List of all users that have this property.

In this experiment there are an excess of dimensions with many options for selecting features and inducers. Therefore I restricted the degree of freedom by selecting representative options in matter of coverage and range that can be feasible to access. However, the options that have "♦" symbol are beyond the experiment scope. For the inducers, I select inducers from different classifier system including the decision tree, rule generators and statistical analysis as shown in Table 5-9

Inducer	Features selection			
	Common interest	Lift measure	All IM	IM + Graph Features
B-First tree	Yes	♦	♦	♦
IB1	Yes	Yes	♦	♦
Random Forest	Yes	♦	Yes	Yes
OneR	Yes	♦	Yes	♦

Yes: Available, ♦: Beyond the scope IM : Interestingness measures

Table 5-9 Features selecting for interests in classification phase

5.3.3 Predicting Friendship Relation using Common Communities Membership

The second experiment will use communities' membership information and select proper interestingness measures to improve link prediction (friendship). In previous research, Hsu, *et al*. (2007) found that using mutual interests alone results in a very poor prediction accuracy using all inducers of their experiment because of the weakness of the interest information. In this experiment, I consider all graph features with the user communities' membership information measures as a new feature. In addition, I use the community information to construct two versions of numerical feature (one with normalization factor and another without it) for predicting friendship relation between users. Each measure in the numerical features is a statistic over the set common communities' membership of u and v, expressed as a function of the rule u → v.

1. The number of common interests: | Itemsets(u) ∩ Itemsets(v) |
2. Support (u → v) = Support (v → u) = $\mathrm{P}(\mathrm{u}, \mathrm{v})$
3. Confidence (u → v) = $\mathrm{P}(\mathrm{v}|\mathrm{u})$
4. Confidence (v → u) = $\mathrm{P}(\mathrm{u}|\mathrm{v})$
5. Lift (u → v) = $\frac{\mathrm{P}(\mathrm{v}|\mathrm{u})}{\mathrm{P}(\mathrm{v})}$
6. Conviction (u → v) = $\frac{\mathrm{P}(\mathrm{u})\mathrm{P}(\mathrm{v})}{\mathrm{P}(\mathrm{u},\neg\mathrm{v})}$
7. Match (u → v) = $\frac{\mathrm{P}(\mathrm{u},\mathrm{v})-\mathrm{P}(\mathrm{u})*\mathrm{P}(\mathrm{v})}{\mathrm{P}(\mathrm{u})*(1-\mathrm{P}(\mathrm{u}))}$
8. Accuracy (u → v) = $\mathrm{P}(\mathrm{u}, \mathrm{v}) + \mathrm{P}(\neg\mathrm{u}, \neg\mathrm{v})$
9. Leverage (u → v) $= \mathrm{P}(\mathrm{v}|\mathrm{u}) - \mathrm{P}(\mathrm{u})\mathrm{P}(\mathrm{v})$

First I use two measures with graph features (numbers 1 and 5) to illustrate the improvement of numerical features. For evaluation, I select inducers from different classifier models including and selecting all numerical features and common communities alone as one prediction feature, as shown in Table 5-10

Inducer	Features selection	
	Common Communities	All IM
J48	Yes	Yes
Random Forest	Yes	Yes
OneR	Yes	Yes

Yes: Available , IM : Interestingness measures

Table 5-10 Features Selection for common communities' measures

Next experiment, I use all numerical features without graph features to compare the normalized and unnormalized measures. For the experimental dataset, I use a dataset that consists of approximately 6000 user pairs where 50% are positive.

In another experiment with different setting I use training and test data set consisting of ~6000 user pairs each. Training data set consists of about 50% friend pairs and 50% non-friend pairs while the test data consists of randomly selected user pairs to preserve the original distribution of positive-negative instances in *LiveJournal*

(5991: 9 which is about less than 1% actual friends). I use the nine numerical features from the common community and other selected numerical features.

5.3.4 Ontology Based Refinement of User Interests

The next experiment addresses the prediction of friendships in *LiveJournal* using the association rule interestingness measures (as numerical features) for users' common interests with the use of the ontology. In related work, Bahirwani *et al.* (2008) have implemented a hybrid clustering algorithm (HAD) to automatically build a concept hierarchy of interests. As mentioned earlier, the accuracy of predicting friendship links in a social network, for instance, in absence of graph features is very low (Hsu, King, Paradesi, Weninger, & Pydimarri, 2006). In our paper (Bahirwani, Aljandal, Hsu, & Caragea, 2008), we explore how ontologies can be used to improve this performance with the use of a small dataset.

The first data set that has two version of measures (normalized and unnormalized), we use the count of common interests, plus eight AR interestingness measures over common interests, as numerical friendship prediction features. Each measure is a statistic over the set common interests of u and v, expressed as a function of the rule u → v.

1. The number of common interests: | Itemsets(u) ∩ Itemsets(v) |
2. Support (u → v) = Support (v → u) = $\mathrm{P(u,v)}$
3. Confidence (u → v) = $\mathrm{P(v|u)}$
4. Confidence (v → u) = $\mathrm{P(u|v)}$
5. Lift (u → v) = $\frac{\mathrm{P(v|u)}}{\mathrm{P(v)}}$
6. Conviction (u → v) = $\frac{\mathrm{P(u)P(v)}}{\mathrm{P(u,\neg v)}}$
7. Match (u → v) = $\frac{\mathrm{P(u,v)-P(u)*P(v)}}{\mathrm{P(u)*(1-P(u))}}$
8. Accuracy (u → v) = $\mathrm{P(u,v)} + \mathrm{P(\neg u,\neg v)}$
9. Leverage (u → v) = $\mathrm{P(u,v)} - \mathrm{P(u)P(v)}$

The second dataset is basket of users that share specific interests. The basket will contain the following:

1. Interest-Id: the name of interest (Art, Games, Movies, etc.)
2. User-Id: List of all users that have interest.

For evaluation we apply inducers from different classifier models include Random Forest, SVM, Logistic, Random-Tree, ADTree and Decision table. In this experiment, the training and test data set consists of 1000 user pairs. The training data set consists of about 50% friend pairs and 50% non-friend pairs while the test data consists of randomly selected user pairs to preserve the original distribution of positive-negative instances in *LiveJournal* which is about 1% positive and 99% negative.

5.3.5 Predicting Protein-Protein Interaction Using Parent- Child Relation

In this experiment, I use a data set containing known protein–protein interactions (PPI) used in Ben-Hur and Noble (2005). The goals of my experiments are to predict PPI using normalized and unnormalized numerical feature from parent-child relationships. My experiment design is modeled after that used by Taskar *et al.* (2003) in the social network domain.

The data set of PPI consists of more than 10,000 positive protein pairs and around 10,000 known negative protein pairs. For preparing the datasets both positive and negative sets are split into two parts for testing/training and all links (positive pairs) that connect the two sets are removed (they are independent). The first step in training is to build a graph based on positive PPI and to represent the parent-child relations in the dataset like a market basket for similar analysis. The next step is to use 10,000 protein pairs – made of 50% positive and 50% negative examples – as the training set, and construct numerical features from the co-occurrence of proteins in the training parent-child dataset. For testing, the positive proteins pairs will be used after hiding number of existing links (positive example of PPI) (50%, 75%) and the rest are used to build an incomplete graph of positive pairs. Next, we construct four 5,000 protein-pair-datasets with 1%, 2%, 5%, 10% positive and 99%, 98%, 95%, 90% negative examples respectively as the test set. I follow this technique because the real ratio of negative examples to positive examples is currently unknown. Finally, from only the known part of the graph we construct numerical features based on co-occurrence of proteins in the testing parent-child dataset. Therefore, the module will predict unknown links (from the hidden part) using numerical features of the known part.

This experiment uses only the connection structure of the positive PPI. I evaluated the normalized and unnormlized numerical features using two classifier models and inductive learning algorithms: the k-nearest neighbor approach IB1, and the rule based approach OneR (Aljandal, Hsu, & Xia, 2009).

5.3.6 Divided-by-N Vs Itemset Size-Sensitive Normalization Factor

Divided-by-N is a well-known method in graph theory related to a water flow concept where the amount of flow into a node equals the amount of flow out of it. In the *LiveJournal* dataset, I consider the set of users that share a property (such as a community) as a node that connected to set of other nodes (users), as shown in Figure 5-10. Therefore, based on the divided-by-N concept, the importance of this property comes from the number of users that share this property. Because the item size-sensitive joint probability is associated with the size of each itemset, we can see that the concept of divided-by-N is the only concept related to the itemset size. So I consider divided-by-N a normalization factor and compare it with my normalization factor.

In this experiment I am going to produce two classification files with four association rule interestingness measures:

1. Common Communities
2. Lift of Common Communities
3. Conviction of Common Communities
4. Leverage Common Communities

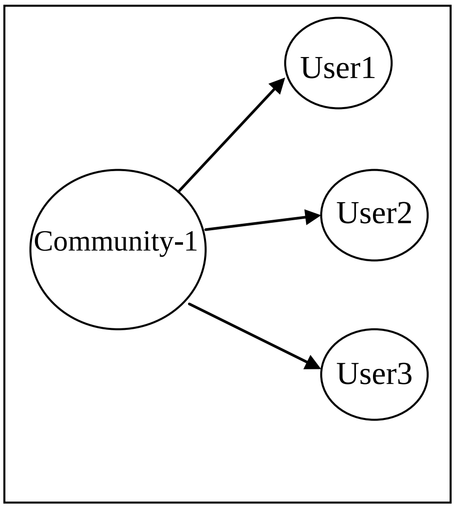


Figure 5-10 Community Graph

The next step is to substitute the original joint probability on each measure with item-size-sensitive joint probability (include the new normalization factor) and joint probability with divided-by-N normalization factor

We can define a joint probability with divided-by-N as follows:

Let $L \equiv \{x_1, x_2 \ldots x_k\}$ be the set of items. Let D be a set of transactions ($|D| = N$), where each transaction T is a set of items such that $T \subseteq L$. Then:

$$\hat{p}(x_1, x_2, \ldots, x_q) \triangleq \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|T_i|} \tag{5}$$

The dataset in this experiment consist of ~6000 user pairs among which about 50% are friend. The number of communities is ~ 44,000 communities. I compare the two methods by using three classifier models, Random Forest, OneR and IB1, using 10-folds cross validation.

5.3.7 Numerical Features for Interest Prediction

In this experiment I show how numerical features can be used as prediction features. By using the same dataset that I used in Section 5.3.4, we constructed two types of interest-based features:

Nominal: measured for grouped relationships for a candidate pair of entities by name (e.g., Are u and v both interested in topics under the category of mobile computing?).

Numerical: interestingness measures that are computed across these grouped relationships (e.g., how many interests that u is interested does v share, and how rare are these interests?).

All features in these two categories are examples of pair-dependent co-membership features and can be computed using the ontology.

We used the 1000-user data set, which includes about 22000 unique interests that are shared by at least two users. (Interests held by only one user are of no interest for link prediction, so singleton itemsets are pruned as is often done in frequent itemset mining.) Hierarchical Agglomerative and Divisive (HAD) clustering, a hybrid bottom-up linkage-based and divisive (partitional) algorithm, was used to generate the hierarchy. The output, consisting of 19 clusters, is summarized in Figure 5-11; note that the level of abstraction can be manually set, as I do in my experiments. I refer the interested reader to (Bahirwani, Aljandal, Hsu, & Caragea, 2008) for additional details of the clustering algorithm and documentation on the data sources consulted.

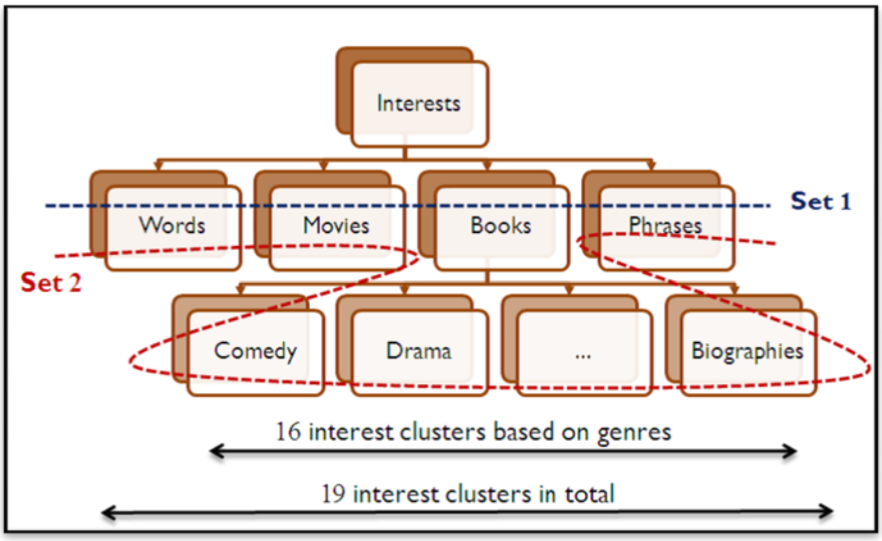


Figure 5-11 Example of clusters found using Hierarchical Agglomerative and Divisive (HAD) algorithm.

From (HAD) clustering we get 19 clusters, resulting a 19 + 19 = 38 nominal features for every candidate pair (*u*, *v*). To these we add the original 7 graph features and the 9 numerical features. We use the integrated, ontology-enhanced data set to predict whether an individual user *u* lists a member of one of the 19 abstract interest categories, given the fraction of their friends in the network that also list that category.

We evaluated the nominal and numerical features using five classifier models and inductive learning algorithms: support vector machines (SVM), Logistic Classification, Random Forests, J48, and OneR.

CHAPTER 6 - **RESULTS**

This chapter documents the evaluation of the new normalization factor impact that can improve the sensitivity of the interestingness measure regarding the itemset size. First, I use the information gain framework that I designed in Chapter 5 to demonstrate how interestingness measures with new normalization factor provide enhanced ranking method compared with the original interestingness measure based on attribute surprisingness, even though the market basket domain is one of these that has exogenous variables that can affect data properties. In the second experiment, we go through another framework based on classification methods that I designed in the second section of Chapter 5. This framework uses different datasets to illustrate the improvement of accuracy measures when we use the new normalization factor with association rule measures as predicting features.

6.1 Experimental Results for Information Gain and Rules Selection

In this experiment I use small data sets. Table 6-1 shows the size of each dataset, the number of distinct items and the total number of transactions. The association rules were generated by using Magnum opus with limitation of support for each of the datasets, as show in column "Limitation" in Table 6-1.

#	Topic	Source	Construction Method	Items	Transaction	Number of AR	Limitation
1	Market Basket	Magnum opus	Original	16	1000	6151	Supp >0.001
2	Artificial	Magnum opus	Hand-modified	13	10	28	Supp ≥ 0.2
3	Artificial	Magnum opus	Hand-modified	10	10	44	Supp ≥ 0.2
4	Market Basket	Brijs's Dissertation	Sample	303	1000	6505	Supp >0.02

Table 6-1 Datasets information and limitation

After generating the rules with no redundancy, I use the experiment framework that I designed in Chapter 5 to evaluate the result.

Dataset #	Compare by Candidates	Inversions (order)		Different
		Lift (without R)	Lift (with R)	
First dataset	Top n (n=14)	76	79	-4%
	All	163	132	19%
Second dataset	Top n (n=14)	158	140	11%
	All	513	407	21%
Third dataset	Top n (n=14)	158	140	11%
	All	513	407	21%
Forth dataset	Top n (n=2000)	5.367E6	4.916E6	8.%
	All	1.77E7	1.299E7	27%

Table 6-2 Experiment result (information gain framework)

Table 6-2 shows the result of the inversion comparison between the two versions of Lift measure (one with the normalization factor “with R” and another without it; “without R”). For both parts of the candidates, the “top n” and “all rules” the Lift measure (with R) record better result compared with the original Lift measure except in the first dataset in “top n” option, which is insignificantly different. This happened because the number of inversion is very low which indicate that the first N rules contained items with strong relationship that normalization factor can not improve the ranking. This gives a clear indication that the list that has fewer inversions assigns a high score to the more interesting rules based on attribute surprisingness. Therefore, using the new normalization factor with the market basket domain can be useful even though the domain has some exogenous variables that explain why the improvement is not that significant. The improvement can be seen when the normalization factor adjust the Lift measure to make it more sensitive to its own data behavior and produce more interesting (surprising) rules.

6.1.1 Rules clustering

From the previous results, I chose the result of the third dataset in Table 6-1 and apply a clustering method to show ranking distribution. Visualizing the clustering result in the chart gives a clear explanation about ranking distribution between the two local lists and the global one. The resulting local list of rules (locals) have been ranked based on two measures, the original lift “without-R” and the lift with the normalization factor “With-R.” the global list has been ranked based on the attribute surprisingness (Information Gain) and is subsequently labeled “Global.” I use K-means clustering discussed in Chapter 3 to group the rules in each list into five different groups based on their ranks.

The result of clustering the rules into five clusters is shown in the Table 6-3. The center of each cluster represents the average rules ranking.

Lists / Clusters	Global	With-R	Without-R
Cluster 0	5.36	3.59	1.00
Cluster 1	13.86	9.45	2.20
Cluster 2	20.50	12.80	4.77
Cluster 3	25.83	15.50	6.40
Cluster 4	28.75	17.00	8.00

Table 6-3 Clustering result

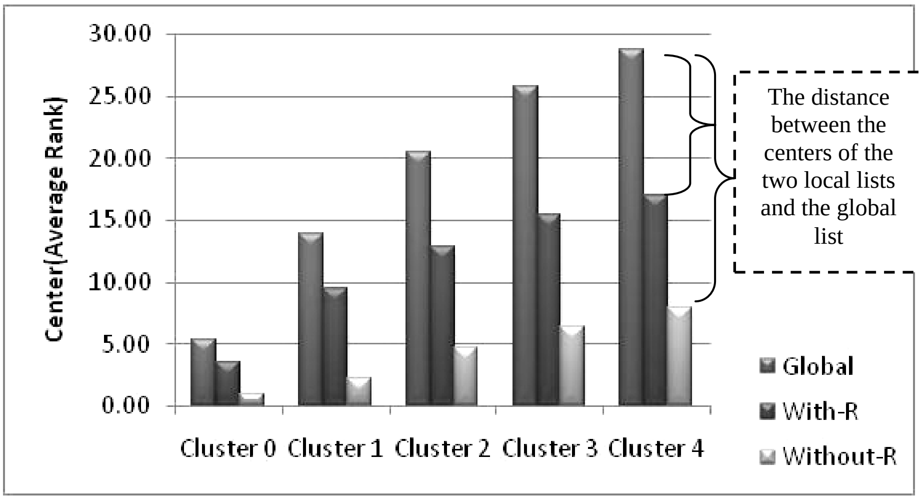


Figure 6-1 Clustering result graph

In Figure 6-1, we can see the distance between clusters’ centers of the global list and the two local lists “with-R” and “without-R” that shows the closeness between the centers of the global list in all clusters and the centers of the local list “with-R” compared with the distance between the centers of the global list and the centers of local list “without-R”. From this result we can indicate that “with-R” list gives better distribution ranking than “without-R” list.

The results from these experiments show that the new normalization factor improves the distribution of the Lift measure ranking which makes it a more sensitive measure. However, using the new normalization factor with other measures may lead to different result based on how these measures handle the joint probability.

6.2 Experiment of Predicting Friendship Relation Using User Interests

This section presents the result of classification using interestingness measures with and without Itemset size normalization factor on *Livejournal* data set.

Data set	Inducer	Accuracy	Precision	Recall	F-measure	Features - selection
Unnormalized	IBl	74.2%	0.418	0.438	0.428	One attribute (Lift)
Normalized	IBl	77.56%	0.491	0.532	0.510	One attribute (Lift)
Unnormalized	OneR	78.65%	0.633	0.071	0.128	One attribute (common Interests)
Normalized	OneR	79.44%	0.597	0.203	0.303	One attribute (common Interests)
Unnormalized	Random Forest	78.73%	0.659	0.070	0.126	One attribute (common Interests)
Normalized	Random Forest	80.41%	0.599	0.333	0.428	One attribute (common Interests)
Unnormalized	IBl	68.59%	0.275	0.260	0.267	One attribute (common Interests)
Normalized	IBl	76.02%	0.450	0.401	0.424	One attribute (common Interests)
Unnormalized	Random Forest	79.15%	0.533	0.428	0.475	IM Features
Normalized	Random Forest	80.55%	0.575	0.448	0.504	IM Features
Unnormalized	OneR	77.67%	0.472	0.124	0.196	IM Features
Normalized	OneR	79.42%	0.594	0.206	0.306	IM Features
Unnormalized	Random Forest	96.99%	0.936	0.927	0.931	All- Features
Normalized	Random Forest	97.12%	0.939	0.929	0.934	All- Features

IM : Interestingness Measures

Table 6-4 Experiment result for predicting friends using user interests

Table 6-4 contains classification measures (Chapter 3) and results from applying three different inducers: OneR, Random Forest and IB1 (Chapter 3). All accuracy measures were collected over 10-fold cross-validated runs. On all inducers that I am using the normalization factor boost the accuracy measures which is a result of improving the interestingness measures sensitivity as friendship prediction features. The improvement occurs in all accuracy measures in general with different ranges based on inducers method. For example, the best accuracy improvement in this experiment was achieved when I used the IB1 inducer with one attribute feature (common interests). It improves by 10.83% (from 68.59 to 76.02). In another experiment, the F-measure, which is a combination of Recall and Precision, improved by 239% (from 0.126 to 0.428) when I used the Random Forest inducer with one attribute feature (common interests).

Figure 6-2 visualizes accuracy measure in all inducers. This plot shows the performance of accuracy percentage for both normalized and unnormalized methods. The results of other measures (Precision, Recall and F-measure) are graphed in Figure 6-3 , Figure 6-4, and Figure 6-5 consecutively.

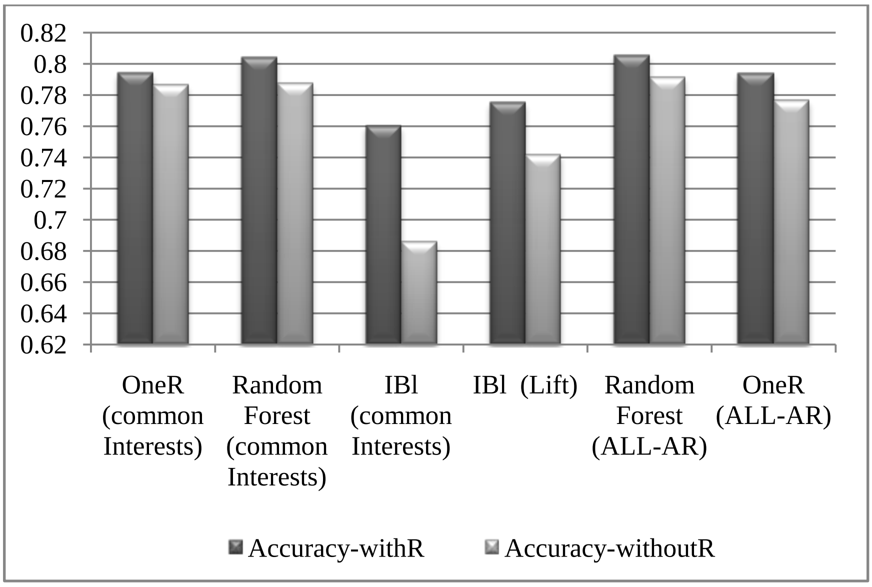


Figure 6-2 Accuracy graph for classification result

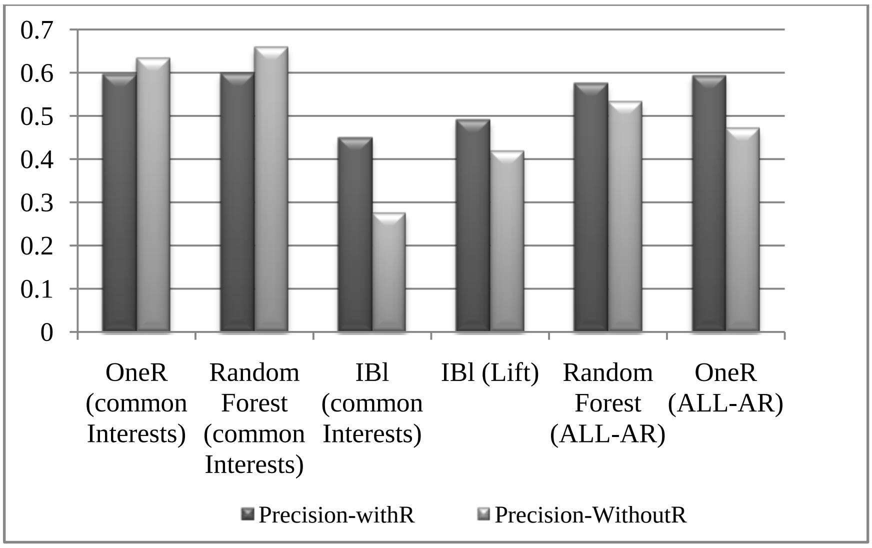


Figure 6-3 Precision graph for classification result

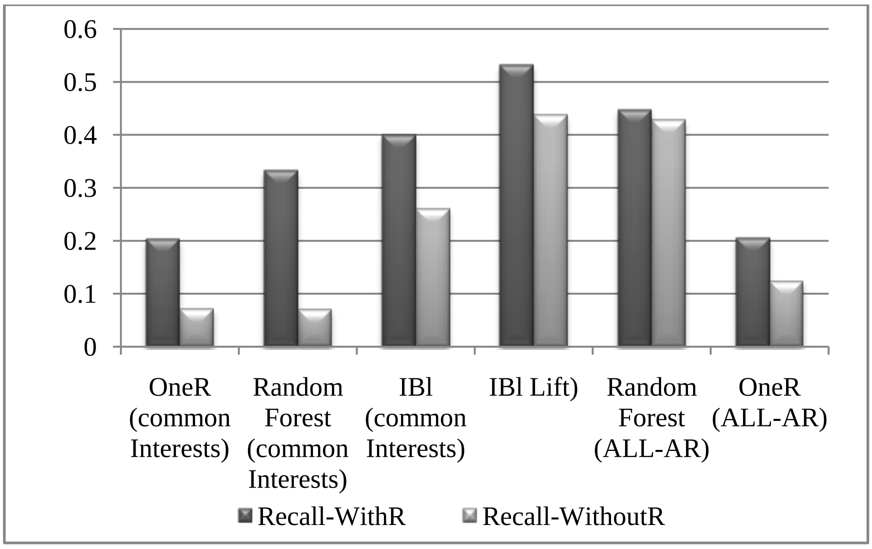


Figure 6-4 Recall graph for classification result

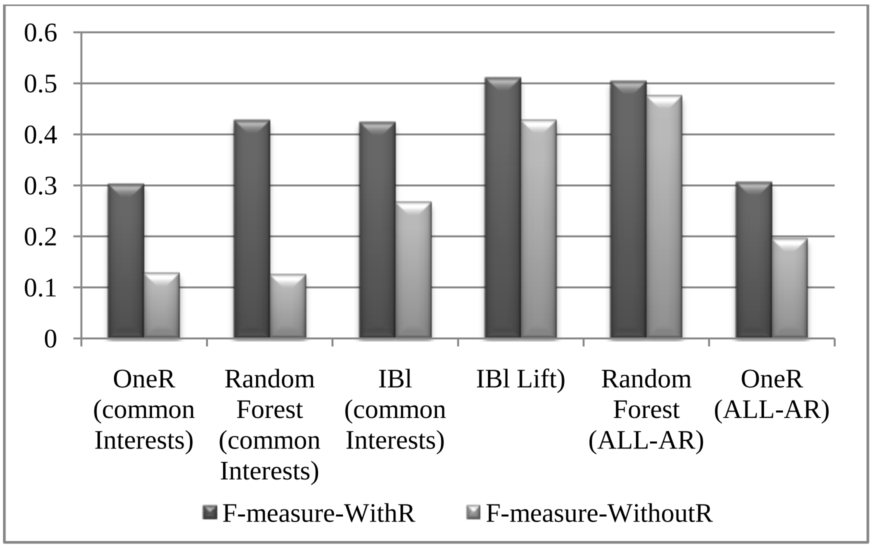


Figure 6-5 F-measure graph for classification result

From Figure 6-3 we can observe that in two out of six cases, “unnormalized” achieves a better precision result both of which are insignificant losses. This result can be explained when we go through more detail about each experiment.

In the next section, I present more detailed results of the OneR with One attribute and the T-test to show the significance. There are more detailed results in Appendix-A.

6.2.1 Detailed results of OneR obtained with common interests

In the classification process I collect accuracy measures over 10-fold cross-validated runs, so going through each of these folds will give some information about the final result. The first experiment is OneR with One attribute (common interest) as shown in Table A-1. In addition, I am going to visualize the detail results in a graph and use a T-test for some of the experiments results.

Fold	precision-WithR	precision WithoutR	recall WithR	recall WithoutR	F-measure WithR	F-measure WithoutR
1	0.563380	0.722222	0.181818	0.059091	0.274914	0.109244
2	0.637363	0.653846	0.263636	0.077273	0.372990	0.138211
3	0.605263	0.724138	0.209091	0.095455	0.310811	0.168675
4	0.614458	0.526316	0.231818	0.045455	0.336634	0.083682
5	0.616438	0.593750	0.204545	0.086364	0.307167	0.150794
6	0.597015	0.529412	0.181818	0.081818	0.278746	0.141732
7	0.535211	0.520000	0.172727	0.059091	0.261168	0.106122
8	0.609375	0.650000	0.177273	0.059091	0.274648	0.108333
9	0.576923	0.761905	0.204545	0.072727	0.302013	0.132780
10	0.600000	0.708333	0.203620	0.076923	0.304054	0.138776

Table 6-5 OneR with one attribute (common interests) - for each fold

Figure 6-6 graphs the precision measure for each fold in the OneR (common interests) experiment. In some folds the new method (withR) has some improvement. But overall the original method (withoutR) records a better result.

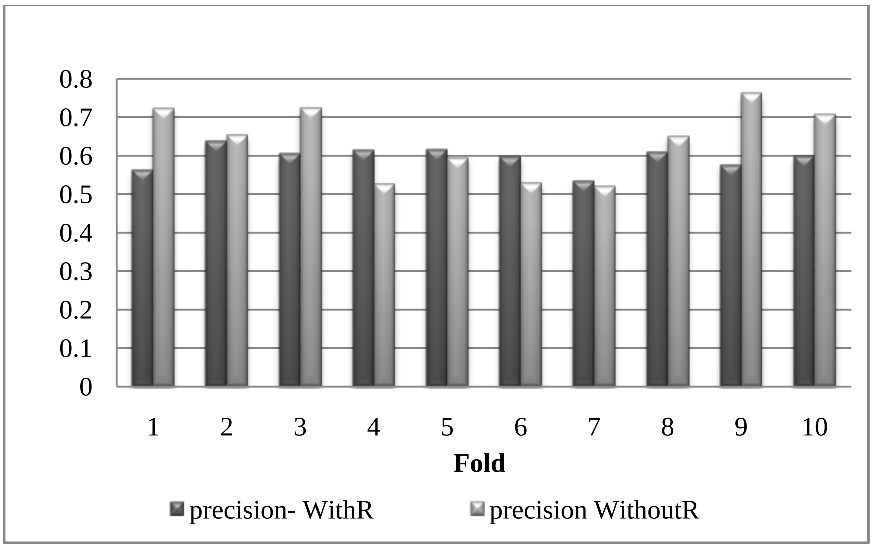


Figure 6-6 Precision of OneR (common interests)- all folds

This difference only appears in precision measure which is a ration of true positive to the total of true positive and false positive. Thus, when we look at Table 6-6 and Table 6-7 which show the prediction rates for both methods (normalized and Unnormalized), we can observe that on average the new method (normalized) has much better rate for predicting true positive elements than the original method (withoutR). The problem is in the false positive where the new method has higher rate than the original one. This situation can be acceptable in some applications like friendship recommendation systems but not in some medical systems. On the other hand, the new method has lower false negative rate (0.797 on average) then the original one (0.929 on average).

Fold	TP_rate	TP	FP_rate	FP	TN_rate	TN	FN_rate	FN
1	0.182	40	0.040	31	0.960	749	0.818	180
2	0.264	58	0.042	33	0.958	747	0.736	162
3	0.209	46	0.038	30	0.962	750	0.791	174
4	0.232	51	0.041	32	0.959	748	0.768	169
5	0.205	45	0.036	28	0.964	752	0.795	175
6	0.182	40	0.035	27	0.965	753	0.818	180
7	0.173	38	0.042	33	0.958	747	0.827	182
8	0.177	39	0.032	25	0.968	755	0.823	181
9	0.205	45	0.042	33	0.958	747	0.795	175
10	0.204	45	0.039	30	0.961	749	0.796	176
Average	0.203	44.7	0.039	30.2	0.961	749.7	0.797	175.4

Table 6-6 Precision of OneR (common interests)- all folds -"WithR "

Fold	TP_rate	TP	FP_rate	FP	TN_rate	TN	FN_rate	FN
1	0.059	13	0.006	5	0.994	775	0.941	207
2	0.077	17	0.012	9	0.988	771	0.923	203
3	0.095	21	0.010	8	0.990	772	0.905	199
4	0.045	10	0.012	9	0.988	771	0.955	210
5	0.086	19	0.017	13	0.983	767	0.914	201
6	0.082	18	0.021	16	0.979	764	0.918	202
7	0.059	13	0.015	12	0.985	768	0.941	207
8	0.059	13	0.009	7	0.991	773	0.941	207
9	0.073	16	0.006	5	0.994	775	0.927	204
10	0.077	17	0.009	7	0.991	772	0.92308	204
Average	0.071	15.7	0.012	9.10	0.988	770.8	0.929	204.4

Table 6-7 Precision of OneR (common interests)- all folds -"WithoutR "

Even though, the original method has slightly better results in precision measure still the new method records a much better result in Recall measures as shown in Figure 6-7. In addition, F-measure results shown in Figure 6-8, combine precision and recall, show that the new method records significantly improved result compared with the original one.

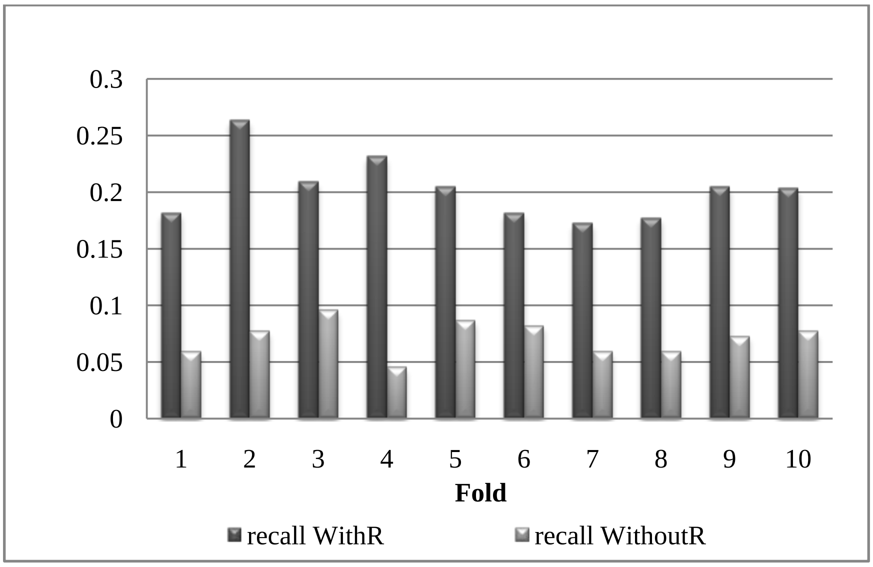


Figure 6-7 Recall of OneR (common interests)- all folds

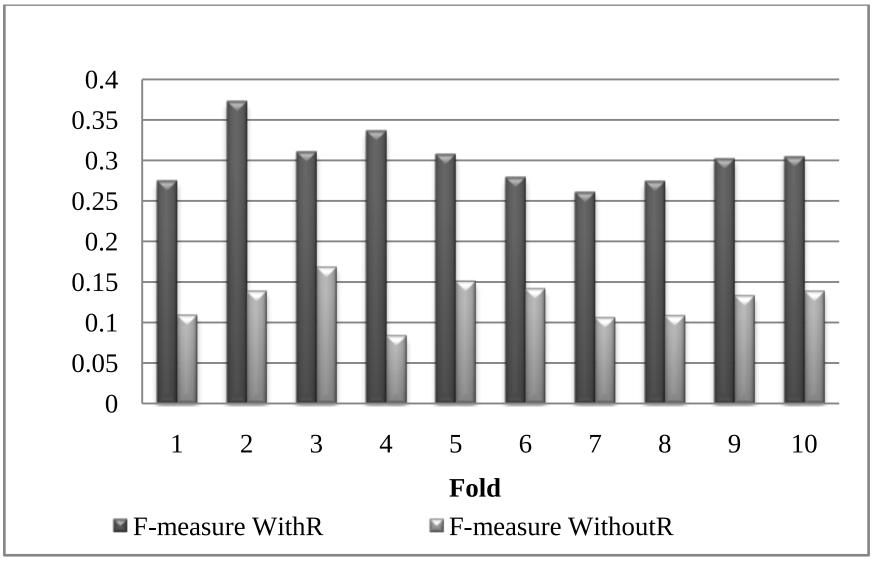


Figure 6-8 F-measure of OneR (common interests)- all folds

I use the T-test to assess some of my results specifically to test the statistical hypothesis that the mean precision, recall, or F-measure of one experiment is higher than that of on other. This analysis is appropriate in my experiment to compare the means of the two fold's results of normalized and unnormalized methods for each measure. All of the T-test resulted with 95% confident (with alpha level set at 0.05). Table 6-8 shows statistical information with the T-test results for the precision measure and we can see the value is negative indicating that the mean of first list "Normalized" is less than mean of the second one "Unnormalized".

	precision WithR	*precision WithoutR*
Mean	0.5955	0.6390
Variance	0.0009	0.0084
Observations	10	10
df	9	
t Stat	-1.4444	
P(T<=t) one-tail	9.13E-02	
t Critical one-tail	1.8331	

Table 6-8 T-test of precision (OneR - common interests)

Table 6-9 and Table 6-10 show that there is a significant difference between the two results. For example the P-value in the recall is "1.10E-07" which is much less than the 0.05.

	recall WithR	*recall WithoutR*
Mean	0.2031	0.0713
Variance	0.0008	0.0002
Observations	10	10
df	9	
t Stat	13.8848	
P(T<=t) one-tail	1.10E-07	
t Critical one-tail	1.8331	

Table 6-9 T-test of Recall (OneR - common interests)

	F-measure WithR	*F-measure WithoutR*
Mean	0.3023	0.1278
Variance	0.0011	0.0006
Observations	10	10
df	9	
t Stat	14.4041	
P(T<=t) one-tail	8.02E-08	
t Critical one-tail	1.8331	

Table 6-10 T-test of F-measure (OneR - common interests)

6.3 Experiment of Predicting Friendship using Common Community Membership

The second experiment shows that by selecting proper interestingness measures with available properties will improve the link prediction. Using communities' membership information with selected features (graph features and two numerical features) with and without the Itemset size normalization factor will show the superiority of my method for friendship prediction. Table 6-11 shows the J48 classification results for ~6000 user pairs where 50% are friends and using 10-fold cross validation.

Feature	Accuracy (%)	Precision	Recall	F-measure
GF	92.977	0.932	0.903	0.918
GF+AR	93.780	0.936	0.919	0.927
GF + N-AR	94.081	0.941	0.921	0.931

Table 6-11 Result of accuracy, precision, recall, F-measure for J48 (10-fold CV).

(GF: Graph Feature, AR: Interestingness Measures, N-AR: Normalized Interestingness Measures)

The classification results show how link prediction improves when I use selected interestingness measures of user communities as new features. In addition, the improvement will be augmented if we use normalized interestingness measures.

In the classification process I collected accuracy measures for each 10-fold cross-validations run to illustrate the significance of normalized measures. In the detailed results of J48 with different selection of features (GF and GF+N-AR), a significant improvement was observed across all measures (Accuracy, Precision, Recall and F-measure), especially when I use the normalized interestingness measures with graph features as shown in Figure 6-9 through Figure 6-12.

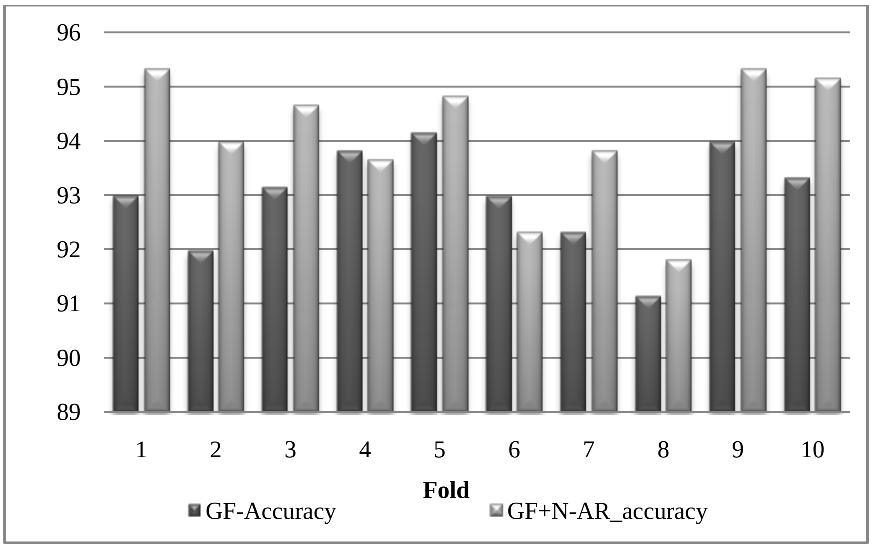


Figure 6-9 Percent accuracy for J48 (for each fold)

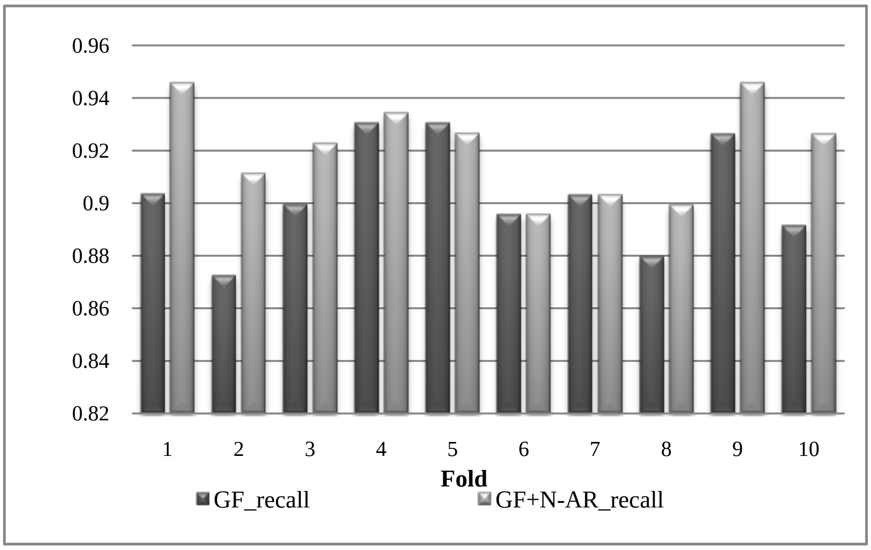


Figure 6-10 Percent of recall for J48 (for each fold)

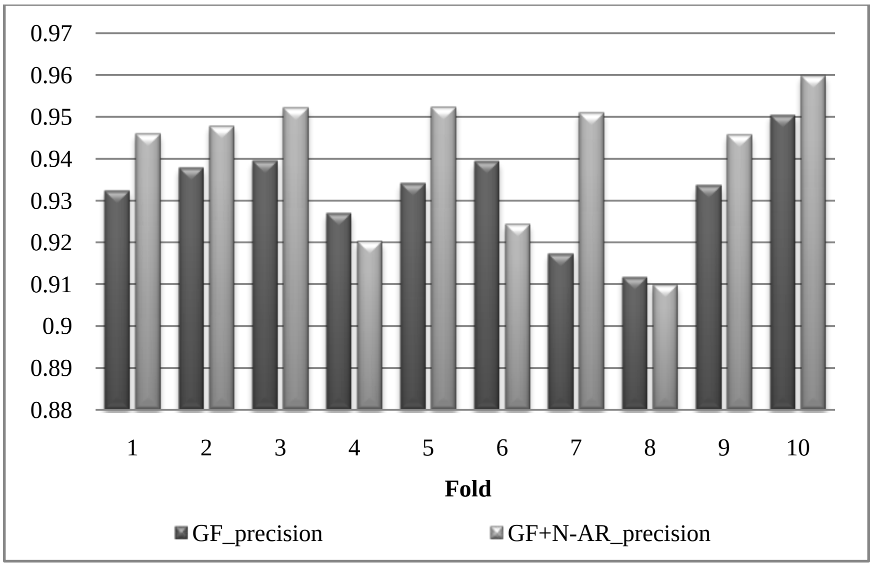


Figure 6-11 Percent of precision for J48 (for each fold)

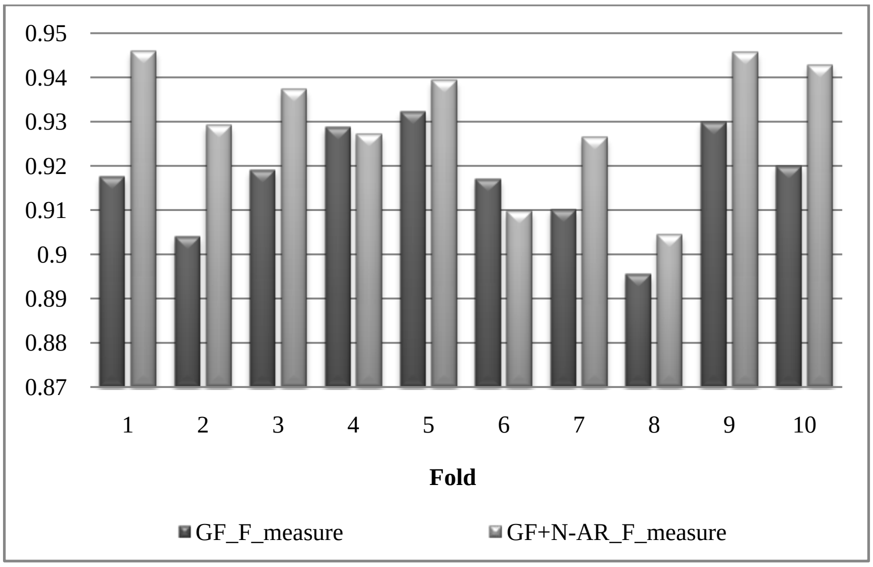


Figure 6-12 Percent of F-measure for J48 (for each fold)

We again use a *T-test* to evaluate the significance of the results at the 95% level of confidence (with alpha level 0.05). Table 6-12 through Table 6-15 show the test results for accuracy, precision, recall and F-measure.

	GF+N-AR accuracy	GF-Accuracy
Mean	94.08104	92.9777
Variance	1.519662	0.89485
Observations	10	10
t Stat	3.61336	
P(T<=t) one-tail	0.00281	
t Critical one-tail	1.83311	

Table 6-12 T-test for accuracy measure on J48 – (all folds)

	GF+N-AR precision	GF precision
Mean	0.94091	0.93228
Variance	0.00027	0.00012
Observations	10	10
t Stat	1.99724	
P(T<=t) one-tail	0.03844	
t Critical one-tail	1.83311	

Table 6-13 T-test for precision measure on J48 – (all folds)

	GF+N-AR recall	GF recall
Mean	0.921109	0.90332
Variance	0.00033	0.00041
Observations	10	10
t Stat	3.269366	
P(T<=t) one-tail	0.004846	
t Critical one-tail	1.83311	

Table 6-14 T-test for recall measure on J48 – (all folds)

	GF+N-AR F-measure	GF F-measure
Mean	0.93081	0.91743
Variance	0.00020	0.000136
Observations	10	10
t Stat	3.67845	
P(T<=t) one-tail	0.00254	
t Critical one-tail	1.83311	

Table 6-15 T-test for F-measure on J48 – (all folds)

For all measures, the results of the T-test reflect a significant improvement attained by using normalized association rules measures with graph features.

In the next experiment with a different setting, I use training and test data set consisting of ~6000 user pairs. I use the nine numerical features from the common community.

Table 6-16 shows the classification measures from using the nine association measures as prediction features based on users' common communities. The normalized measures either improve the prediction measures or stay similar to the unnormalized measures.

Table 6-17 shows the classification measures for normalized and unnormalized support and lift of common communities.

Inducer	Method	Accuracy (%)	Precision	Recall	F-Measure	AUC
J48	unnormalized	96.65	0.015	0.029	0.029	0.65
	normalized	98.23	0.023	0.333	0.054	0.66
Random Forest	unnormalized	76.31	0.002	0.333	0.004	0.601
	normalized	77.37	0.002	0.333	0.004	0.616
OneR	unnormalized	93.93	0.008	0.333	0.016	0.637
	normalized	94.23	0.009	0.333	0.017	0.638

Table 6-16 Classification result for nine numerical features of common communities

Inducer	Method	Accuracy (%)	Precision	Recall	F-Measure	AUC
J48	unnormalized	96.53	0.015	0.333	0.028	0.65
	normalized	98.15	0.028	0.333	0.051	0.658

Table 6-17 Classification result for Support &Lift of common communities

The results of the last experiment, shown in Table 6-17 illustrate the positive effects of the itemset size normalization factor in association measures (support and lift). The best improvement is shown in F-Measure, which increased from 0.028 to 0.051 which is more than on 80% improvement. However, the low number of friends in the test set, which comes from normal distribution, causes the closeness on the final result between some measures. I have done a variety of feature selections but overall, the normalized measures either improve the result or does not harm the final result.

6.4 Experiment of Ontology-Based Refinement of User Interests.

The third experiment addresses the prediction of friendships in *LiveJournal* using association rule interestingness measures (as numerical features) for users' common interests with the use of the ontology. The results of six different inducers are listed in Table 6-18 and Table 6-19.

The itemset size normalized factor works well with a large dataset because the dataset will express the skewness of the data. Even though in this experiment I use only 1000 user pairs, the normalized measures improve the classification measurers. These normalized measures take into account the popularity that particular interests hold in common, where the most popular interests (held by a significant proportion of users) being slightly less revealing than rarer interests. Furthermore, I investigate how the ontology improves the classification measurers especially when I use normalized measures, which boost the measures sensitivity in regards to interest popularity. When computing the measures, I modify the interests of users by viewing the interests at the "best" level of abstraction as suggested in by Bahirwani, *et al.* (2008).

Inducer	Method	Accuracy (%)	Precision	Recall	F-Measure	AUC
Random Forest	unnormalized	67.5	0.015	0.556	0.030	0.688
	normalized	65.3	0.014	0.556	0.028	0.605
SVM	unnormalized	89.3	0.038	0.444	0.07	0.709
	normalized	92.3	0.053	0.444	0.094	0.711
Logistic	unnormalized	74.4	0.019	0.556	0.038	0.678
	normalized	85.5	0.034	0.556	0.065	0.68
Random-Tree	unnormalized	66.4	0.015	0.556	0.029	0.640
	normalized	65.9	0.015	0.556	0.028	0.637
ADTree	unnormalized	73.7	0.019	0.556	0.037	0.671
	normalized	78.8	0.023	0.556	0.045	0.694
Decision table	unnormalized	83.8	0.031	0.556	0.058	0.67
	normalized	82.3	0.028	0.556	0.053	0.689

Table 6-18 Classification result- without ontology

Inducer	Method	Accuracy (%)	Precision	Recall	F-Measure	AUC
Random Forest	unnormalized	67.6	0.018	0.857	0.036	0.773
	normalized	70	0.020	0.857	0.038	0.829
SVM	unnormalized	89.4	0.046	0.714	0.086	0.889
	normalized	89.9	0.048	0.714	0.090	0.893
Logistic	unnormalized	86.8	0.037	0.714	0.070	0.912
	normalized	89.7	0.056	0.857	0.104	0.894
Random-Tree	unnormalized	66.9	0.012	0.571	0.024	0.606
	normalized	67.5	0.015	0.714	0.03	0.689
ADTree	unnormalized	77.8	0.026	0.857	0.051	0.90
	normalized	82.7	0.034	0.857	0.065	0.925
Decision table	unnormalized	81.8	0.032	0.857	0.062	0.872
	normalized	86.8	0.037	0.714	0.07	0.873

Table 6-19 Classification result- with ontology

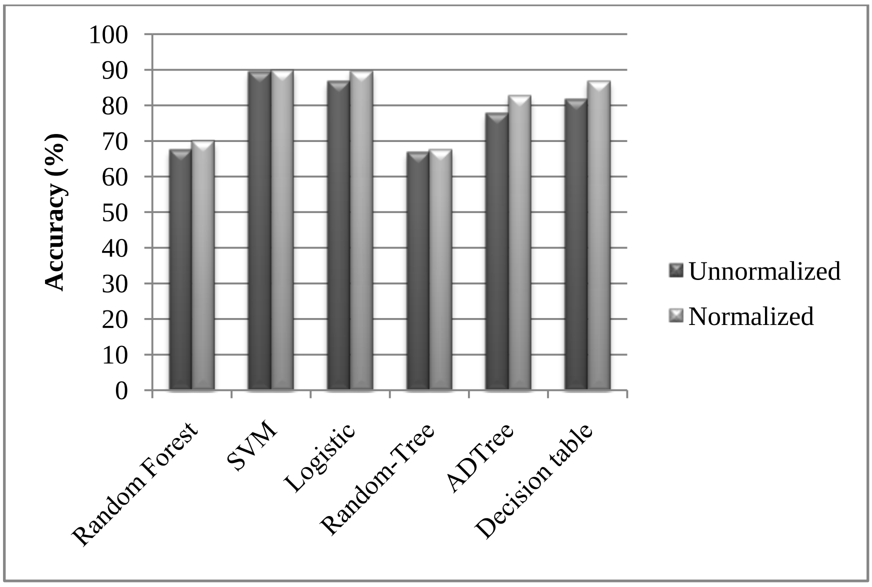


Figure 6-13 Result of Accuracy - with ontology

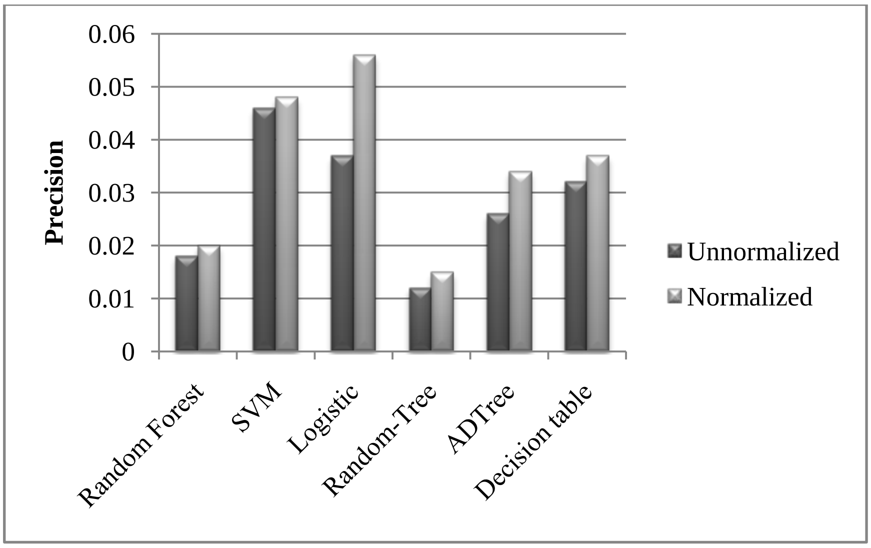


Figure 6-14 Result of Precision - with ontology

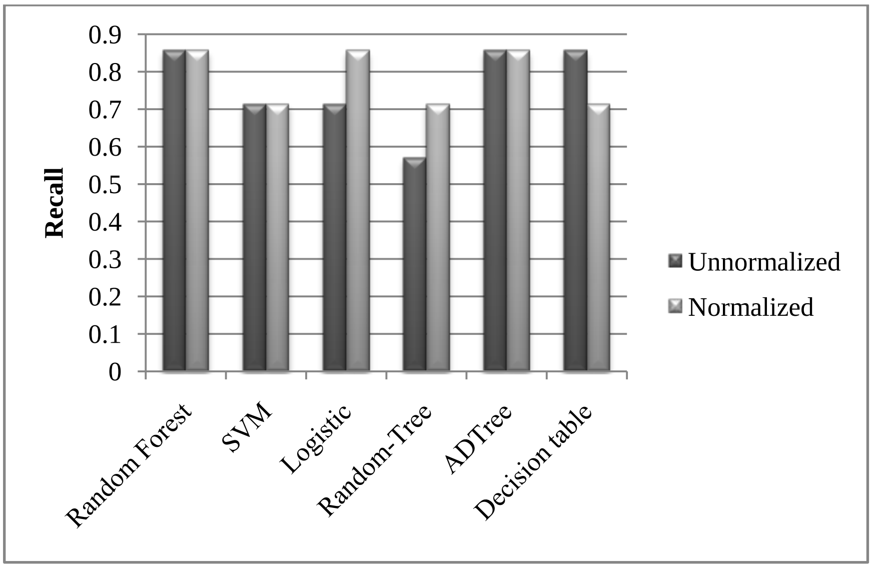


Figure 6-15 Result of Recall - with ontology

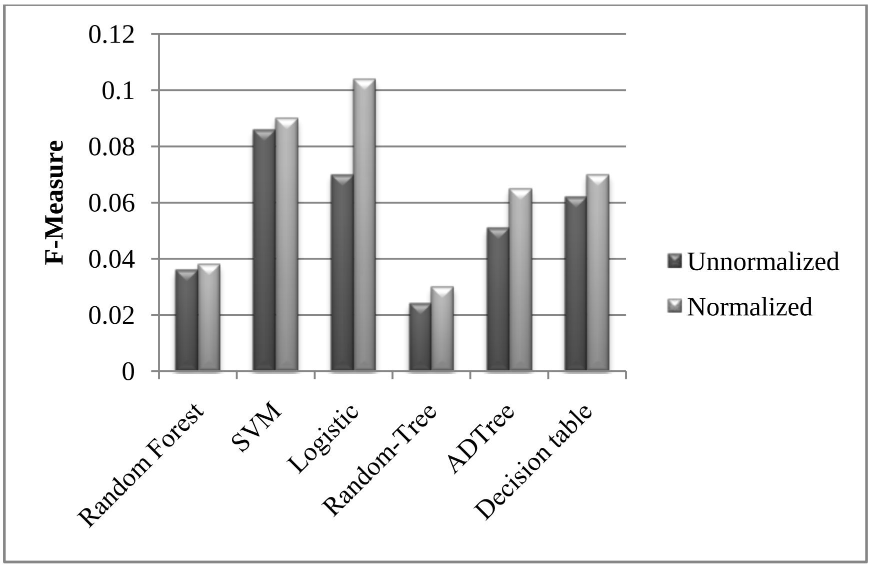


Figure 6-16 Result of F-Measure - with ontology

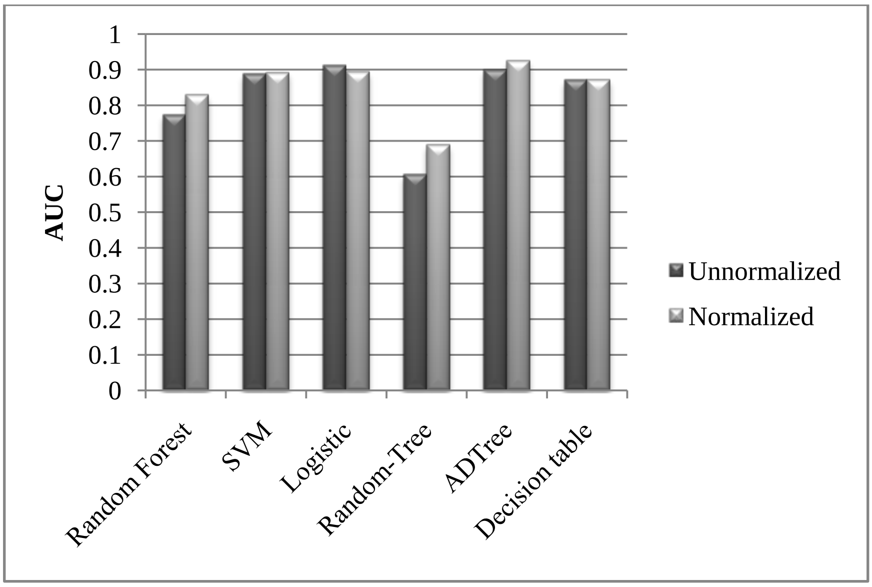


Figure 6-17 Result of AUC - with ontology

Table 6-18 and Table 6-19 summarize the expected improvements. For example, without an ontology, better results are observed for unnormalized measures using the Random Forest inducer (in most of the classification measures) but when I modify the data according to a concept hierarchy, the improvement using normalized measures consistently exceeds that achieved using unnormalized measures which also shown in Figure 6-13 through Figure 6-17.

6.5 Experiment of Predicting Protein–Protein Interaction

The performance expectation of the result shows how models learned from the unnormalized numerical features are able to predict PPI relationships, and how performance further improves using normalized measures. In this experiment, I present only the significant results for two of the numerical features: Accuracy and Leverage. The results shown in Table 6-20 illustrate the classification performance measures in terms of precision, recall, F-measure, and area under the ROC curve (ROC-AUC, henceforth "AUC") based on either Accuracy or Leverage features alone and 50% observed positive proteins pairs using the IB1 classification method. We see that the best AUC recorded was 0.854 in the dataset with 2% positive examples with Normalized accuracy.

When I use the second testing sets with only 25% observed PPI (75% hidden), the performance was reduced from the previous case with an AUC of 0.781. Complete results are shown in Table 6-21.

50%					
	Method	Precision	Recall	F-Measure	AUC
1%	U- Accuracy	0.304	0.280	0.292	0.637
	N- Accuracy	0.174	0.660	0.275	0.814
	Different	-42.76%	135.71%	-5.82%	27.79%
	U- Leverage	0.319	0.300	0.309	0.647
	N- Leverage	0.288	0.340	0.312	0.666
	Different	-9.72%	13.33%	0.97%	2.94%
2%	U- Accuracy	0.458	0.270	0.340	0.632
	N- Accuracy	0.320	0.740	0.447	0.854
	Different	-30.13%	174.07%	31.47%	35.13%
	U- Leverage	0.492	0.310	0.380	0.652
	N- Leverage	0.468	0.370	0.413	0.681
	Different	-4.88%	19.35%	8.68%	4.45%
5%	U- Accuracy	0.624	0.212	0.316	0.603
	N- Accuracy	0.505	0.640	0.564	0.804
	Different	-19.07%	201.89%	78.48%	33.33%
	U- Leverage	0.648	0.236	0.346	0.615
	N- Leverage	0.622	0.276	0.382	0.634
	Different	-4.01%	16.95%	10.40%	3.09%
10%	U- Accuracy	0.761	0.204	0.322	0.599
	N- Accuracy	0.667	0.630	0.648	0.799
	Different	-12.35%	208.82%	101.24%	33.39%
	U- Leverage	0.778	0.224	0.348	0.609
	N- Leverage	0.753	0.256	0.382	0.624
	Different	-3.21%	14.29%	9.77%	2.46%

Table 6-20 IB1-Classification measures for 50% hiding

(U- : Unnormalized, N- : Normalized)

Figure 6-18 shows the comparison between normalized and unnormalized accuracy features based on AUC from the 50% observed data. The superiority of normalized features came from their ability to capture the rarity of childhood and parenthood of positive proteins. In addition, with every decrease in precision result we get, there is high improvement in recall and F-measure, which is a similar effect to that observed in some of the social networks experiments.

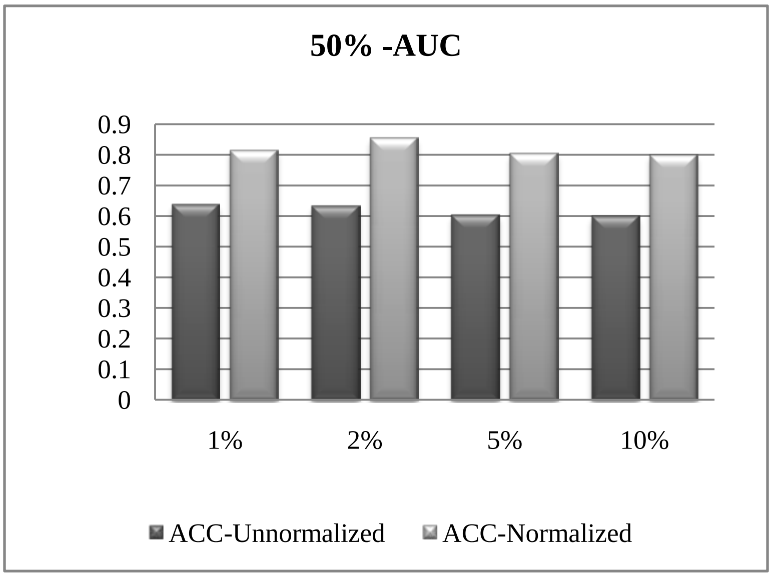


Figure 6-18 AUC result for 50% observed data for normalized and unnormalized Accuracy

Figure 6-19 presents the AUC measure for 75% hidden pairs. The result shows that the unnormalized measure is affected more when hiding more pairs that are positive. Table 6-21 show the complete compression result for the Accuracy measure.

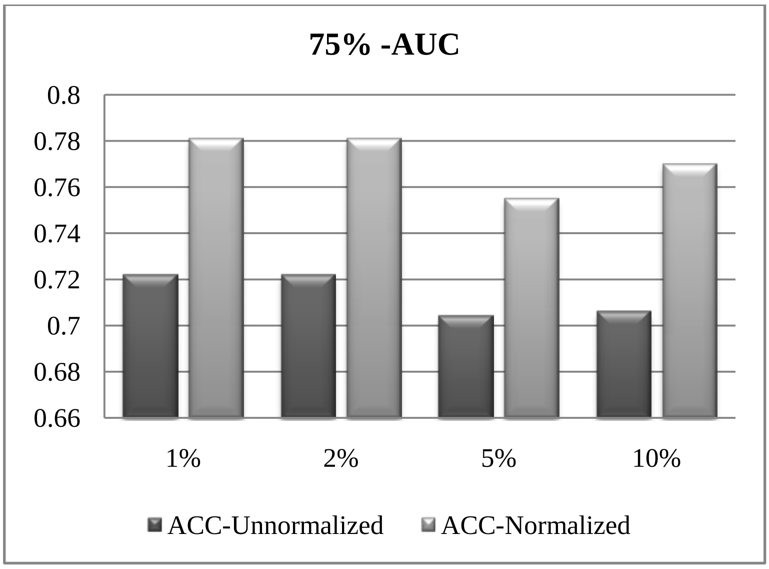


Figure 6-19 AUC result for 75% hiding data for normalized and unnormalized Accuracy

75%					
	method	Precision	Recall	F-Measure	AUC
1%	U- Accuracy	0.118	0.480	0.189	0.722
	N- Accuracy	0.135	0.600	0.221	0.781
	Different	14.41%	25.00%	16.93%	8.17%
	U- Leverage	0.117	0.380	0.179	0.676
	N- Leverage	0.138	0.760	0.234	0.856
	Different	17.95%	100.00%	30.73%	26.63%
2%	U- Accuracy	0.211	0.480	0.293	0.722
	N- Accuracy	0.238	0.600	0.341	0.781
	Different	12.80%	25.00%	16.38%	8.17%
	U- Leverage	0.192	0.340	0.245	0.656
	N- Leverage	0.235	0.730	0.356	0.841
	Different	22.40%	114.71%	45.31%	28.20%
5%	U- Accuracy	0.381	0.444	0.410	0.704
	N- Accuracy	0.416	0.548	0.473	0.755
	Different	9.19%	23.42%	15.37%	7.24%
	U- Leverage	0.381	0.352	0.366	0.662
	N- Leverage	0.445	0.760	0.561	0.856
	Different	16.80%	115.91%	53.28%	29.31%
10%	U- Accuracy	0.554	0.448	0.496	0.706
	N- Accuracy	0.601	0.578	0.589	0.770
	Different	8.48%	29.02%	18.75%	9.07%
	U- Leverage	0.545	0.342	0.420	0.657
	N- Leverage	0.617	0.762	0.682	0.857
	Different	13.21%	122.81%	62.38%	30.44%

Table 6-21 IB1-Classification measures for 75% hiding

(U- : Unnormalized, N- : Normalized)

In addition, we can observe another effect in the Leverage measure as shown in Figure 6-20 and Figure 6-21 that presents the AUC measure for 50% and 75% hidden. The result shows that the unnormalized measure is affected more when hiding more pairs that are positive. Table 6-21 shows the complete compression result for Leverage measure.

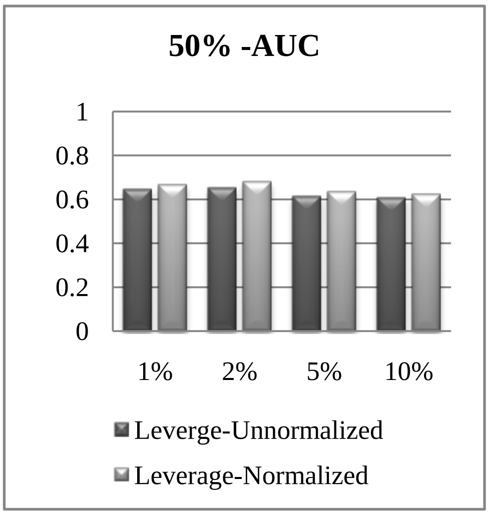


Figure 6-20 AUC result for 50% observed data for normalized and unnormalized Leverage

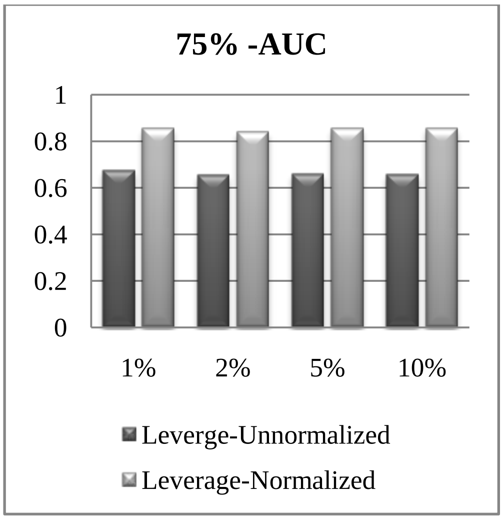


Figure 6-21 AUC result for 75% hiding data for normalized and unnormalized Leverage

Using all numerical features, in the case where we hide 50%, we see the difference between the normalized and unnormalized measures as shown in Table 6-22. In the next case where we hide a 75%, there is no difference as shown in Table 6-23 because the skewness of itemsets size becomes insignificantly recognizable. In general, the numerical feature records a significant result where the AUC record between 0.973 and 0.98 with 50% hidden datasets and between 0.873 and 0.89 in 75% hidden datasets.

50%					
%	method	Precision	Recall	F-Measure	AUC
1%	U- Accuracy	0.222	0.96	0.361	0.963
	N- Accuracy	0.221	0.980	0.360	0.973
	Different	-0.45%	2.08%	-0.28%	1.04%
2%	U- Accuracy	0.359	0.940	0.519	0.953
	N- Accuracy	0.364	0.990	0.532	0.978
	Different	1.39%	5.32%	2.50%	2.62%
5%	U- Accuracy	0.586	0.952	0.726	0.959
	N- Accuracy	0.588	0.988	0.737	0.977
	Different	0.34%	3.78%	1.52%	1.88%
10%	U- Accuracy	0.739	0.952	0.832	0.959
	N- Accuracy	0.742	0.994	0.850	0.980
	Different	0.41%	4.41%	2.16%	2.19%

Table 6-22 OneR-Classification measures for 50% hiding

The results further show the usefulness of using numerical features with proteins that share properties. The results obtained using normalized features are superior to those obtained using the original features (Aljandal, Hsu, & Xia, 2009).

75%					
%	method	Precision	Recall	F-Measure	AUC
1%	U- Accuracy	0.365	0.760	0.494	0.873
	N- Accuracy	0.365	0.760	0.494	0.873
	Different	0.00%	0.00%	0.00%	0.00%
2%	U- Accuracy	0.532	0.750	0.622	0.868
	N- Accuracy	0.532	0.750	0.622	0.868
	Different	0.00%	0.00%	0.00%	0.00%
5%	U- Accuracy	0.747	0.780	0.763	0.883
	N- Accuracy	0.747	0.780	0.763	0.883
	Different	0.00%	0.00%	0.00%	0.00%
10%	U- Accuracy	0.857	0.794	0.825	0.890
	N- Accuracy	0.857	0.794	0.825	0.890
	Different	0.00%	0.00%	0.00%	0.00%

Table 6-23 OneR Classification measures for 75% hiding

(U- : Unnormalized, N- : Normalized)

In my future work, I will continue working in the domain of protein-protein interaction by adding more numerical features extracted from repositories of biological information. In addition, there is a possibility of using a genetic algorithm (GA) to select from structural and biological features, which may lead to a further incremental boost in prediction quality.

6.6 Divided-by-N vs. Itemset Size-Sensitive Normalization Factor

In this experiment, I compare the divided-by-N as normalization factor with the itemset size-sensitive normalization factor (normalized using R) by using the classification framework. I use three inducers for classification Random Forest, OneR and IB1 where they are from different classification based models. On the other hand, in the *LiveJournal* dataset I have a large number of varieties of attributes' selection. Therefore, I have selected a number of attributes as friendship prediction features to represent experiment objectives which is a comparison between the two methods. I use four different attribute selections and present the classification result for each one (all result using 10-fold-cross validation):

The first result with the following attributes:

a. Support of Common Communities

b. Lift of Common Communities

c. Conviction of Common Communities

d. Leverage of Common Communities

Inducer	Method	Accuracy (%)	Precision	Recall	F-Measure	AUC
Random Forest	Divided by N	59.88	0.546	0.426	0.479	0.6
	Normalized by R	67.24	0.66	0.501	0.569	0.691
Percentage Different		12%	21%	18%	19%	15%
OneR	Divided by N	62.06	0.606	0.349	0.443	0.588
	Normalized by R	70.72	0.878	0.375	0.526	0.668
Percentage Different		14%	45%	8%	19%	14%
IB1	Divided by N	54.58	0.475	0.471	0.473	0.537
	Normalized by R	61.34	0.553	0.557	0.555	0.607
Percentage Different		12%	17%	18%	17%	13%

Table 6-24 Classification result obtained using four attributes (divided-by-N & itemset-size)

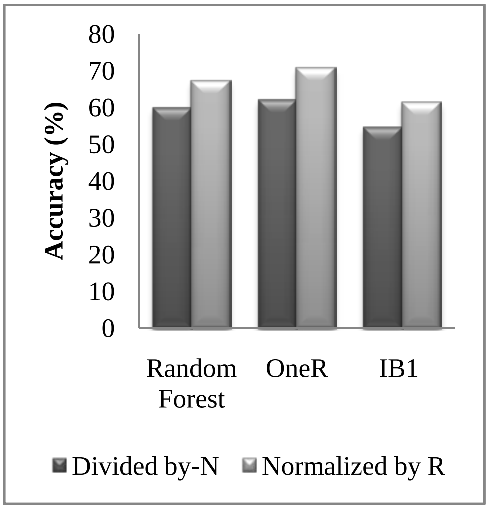


Figure 6-22 Accuracy obtained using four attributes (divided-by-N & itemset-size)

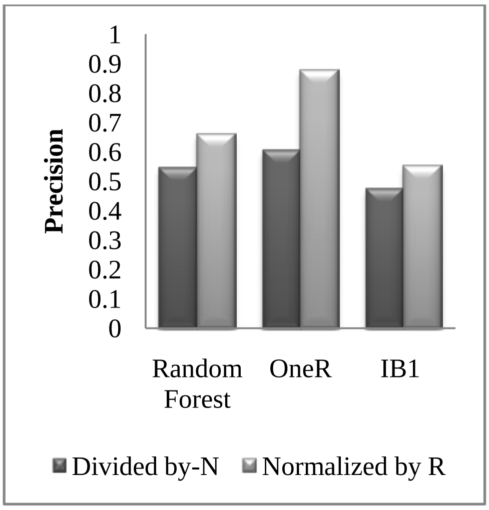


Figure 6-23 Precision obtained using four attributes (divided-by-N & itemset-size)

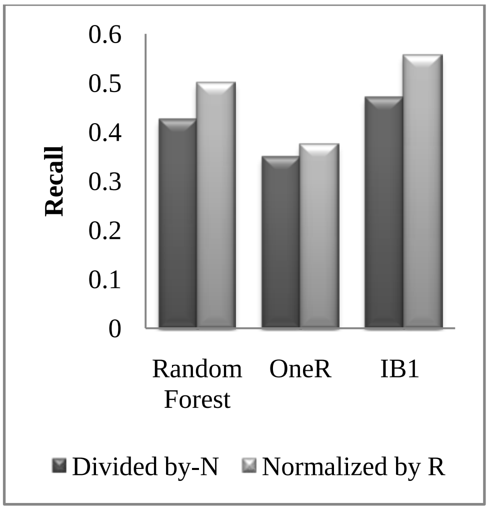


Figure 6-24 Recall obtained using four attributes (divided-by-N & itemset-size)

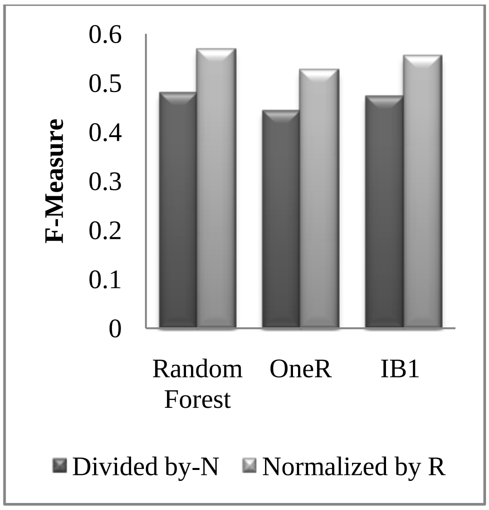


Figure 6-25 F-Measure obtained using four attributes (divided-by-N & itemset-size)

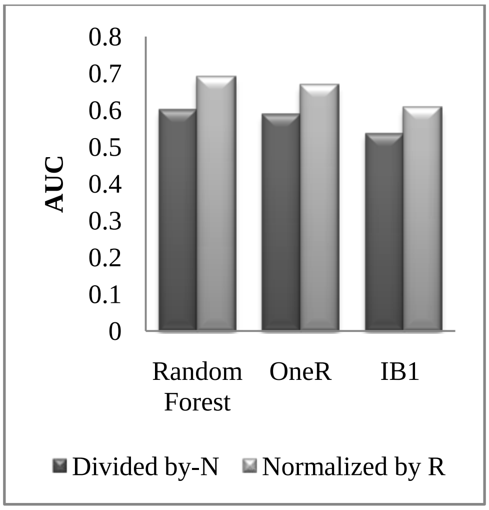


Figure 6-26 AUC obtained using four attributes (divided-by-N & itemset-size)

Superior results are achieved using my normalization factor compared to the divided-by-N method as shown in Figure 6-22 through Figure 6-26. The first result used four different attributes as friendship prediction features that are related to communities' membership (users' common communities). Table 6-24 shows the classification result with a percentage difference between the two methods (respect to my method). The percentage difference range for all classification measures are between 7.45% and 44.88% which shows the superiority of my method.

The second result has the attribute:

a) Leverage of Common Communities

Inducer	Method	Accuracy (%)	Precision	Recall	F-Measure	AUC
Random Forest	Divided by N	55.32	0.467	0.236	0.313	0.553
	Normalized by R	67.21	0.811	0.315	0.454	0.62
Percentage Different		22%	74%	33%	45%	12%
OneR	Divided by N	55.75	0.478	0.252	0.33	0.521
	Normalized by R	62.78	0.656	0.293	0.405	0.588
Percentage Different		13%	37%	16%	23%	13%
IB1	Divided by N	50.82	0.429	0.417	0.423	0.497
	Normalized by R	59.02	0.528	0.495	0.511	0.579
Percentage Different		16%	23%	19%	21%	17%

Table 6-25 Classification result for Leverage (divided-by-N & itemset-size)

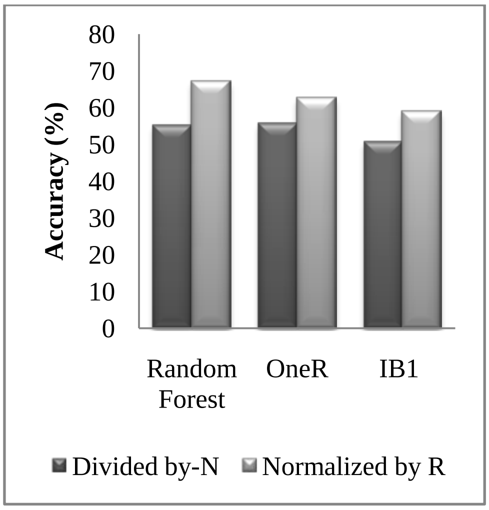


Figure 6-27 Accuracy obtained using Leverage (divided-by-N & itemset-size)

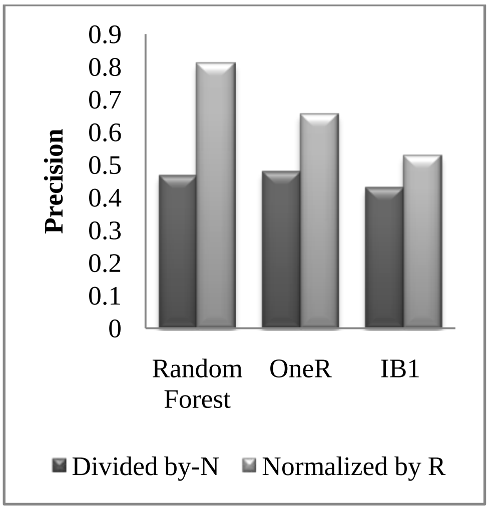


Figure 6-28 Precision obtained using Leverage (divided-by-N & itemset-size)

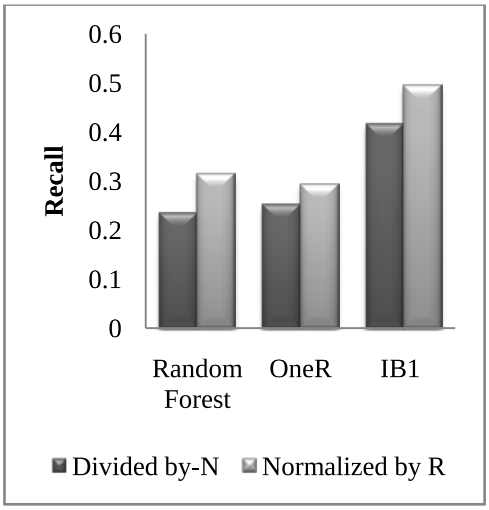


Figure 6-29 Recall obtained using Leverage (divided-by-N & itemset-size)

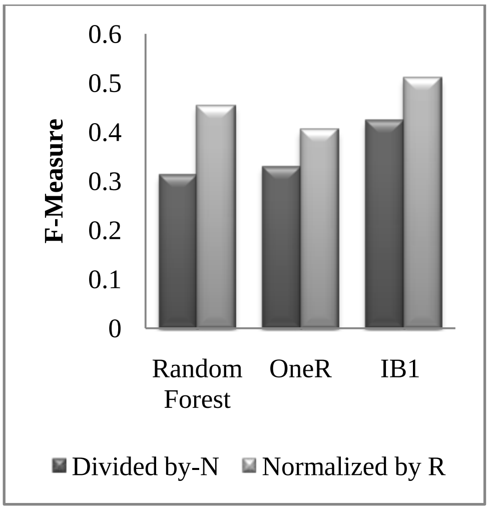


Figure 6-30 F-Measure obtained using Leverage (divided-by-N & itemset-size)

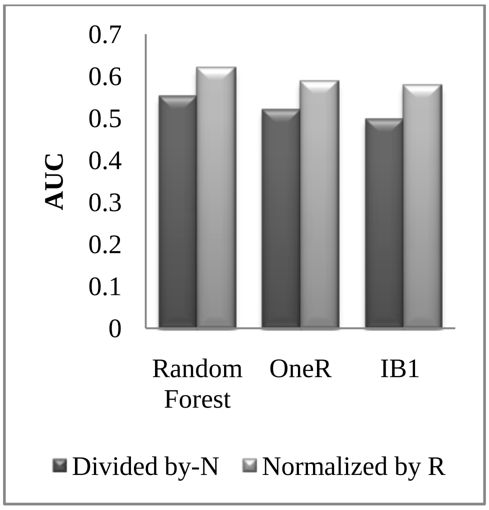


Figure 6-31 AUC obtained using Leverage (divided-by-N & itemset-size)

The second result used one attribute as a friendship prediction feature which is the Leverage of users' common communities as shown in Figure 6-27 through Figure 6-31. Table 6-25 shows the classification result with the percentage difference between the two methods (respect to my method). The percentage difference records superior results using the Normalized by R method in all measures and the range for that difference in all classification measures are between 12.12% and 73.66% which shows the superiority of my method.

The third result has the attribute:

a) Lift of Common Communities

Inducer	Method	Accuracy (%)	Precision	Recall	F-Measure	AUC
Random Forest	Divided by N	59.35	0.987	0.061	0.114	0.529
	Normalized by R	69.98	0.919	0.335	0.491	0.644
Percentage Different		18%	-7%	449%	331%	22%
OneR	Divided by N	59.38	0.988	0.061	0.116	0.53
	Normalized by R	70.97	0.92	0.36	0.518	0.668
Percentage Different		20%	-7%	490%	347%	26%
IB1	Divided by N	49.64	0.442	0.623	0.517	0.511
	Normalized by R	53.20	0.473	0.712	0.568	0.553
Percentage Different		7%	7%	14%	10%	8%

Table 6-26 Classification result obtained using Lift (divided-by-N & itemset-size)

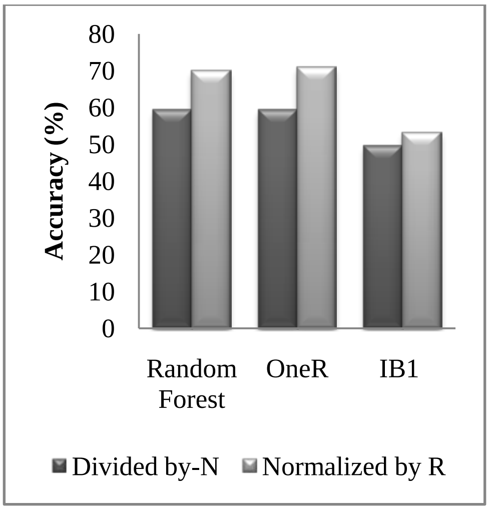


Figure 6-32 Accuracy obtained using Lift (divided-by-N & itemset-size)

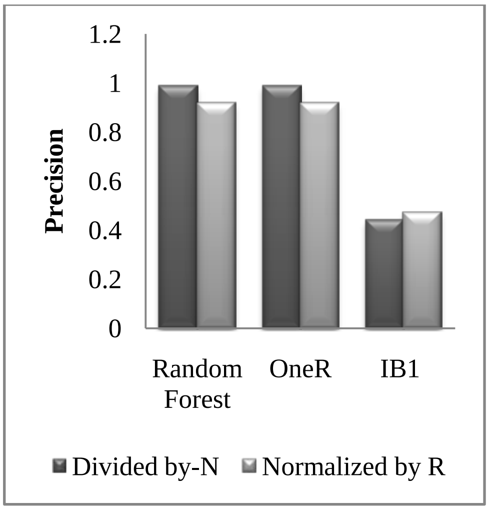


Figure 6-33 Precision obtained using Lift (divided-by-N & itemset-size)

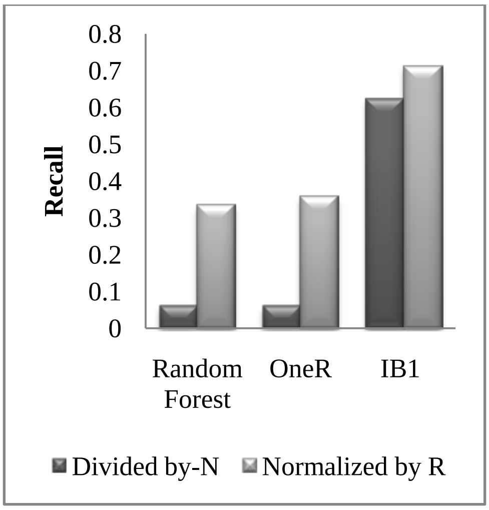


Figure 6-34 Recall obtained using Lift (divided-by-N & itemset-size)

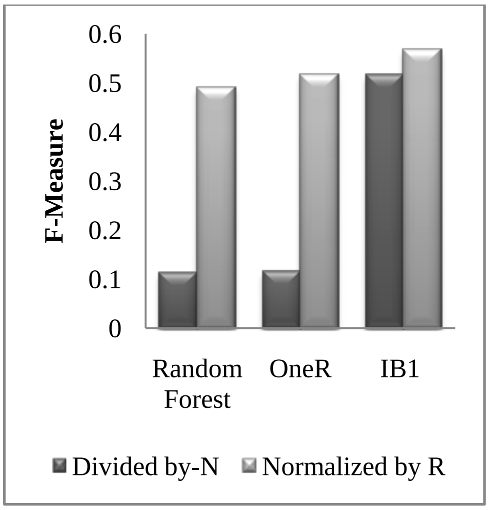


Figure 6-35 F-Measure obtained using Lift (divided-by-N & itemset-size)

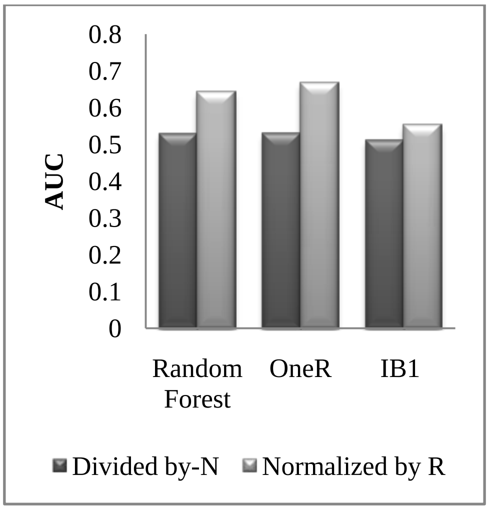


Figure 6-36 AUC obtained using Lift (divided-by-N & itemset-size)

The third result used one attribute as friendship prediction features which is the Lift of users' common communities as shown in Figure 6-40 through Figure 6-36. Table 6-26 shows the classification result with the percentage difference between the two methods (with respect to my method). The percentage difference records superior results to Normalized by R's method in all measures except in precision of two inducers Random forest and OneR, but when we look at the results we can see those who divided-by-N record a very low recall (0.016 in both inducers) compared with my method (0.335 in Random Forest and 0.36 in OneR). This is a more than 450% difference for both cases. However, the F-measure, which is a combination of precision and recall, shows superiority to my method in all inducers.

The last result has the attribute:

a) Support of common communities

Inducer	Method	Accuracy (%)	Precision	Recall	F-Measure	AUC
Random Forest	Divided by N	59.38	0.988	0.061	0.116	0.529
	Normalized by R	71.17	0.959	0.348	0.511	0.666
Percentage Different		20%	-3%	470%	341%	26%
OneR	Divided by N	59.388	0.988	0.061	0.116	0.53
	Normalized by R	71.17	0.958	0.349	0.511	0.669
Percentage Different		20%	-3%	472%	341%	26%
IB1	Divided by N	49.67	0.442	0.623	0.517	0.512
	Normalized by R	53.88	0.478	0.721	0.575	0.561
Percentage Different		9%	8%	16%	11%	10%

Table 6-27 Classification result obtained using Support (divided-by-N & itemset-size)

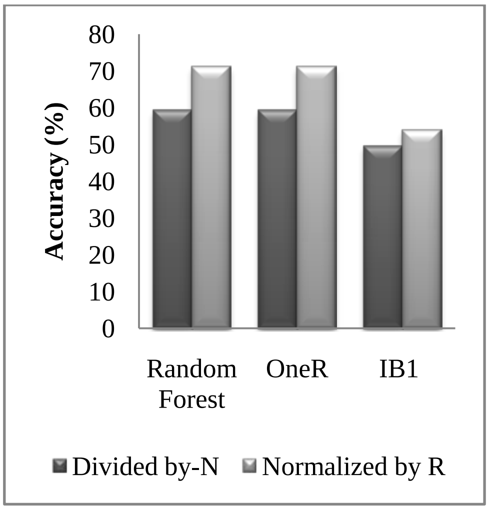


Figure 6-37 Accuracy obtained using Support (divided-by-N & itemset-size)

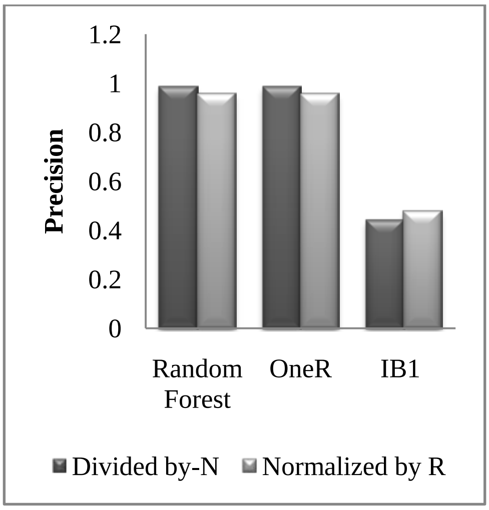
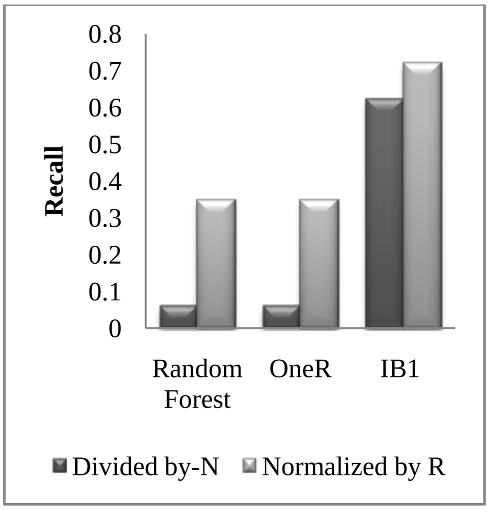


Figure 6-38 Precision obtained using Support (divided-by-N & itemset-size)



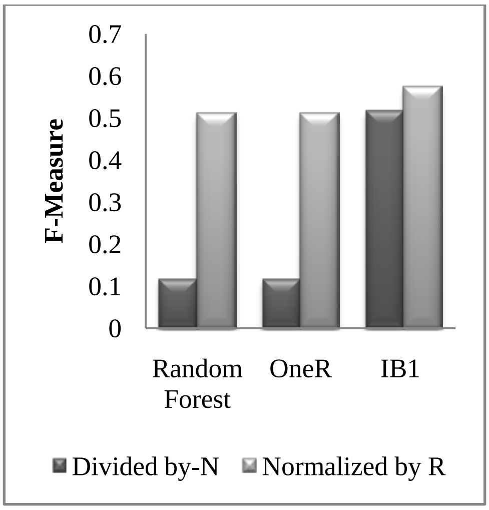


Figure 6-39 Recall obtained using Support (divided-by-N & itemset-size)

Figure 6-40 F-Measure obtained using Support (divided-by-N & itemset-size)

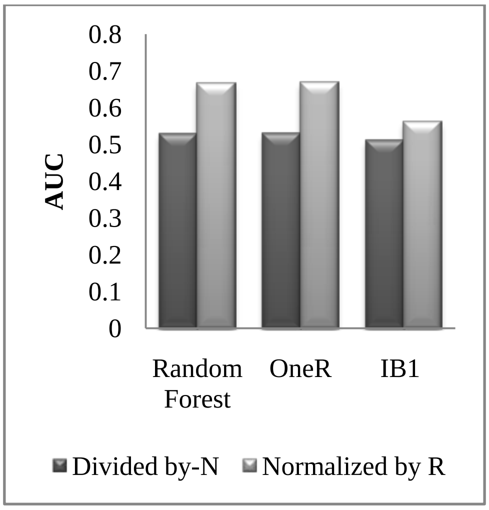


Figure 6-41 AUC obtained using Support (divided-by-N & itemset-size)

The last result used one attribute as friendship prediction features which is the Support of users' common communities as shown in Figure 6-37 through Figure 6-41. Table 6-27 shows the classification result with the percentage difference between the two methods (respect to my method). The percentage difference records superior results to Normalized-by-R method in all measures except in precision of two inducers Random forest and OneR (as the previous result), but when we look at the result we can see that in these cases divided-by-N achieves a very low recall (0.016 in both inducers) compared with my method (0.348 in Random Forest and 0.349 in OneR). This is more than 470% different for both cases. However, the F-measure, which is a combination of precision and recall, shows super result using my method with all inducers.

6.7 Numerical Features for Interest Prediction

In this experiment, I evaluate the nominal and numerical features using five classifier models and inductive learning algorithms: support vector machines (SVM), Logistic Classification, Random Forests, decision trees (J48), and decision stumps (OneR). Table 6-28 and Table 6-29 list the results for SVM and Logistic Classification, which achieved the highest ROC-AUC score using all available features. The overall highest AUC was achieved using numerical features along with Logistic Classification, although the precision is still improved by the inclusion of nominal features for other classifier models as showen in Table 6-30, Table 6-31 and Table 6-32 (Aljandal, Bahirwani, Caragea, & Hsu, 2009).

Nom	Num	Precision	Recall	F-Measure	AUC
*		0.617	0.693	0.601	0.558
	*	0.829	0.826	0.817	0.918
*	*	0.833	0.838	0.829	0.921

Table 6-28 Classification results using SVM for interest prediction

(Nom: Nominal, Num : Numerical)

Nom	Num	Precision	Recall	F-Measure	AUC
*		0.618	0.684	0.611	0.570
	*	0.838	0.846	0.839	0.924
*	*	0.845	0.844	0.843	0.919

Table 6-29 Classification results using Logistic classifier for interest prediction

Nom	Num	Precision	Recall	F-Measure	AUC
*		0.623	0.721	0.646	0.529
	*	0.849	0.805	0.816	0.845
*	*	0.828	0.817	0.820	0.838

Table 6-30 Classification results using J48 for interest prediction

Nom	Num	Precision	Recall	F-Measure	AUC
*		0.613	0.671	0.638	0.572
	*	0.801	0.819	0.810	0.878
*	*	0.818	0.843	0.829	0.894

Table 6-31 Classification results using Random Forest for interest prediction

Nom	Num	Precision	Recall	F-Measure	AUC
*		0.565	0.684	0.600	0.513
	*	0.825	0.805	0.812	0.807
*	*	0.825	0.805	0.812	0.807

Table 6-32 Classification results using OneR for interest prediction

CHAPTER 7 - **CONCLUSIONS AND FUTURE WORK**

In this chapter I present conclusions drawn from my framework development process and experimental results, and relate these findings to previous work. Then I will discuss some areas of research that are worth investigating as future work.

7.1 Conclusion

Discovering associations from data is one research challenge that needs more sophisticated measures that can capture interesting patterns. Previous work in the area of subjective measures for association rules reveals a great diversity of applicable statistical methods these can be applied to build different measures to meet users' needs in different domains. Deriving an objective interestingness measure usually involves estimating some aspect of a candidates rule structure, analytical performance and statistical significance with respect to observed itemset data. Compound measures are based on primitive measures grounded in probability density functions, with some – such as the normalization approach that I described- based on parametric fusion of these primitive measures, while others are based on more *ad hoc* rules of combination.

In this dissertation, I investigated the property of the itemset size and drive a normalization factor that increases the sensitivity of joint probability. Also I demonstrated an improvement achieved by driving a generic itemset size-sensitive joint probability that is compatible with several existing interestingness measures. Related work in this area began with Piatetsky-Shapiro (1991), who first proposed using statistical independence of rules as an interestingness measure. The Itemset size property has not been involved in any of the objective measures. More methods have since been proposed using different statistical approaches. Brin, *et al.* (1997) proposed lift and $\chi 2$ (chi-squared) as correlation measures and developed an efficient mining method. Hilderman, *et al.* (2001) and Tan, *et al.* (2002) have comparative studies of different interestingness measures. Therefore, I showed that normalized measures (using itemset size-sensitive joint probability) increase the sensitivity of the interestingness measure to the distribution of data in the context of itemset size, thus improving upon different measures. As a consequence of the Tan, Kumar and Srivastava's (2002) study that ends with the conclusion that there is no measure that is consistently better than others in all application domains, I have used this method with several datasets in different domains such as market basket, social network and bioinformatics (domains with an autonomous or semi-autonomous property). Regarding some exogenous variables in some domain which may affect the property of itemset size, the result that I obtained is statistically significant in improving the sensitivity of interestingness measurers and enhances numerical features in link prediction problem using classification methods. In addition, I use ontology to make explicit relationships between various interests in item sets, because my normalization method is related to the itemset size. Therefore, I showed an improvement using this interest's ontology for predicting link existence such as friendship relations in the *LiveJournal* data set.

The problem of link prediction in most real-world domains depends upon the availability of features that can assist in building effective classification models. In previous work, different classification approaches as have been proposed, such as "simple yet general" the collective classification algorithm (Bilgic, Namata, & Lise, 2007) which addresses problems in object classification and link prediction. In another study, using graph features as done by Hsu *et al.* (2006), can be useful in some domain. In this dissertation I investigated using interestingness measurers as numerical features. However, using numerical features, which are association measures for some instance properties, extend the improvement achieved by other features such as graph features. In addition, I achieved further improvement with using itemset size-sensitive normalized measures in some domains. As I showed in predicting friendship relation in *LiveJournal* data set, the results that Hsu *et al.* (2006) achieved by using graph features are further improved when I add normalized numerical features with ontology. The quality of results consistently exceeds that achieved by using unnormalized measures.

In the area of bioinformatics, numerical features in my experiments (overall) show a significant prediction result. In my experiments, I used numerical features (association measures) from parent-child relationship in protein-protein interaction network to build a prediction model. The results of constructing a variety of numerical features from different data properties give support to our view about this technique and the ability to be adopted in different domains.

This research shows the effectiveness of using Itemset size properties with interestingness measures in some domains. The resultant effect is to increases the sensitivity of the interestingness measure to the distribution of data in the context of itemset size. Although, as I discussed in Chapter 4, there are some domains affected by uncontrollable exogenous variables that weaken the item size property effects. However, it will be important for our future discovery methods to capture some of the domain-specific semantics of links and itemset membership systematically. Finally, I introduce two evaluation frameworks based on principles of information theory and classification models, to cover both domains that have ground truth and this that lack it.

7.2 Future work

There are many opportunities and challenges for continued investigation related to link mining and classification using association rule mining techniques and measures. With my research there are many points that can be extended in future research.

- Association rule mining and domain specific: I have studied the itemset size property and the effect that can be taken in account when we are looking for associations in some domains. Still, there are many domains that need be investigated which raise the key question that I am continuing to explore: how the domain-specific semantics of links and itemset membership affect the mining process in some of the domains as I mentioned in chapter 4.1.
- Investigate the impact of the Itemset size normalization approach on other interestingness measures, datasets, and association and classification tasks. Therefore, the Table 4-2 shows some domains and possible links and itemset preparation that can be used.
- Itemset size-sensitive joint probability derived from itemset size relation. One of my future goals is to optimize the equation that I introduced to another equation derived from a learning model.
- In the section where I used a user interest ontology in social network, some possible future work can examine how to extend the framework to incorporate multi-word user interests and technical definitions. There are other memberships that may also benefit from ontology discovery. Examples of pair-dependent attributes include measures of overlap among common:
    - Communities, forums, groups
    - Fandoms (fan of), endorsements (supporter of)
    - Institutions (schools, colleges and universities, companies, etc.)

In particular, fandoms and communities have their own description pages and metadata in most social networks that make it worth investigation.

Ontology that includes temporal fluents such as part-of (“Blogger became part of Google in 2004”) and use them to infer relational fluents (“u and v have been Google employees since 2004”) will allow us to construct semantically richer feature sets that I believe will be more useful for link existence and persistence prediction.

In another aspect, structure of ontology hierarchy can be used as a description of a multi level semantic relation. This aspect can be seen, for example, in user interests in social network where users who are interested in “JAVA” and others interested in “C#”, are all interested in “Computer Programming Language” based on semantic concepts (i.e. ontology). However, the relations between users who share interest in low level “C#” are suppose to be more interested than relations between users who share interest in higher level “Computer Programming Language”. It is worth investigating whether taking into account hierarchy levels in relation discovery, can increase prediction sensitivity.

The association rule mining approach and the semantics of itemset size extend naturally to different domains, making these a promising area for exploration of ontology-aware classification in order to be able to account for the relationship between membership popularity and significance towards link existence.

- In the area of bioinformatics, there is a massive amount of data still needing to be processed. I use a numerical feature constructed from link structure (parent-child relation) to predict protein interaction. However, there is more information that we can construct numerical features from such as Protein functions, domains and sequences.
- The problem of link prediction in different domains such as social networks and bioinformatics can be addressed by using a variety of features, e.g. interest-based features and graph-based features. I have shown that incorporating semantic knowledge into interest-based features helps improve the performance of classifiers trained to predict friends for example. However, the area of link prediction using numerical features of some available

properties can be more permissible in some domains which make this a researchable area.

Appendix A - DETAILED RESULT FOR PREDICTING FRIENDS USING INTERESTS

A-1 Detailed result of IB1 with One attribute (common interest)

Table A-1 shows the detailed results for each fold. I visualize the result in the Figure A-1, Figure A-2 and Figure A-3 for precision, recall and f-measure respectively. We can see from the figures that the new method “Normalized” record better result for most of the folds.

Fold	precision- WithR	precision WithoutR	recall WithR	recall WithoutR	F-measure WithR	F-measure WithoutR
1	0.458333	0.263598	0.300000	0.286364	0.362637	0.274510
2	0.520468	0.296928	0.404545	0.395455	0.455243	0.339181
3	0.257410	0.221106	0.750000	0.600000	0.383275	0.323133
4	0.550725	0.555556	0.345455	0.068182	0.424581	0.121457
5	0.562963	0.355140	0.345455	0.172727	0.428169	0.232416
6	0.596386	0.414634	0.450000	0.231818	0.512953	0.297376
7	0.542857	0.277273	0.345455	0.277273	0.422222	0.277273
8	0.525926	0.228346	0.322727	0.263636	0.400000	0.244726
9	0.570423	0.255000	0.368182	0.231818	0.447514	0.242857
10	0.557047	0.653846	0.375566	0.076923	0.448649	0.137652

Table A-1 IB1 One attributes (common interest) -for each fold

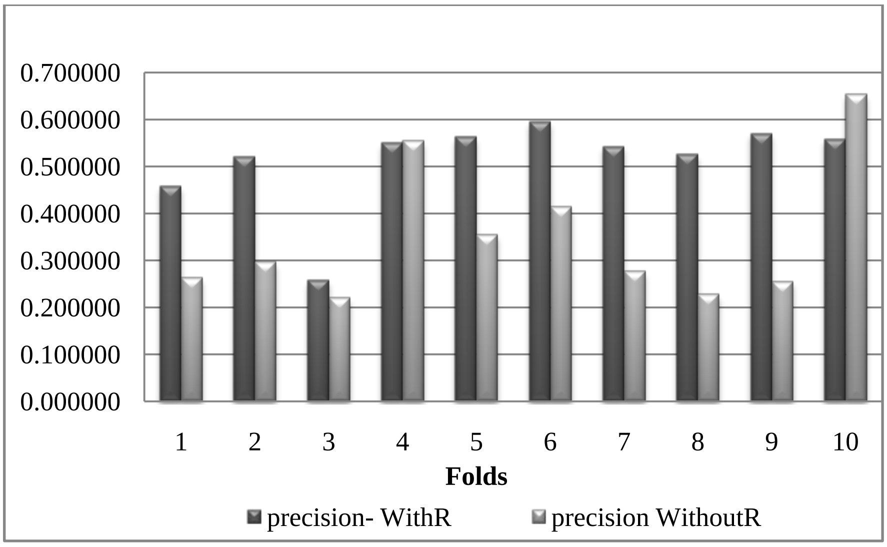


Figure A-1 Precision of IB1 (common interests)- all folds

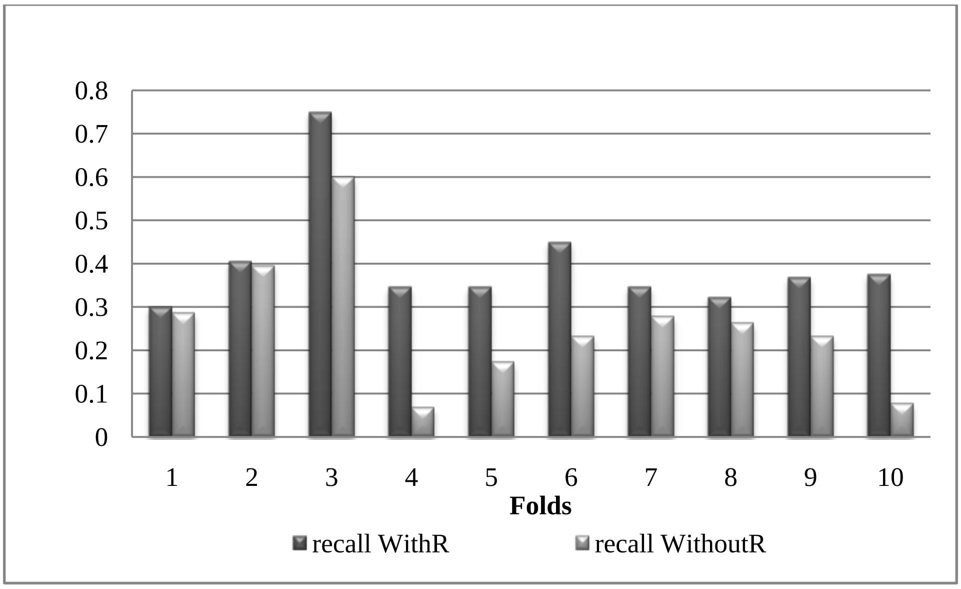


Figure A-2 Recall of IB1 (common interests)- all folds

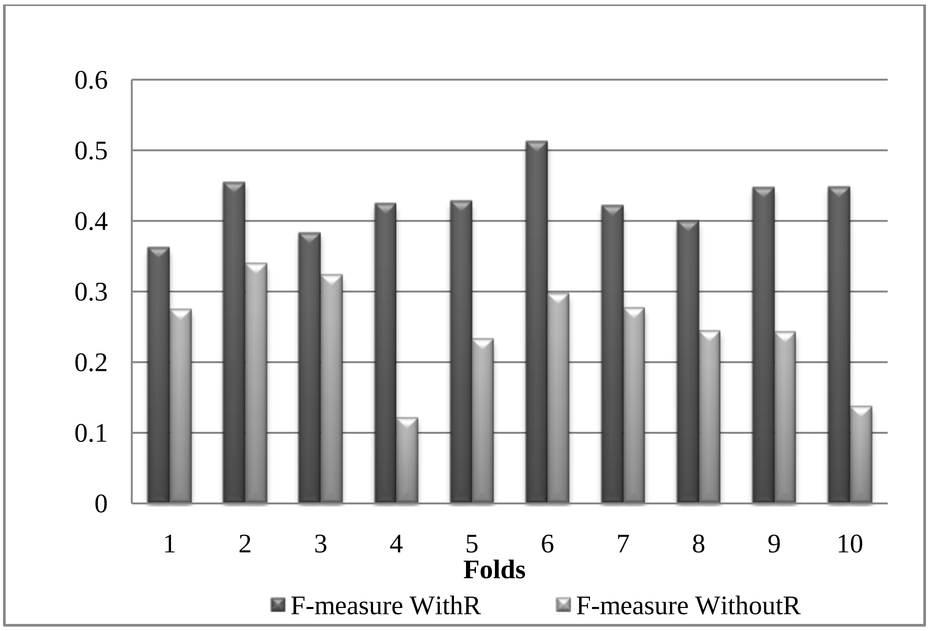


Figure A-3 F-measure of IB1 (common interests)- all folds

For the T-test results are shown in Table A-2, Table A-3 and Table A-4 for precision, recall and F-measure respectively. All of the T-test results are computed at the 95% level of confidence (i.e. set alpha level of 0.05). The T-test result shows that there is a significant difference between the two results. For example the p-value of F-measure test is 0.00004 and the recall's p-value for recall is 0.001. All results of the p-value are less than 0.05, which is the significance threshold.

	precision-WithR	*precision WithoutR*
Mean	0.51425	0.35214
Variance	0.00950	0.02164
Observations	10	10
df	9	
t Stat	3.72336	
P(T<=t) one-tail	0.00237	
t Critical one-tail	1.83311	

Table A-2 T-test of precision (IB1 - common interests)

	recall WithR	recall WithoutR
Mean	0.40074	0.26042
Variance	0.01684	0.02383
Observations	10	10
df	9	
t Stat	4.29377	
P(T<=t) one-tail	0.00100	
t Critical one-tail	1.83311	

Table A-3 T-test of Recall (IB1 - common interests)

	F-measure WithR	*F-measure WithoutR*
Mean	0.42852	0.24906
Variance	0.00176	0.00516
Observations	10	10
df	9	
t Stat	6.77389	
P(T<=t) one-tail	0.00004	
t Critical one-tail	1.83311	

Table A-4 T-test of F-measure (IB1 - common interests)

A-2 Detailed result of Random Forest with One attribute (common interest)

Table A-5 shows more details regarding application of the Random-Forest inducer with one attribute. In this experiment the precision measure recorded better results in Figure A-4 for the original method “Unnormalized” which is the same situation as in the 5.2.1 experiment. However, recall and F-measure have more significant improvements are achieved for using the new method “Normalized” as shown in Figure A-5 and Figure A-6. These improvements have been validated using the T-test.

Fold	precision-WithR	precision WithoutR	recall WithR	recall WithoutR	F-measure WithR	F-measure WithoutR
1	0.5816	0.7222	0.2591	0.0591	0.3585	0.1092
2	0.5613	0.7200	0.3955	0.0818	0.4640	0.1469
3	0.6228	0.6818	0.3227	0.0682	0.4251	0.1240
4	0.5966	0.7500	0.3227	0.0682	0.4189	0.1250
5	0.6396	0.6061	0.3227	0.0909	0.4290	0.1581
6	0.6800	0.5000	0.3864	0.0773	0.4928	0.1339
7	0.5564	0.6087	0.3364	0.0636	0.4193	0.1152
8	0.5652	0.6250	0.2955	0.0455	0.3881	0.0847
9	0.5802	0.8000	0.3455	0.0545	0.4330	0.1021
10	0.6198	0.7308	0.3394	0.0860	0.4386	0.1538

Table A-5 Random Forest for One attributes (common interest) -for each fold

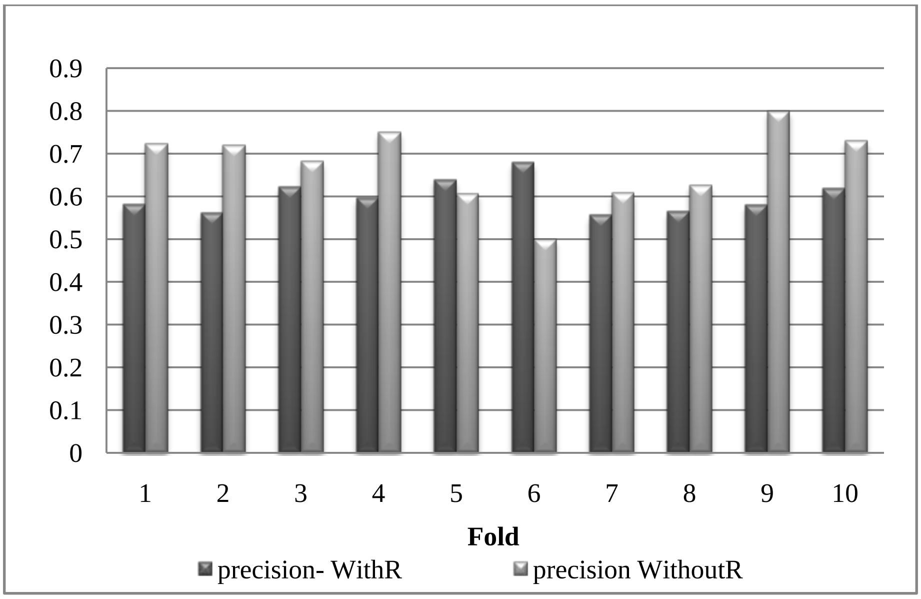


Figure A-4 Precision of Random Forest (common interests)- all folds

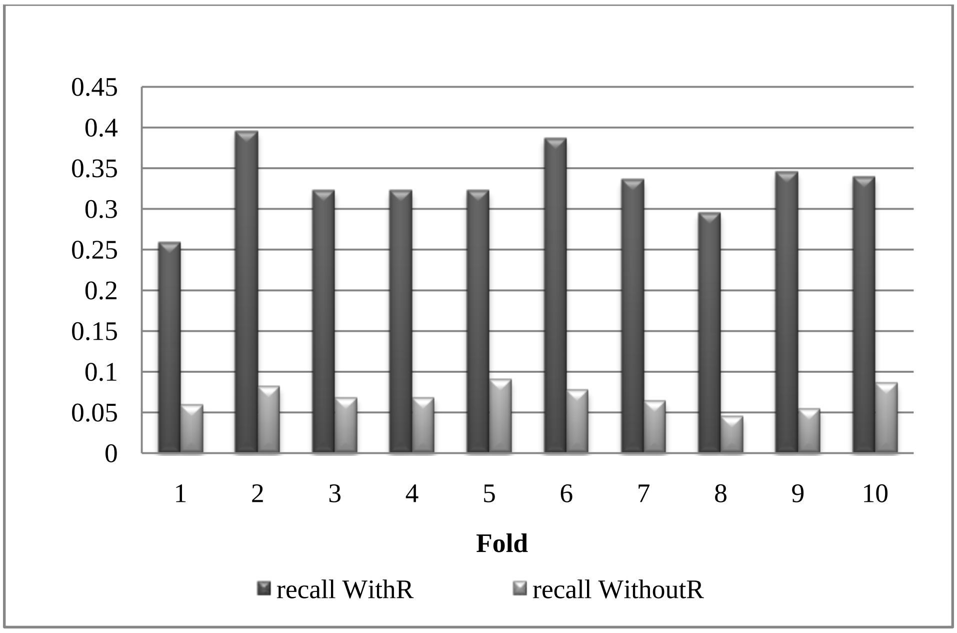


Figure A-5 Recall of Random Forest (common interests)- all folds

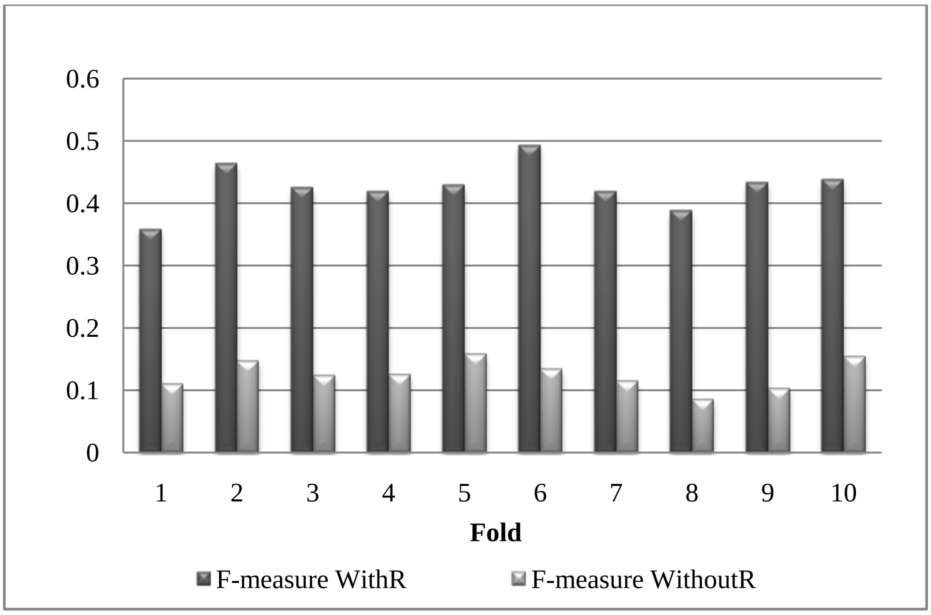


Figure A-6 F-measure of Random Forest (common interests)- all folds

For the T-test results shown in Table A-6, Table A-7 and Table A-8 for precision, recall and F-measure respectively. All of the T-test result with 95% confidence (set the alpha level at 0.05). The T-test results show that there is a significant positive difference between the two methods results except the precision measure that has negative result. For example the p-value of F-measure test is “8.83E-11” also the recall’s p-value is “9.40E-10”. The result of the p-value is less than 0.05 which is the significance point. The last two results (recall and F-measure) are much less than the first one for the precision measure.

	precision-WithR	*precision WithoutR*
Mean	0.60036	0.67446
Variance	0.00158	0.00788
Observations	10	10
Df	9	
t Stat	-2.0504	
P(T<=t) one-tail	0.0353	
t Critical one-tail	1.8331	

Table A-6 T-test of precision (Random Forest - common interests)

	recall WithR	*recall WithoutR*
Mean	0.33257	0.06951
Variance	0.00157	0.00021
Observations	10	10
df	9	
t Stat	23.8960	
P(T<=t) one-tail	9.40E-10	
t Critical one-tail	1.8331	

Table A-7 T-test of Recall (Random Forest - common interests)

	F-measure WithR	*F-measure WithoutR*
Mean	0.42672	0.12531
Variance	0.00136	0.00055
Observations	10	10
df	9	
t Stat	31.1681	
P(T<=t) one-tail	8.83E-11	
t Critical one-tail	1.8331	

Table A-8 T-test of F-measure (Random Forest - common interests)

A-3 Detailed result of OneR with One attribute (Lift)

Table A-9 shows more details about applying OneR inducer with one attribute (Lift). In this experiment all measures (precision, recall and F-measure) record a better result for the new method “Normalized” comparing with the result of original “Unnormalized” as shown in Figure A-7, Figure A-8 and Figure A-9.

Fold	precision- WithR	precision WithoutR	recall WithR	recall WithoutR	F-measure WithR	F-measure WithoutR
1	0.56338	0.45455	0.18182	0.11364	0.27491	0.18182
2	0.63736	0.49315	0.26364	0.16364	0.37299	0.24573
3	0.60526	0.46809	0.20909	0.10000	0.31081	0.16479
4	0.61446	0.54054	0.23182	0.09091	0.33663	0.15564
5	0.60811	0.44262	0.20455	0.12273	0.30612	0.19217
6	0.59701	0.43284	0.18182	0.13182	0.27875	0.20209
7	0.53571	0.46667	0.20455	0.12727	0.29605	0.20000
8	0.59375	0.50000	0.17273	0.14091	0.26761	0.21986
9	0.57692	0.54839	0.20455	0.15455	0.30201	0.24113
10	0.60000	0.38462	0.20362	0.09050	0.30405	0.14652

Table A-9 OneR One attributes (Lift) -for each fold

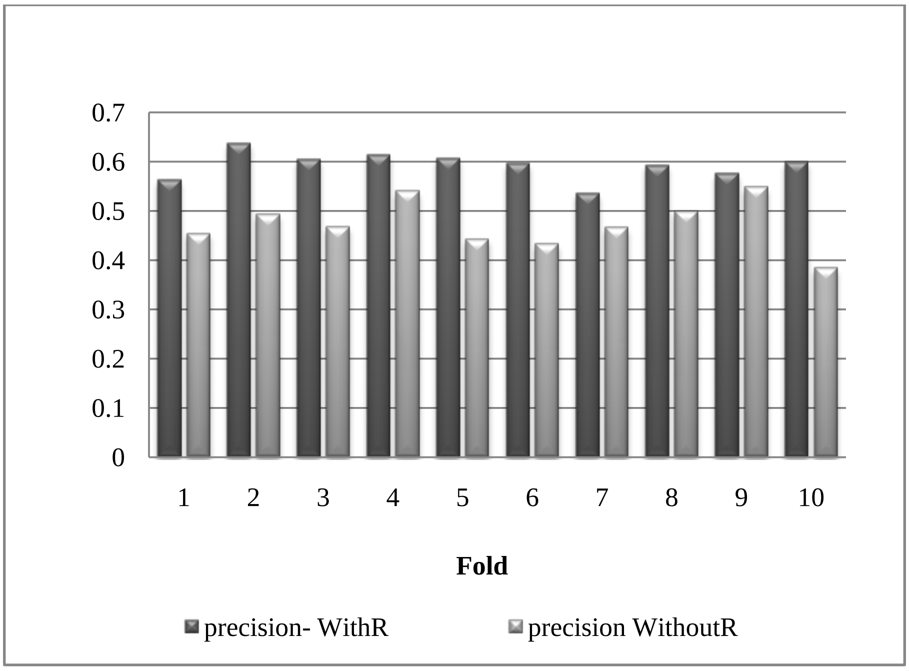


Figure A-7 Precision of OneR (Lift)- all folds

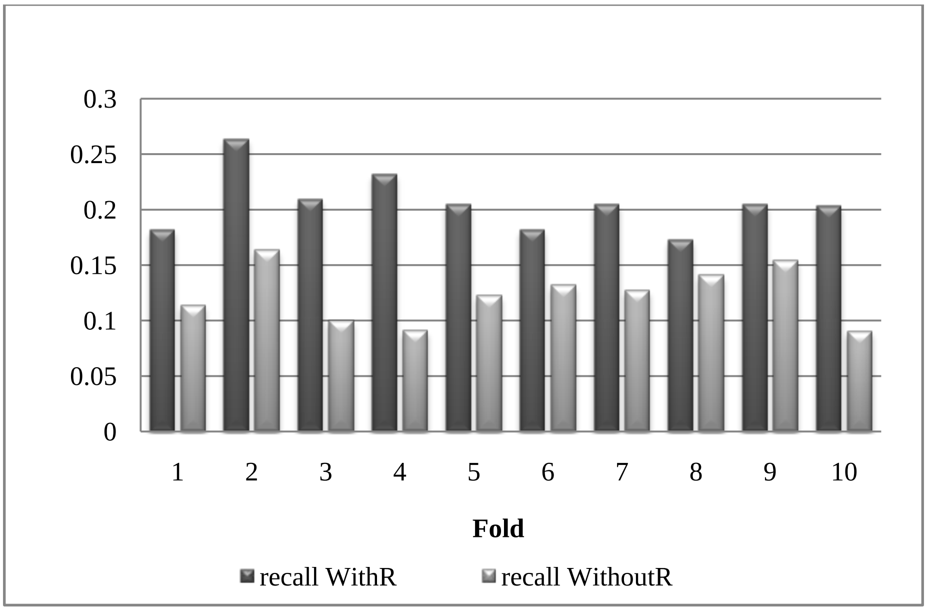


Figure A-8 Recall of OneR (Lift)- all folds

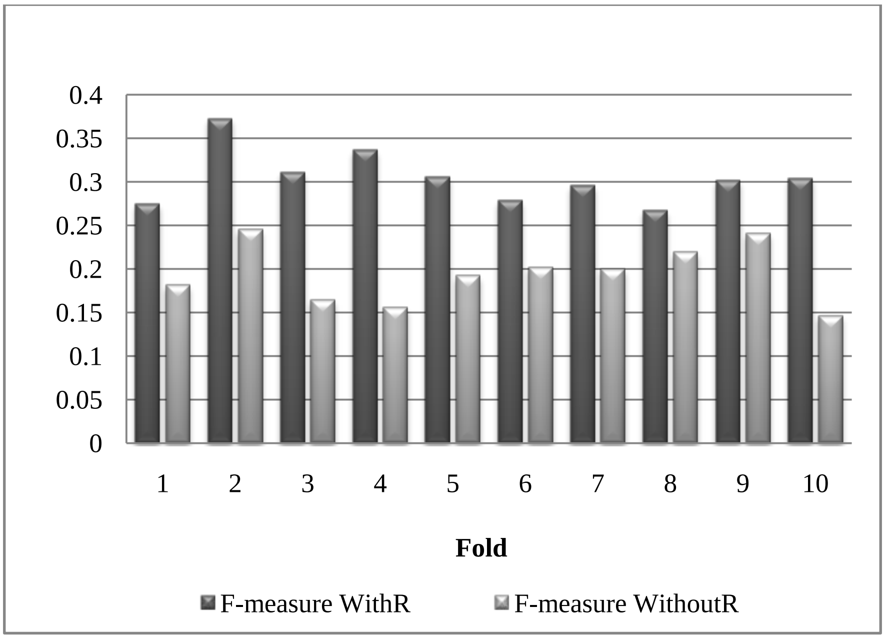


Figure A-9 F-measure of OneR (Lift)- all folds

For the T-test results shown in Table A-10, Table A-11 and Table A-12 for precision, recall and F-measure respectively. All of the T-test result with 95% confident (set the alpha level at 0.05). The T-test results show that there is a significantly positive difference between the two method results in all measures. For example the p-value of F-measure test is “1.05E-05” also the recall’s p-value is “1.51E-05”. The result of the p-value is less than 0.05 which is the significant point.

	precision- WithR	*precision WithoutR*
Mean	0.59320	0.47314
Variance	0.00081	0.00246
Observations	10	10
df	9	
t Stat	6.81860	
P(T<=t) one-tail	3.87E-05	
t Critical one-tail	1.83311	

Table A-10 T-test of precision (OneR - Lift)

	recall WithR	*recall WithoutR*
Mean	0.20582	0.12360
Variance	0.00070	0.00064
Observations	10	10
df	9	
t Stat	7.69480	
P(T<=t) one-tail	1.51E-05	
t Critical one-tail	1.83311	

Table A-11 T-test of Recall (OneR - Lift)

	F-measure WithR	*F-measure WithoutR*
Mean	0.30499	0.19498
Variance	0.00097	0.00115
Observations	10	10
df	9	
t Stat	8.05481	
P(T<=t) one-tail	1.05E-05	
t Critical one-tail	1.83311	

Table A-12 T-test of F-measure (OneR - Lift)

A-4 Detailed result of IB1 with One attribute (Lift)

Table A-13 shows more details about applying IB1 inducer with one attribute (Lift). In this experiment all measures (precision, recall and F-measure) achieved a better result using the new method “Normalized” compared with the result of original “Unnormalized” results shown in Figure A-10, Figure A-11, and Figure A-12.

fold	precision WithR	precision WithoutR	recall WithR	recall WithoutR	F-measure WithR	F-measure WithoutR
1	0.466960	0.428571	0.481818	0.381818	0.474273	0.403846
2	0.514286	0.391304	0.572727	0.450000	0.541935	0.418605
3	0.500000	0.382353	0.513636	0.413636	0.506726	0.397380
4	0.447552	0.415730	0.581818	0.504545	0.505929	0.455852
5	0.443089	0.388889	0.495455	0.445455	0.467811	0.415254
6	0.537383	0.422111	0.522727	0.381818	0.529954	0.400955
7	0.510204	0.393574	0.568182	0.445455	0.537634	0.417910
8	0.469136	0.385650	0.518182	0.390909	0.492441	0.388262
9	0.483193	0.406699	0.522727	0.386364	0.502183	0.396270
10	0.518349	0.384259	0.511312	0.375566	0.514806	0.379863

Table A-13 IB1 for One attributes (Lift) -for each fold

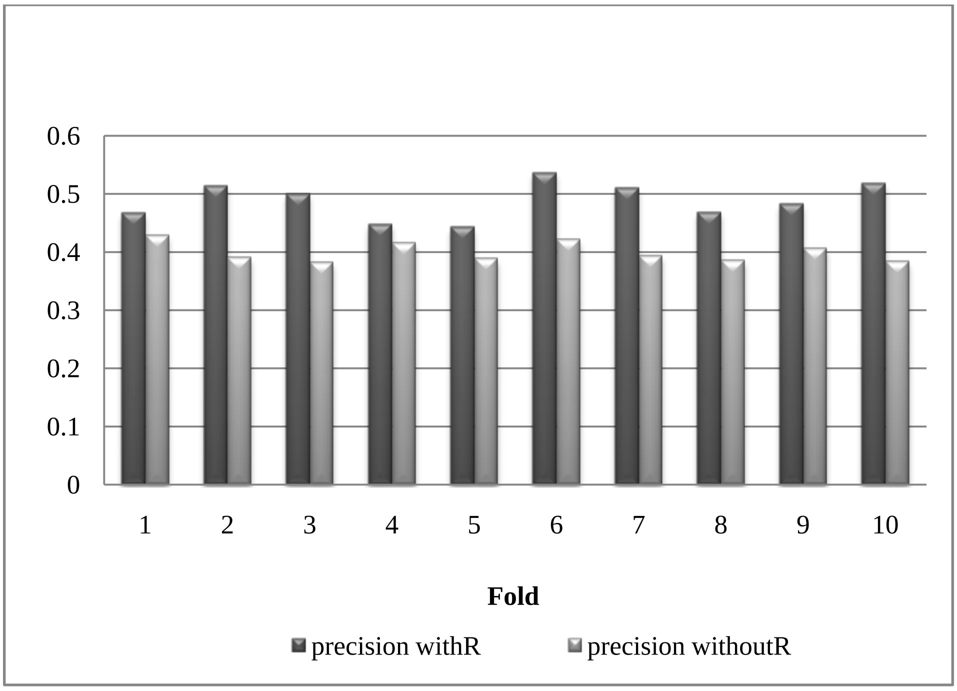


Figure A-10 Precision of IB1 (Lift)- all folds

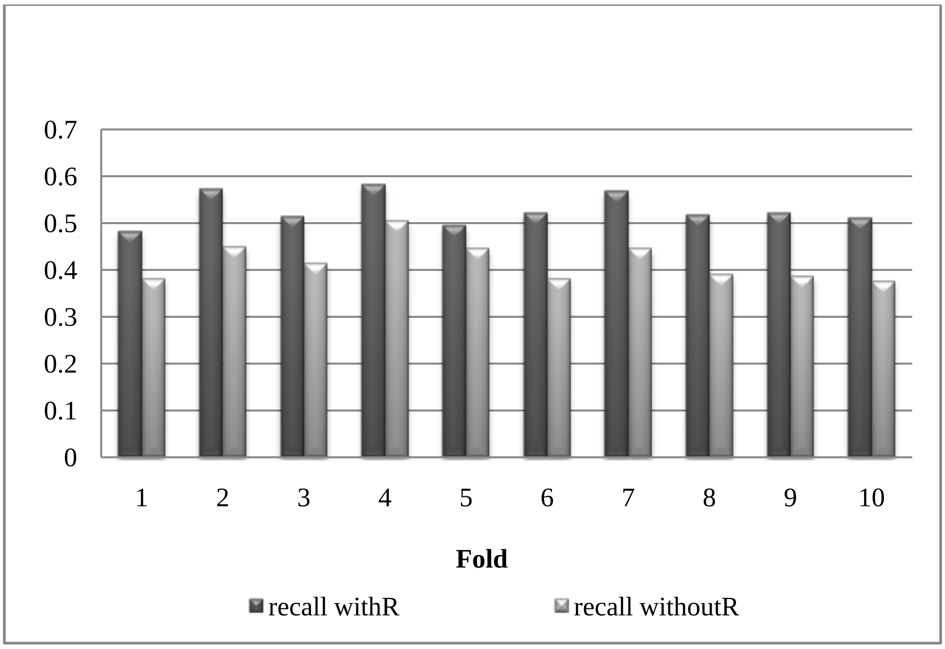


Figure A-11 Recall of IB1 (Lift)- all folds

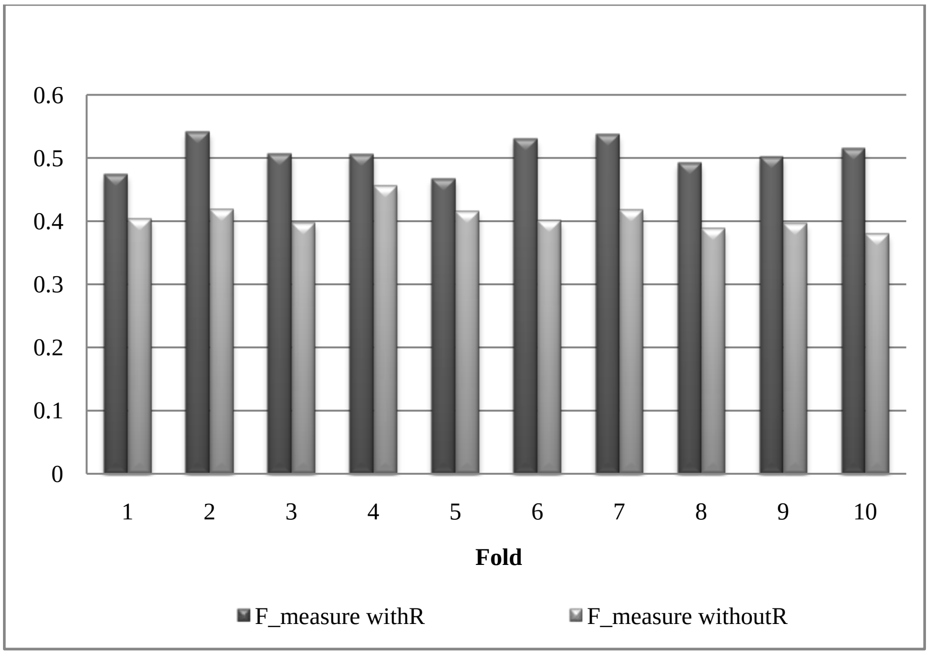


Figure A-12 F-measure of IB1 (Lift)- all folds

In conclusion, the results of these experiment shows how the itemset size-sensitive normalization factor (withR) affects the final result and makes the interestingness measure more sensitive to the behavior of data. I used this method with a small set of data and observed a positive result of improvements for the interestingness measures.

However, the new method can have a different effect when we use it with other interestingness measures based on how these measures deal with joint probability.

REFERENCES

Aggarwal, C. C., & Yu, P. S. (1998). A new framework for itemset generation. *Proceedings of the 1998 ACM symposium on principles of database systems (PODS'98)* (pp. 18–24). Seattle,WA: ACM.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proc. 20th Intl. Conference on Very Large Databases (487--499).*

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD International Conference on Management of Data* (pp. 207-216). Washington DC (USA): ACM SIGMOD.

Agrawal, R., Mannila, H., R., S., Toivonen, H., & Verkamo, A. I. (1996). Fast discovery of association rules,. *In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Editors), Advances in Knowledge Discovery and Data Mining. AAAI Press.*, (pp. 307-328). Menlo Park, California , USA.

Aha, D., & Kibler, D. (1991). Instance-based learning algorithms. *Machine Learning* , 37-66.

Aljandal, W., Bahirwani, V., Caragea, D., & Hsu, W. H. (2009). Ontology-Aware Classification and Association Rule Mining for Interest and Link Prediction in Social Networks. *AAAI Spring Symposium (SSS'09) on Social Semantic Web.* Stanford University, California.

Aljandal, W., Hsu, W. H., & Xia, J. (2009). Predicting Protein–Protein Interactions using Numerical Associational Features. *the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB).* Nashville, TN: IEEE.

Aljandal, W., Hsu, W. H., Bahirwani, V., Caragea, D., & Weninger, T. (2008). Validation-Based Normalization and Selection of Interestingness Measures for Association Rules. *In Proceedings of the 18th International Conference on Artificial Neural Networks in Engineering (ANNIE-2008)* (pp. 517-524). St. Louis, MO: ASME PRESS.

Bahirwani, V., Aljandal, W., Hsu, W. H., & Caragea, D. (2008). Ontology Engineering and Feature Construction for Predicting Friendship Links in the Live Journal Social Network. *the 2nd SNA Workshop, 14th ACM SIGKDD Conference.* Las Vegas, NV: ACM SIGKDD.

Bay, S. D., & Pazzani, M. J. (1999). Detecting Change in Categorical Data: Mining Contrast Sets. . *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM.

Ben-Hur, A., & Noble, W. S. (2005). Kernel methods for predicting protein-protein interactions. *Bioinformatics* , 38-46.

Berberidis, C., Tzanis, G., & Vlahavas, I. (2005). Mining for Contiguous Frequent Itemsets in Transaction Databases. *In Proceedings of the IEEE 3rd International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IEEE.* Sofia, Bulgaria.

Besemann, C., Denton, A., Yekkirala, A., Hutchison, R., & Anderson, M. (2004). Differential Association Rule Mining for the Study of ProteinProtein Interaction Networks. *BIOKDD* (pp. 72-80). Seattle, WA, USA: ACM SIGKDD.

Bilgic, M., Namata, G. M., & Lise, G. (2007). Combining Collective Classification and Link Prediction. *Workshop on Mining Graphs and Complex Structures at the IEEE International Conference on Data Mining, (ICDM).* Omaha, NE.

BioXGEM-Lab. (2006). *DAPID: Domain Annotated Protein-protein Interaction Database*. Retrieved 11 2, 2008, from http://gemdock.life.nctu.edu.tw/dapid/

Blanchard, J., Guillet, F., Gras, R., & Briand, H. (2005). Using information-theoretic measures to assess association rule interestingness. *Proceeding of the 2005 international conference on datamining (ICDM'05)*, (pp. 66–73). Houston, TX.

Bock, J. R., & Gough, D. A. (2001). Predicting protein–protein interactions from primary structure. *Bioinformatics , 17* (5), 455-460.

Bodon, F. (2003). A fast apriori implementation . *IEEE ICDM Workshop on Frequent Itemset Mining Implementations.*

Breiman, L. (2001). Random Forests. *Machine Learning , 45* (1), 5-32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification of RegressionTrees* (1st edition ed.). Chapman & Hall/CRC.

Brijs, I., Swinnen, G., Vanhoof, K., & Wets, G. (1999). The use of association rules for product assortment decisions: a case study. *Proceedings of the Fifth International Conference on*, (pp. 254-260). San Diego (USA).

Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket analysis. *Proceeding of the 1997 ACM-SIGMOD international conference on management of data (SIGMOD'97)* (pp. 255–264). Tucson, AZ: ACM.

Chen, Y.-P. P. (2005). *Bioinformatics Technologies.* Springer.

Christos, B., George, T., & Ioannis, V. (2005). Mining for Contiguous Frequent Itemsets in Transaction Databases. *IEEE 3rd International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2005), IEEE.* Sofia, Bulgaria.

Cleverdon, C., Mills, J., & Keen, M. (1966). Factors determining the erformance of indexing systems. *Cranfield , 1-2.*

Cong, G., Tan, K.-L., Tung, A. K., & Pan, F. (2004). Mining Frequent Closed Patterns in Microarray Data. *Proceedings of the Fourth IEEE International Conference on Data Mining.* Washington, DC, USA: IEEE Computer Society .

Cook, S. (2006). Domain-Specific Modeling. *the Architecture Journal* .

Creighton, C., & Hanash, S. (2003). Mining gene expression databases for association rules. *Bioinformatics , 19* (1).

Deng, M., Sun, F., & Chen, T. (2003). Assessment of the Reliability of Protein-protein Interactions and Protein Function Prediction . *Pacific Symposium of Biocomputing (PSB2003}.*

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters* , 861-874.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (Fall 1996). From data mining to knowledge discovery in databases. *AI Magazine,* (pp. 37-54).

Freitas, A. A. (1999). On rule interestingness measures. *Knowledge-Based Systems* , *12* (5), 309-315.

Ganiz, M. C., Pottenger, W. M., & Yang, X. (2006). link analysis of higher-order path in supervised learning database. *In proceedings of the 4th SIAM Workshop on Link Analysis,Counterterrorism and Security.*

Geng, L., & Hamilton, H. (2006). Interestingness measures for data mining: A survey. *ACM Comput. Surv,* (pp. 38, 3).

George, T., Christos, B., & Ioannis, V. (2006). On the Discovery of Mutually Exclusive Items in a Market Basket Database. *In Proceedings of the 2nd ADBIS Workshop on Data Mining and Knowledge Discovery.* Thessaloniki, Greece.

Getoor, L., & Diehl, C. P. (2005). Link mining: a survey. *ACM SIGKDD Explorations Newsletter* (pp. 3 - 12). New York, NY, USA: ACM.

Grossman, J. (2007). *The Erdös Number Project*. Retrieved 10 10, 2008, from The Erdös Number Project: http://www.oakland.edu/enp/

Gruber, T. R. (1994). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human and Computer Studies*, *43*, pp. 907-928.

Haijian, S. (2007). *Best-first decision tree learning.* Hamilton, NZ.

Han, J., & Kamber, M. (2000). *Data Mining Concept and Techniques 1st ed.* Morgan Kaufmann Publisher.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *Proceedings of the 2000 ACM SIGMOD international conference on Management of data* (pp. 1-12). Dallas, TX: ACM.

Hao, Y., Zhu, X., Huang, M., & Li, M. (2004). Discovering patterns to extract protein--protein interactions from full texts. *Bioinformatics , 20* (18), 3604 - 3612.

He, F., Zhang, Y., Chen, H., Z. ,., & Peng, Y.-L. (2008). The Prediction of Protein-Protein Interaction Networks in Rice Blast Fungus. *BMC Genomics* , 9:519.

Hilderman, R., & Hamilton, H. (2001). *Knowledge discovery and measures of interest.* Boston : Kluwer Academic.

Hilderman, R., Carter, C., Hamilton, H. J., & Cercone, N. (1998). Mining Market Basket Data Using Share Measures and Characterized Itemsets. *Pacific-Asia Conference on Knowledge Discovery and Data Mining.*

Hoan, T., Satou, P. K., & Ho, T. B. (2004). Mining Yeast Transcriptional Regulatory Modules from Factor DNA-Binding Sites and Gene Expression Data. *Genome Inform Ser* , 287-295.

Holte, R. C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. *Machine Learning , 11* (1), 63-90.

Hsu, W. H., King, A. L., Paradesi, M. S., Weninger, T., & Pydimarri, T. (2006). Collaborative And Structural Recommendation Of Friends Using Weblog-Based Social Network Analysis. *AAAI '06: Proceedings of Computational Approaches to Analyzing Weblogs - AAAI.*

Hsu, W. H., Lancaster, J., Paradesi, M. S., & Weninger, T. (2007). Structural link analysis from user profiles and friends networks: A feature construction approach. *ICWSM-2007*, (pp. 75-80). Boulder, CO.

Hung, F.-H., & Chiu, H.-W. (2007). Protein-Protein Interaction Prediction based on Association Rules of Protein Functional Regions. *Proceedings of the Second*

International Conference on Innovative Computing, Informatio and Control (p. 359). IEEE Computer Society.

Jensen, D., & Goldberg, H. (1998). *AAAI Fall Symposium on AI and Link Analysis*. Retrieved 9 29, 2008, from http://kdl.cs.umass.edu/events/aila1998/

Jiang, X.-R., & Gruenwald, L. (2005). Microarray gene expression data association rules mining based on BSC-tree and FIS-tree. *Data & Knowledge Engineering , 53* (1).

Kanazawa-University. (2001). *Yeast Interacting Proteins Database*. Retrieved 11 1, 2008, from http://itolab.cb.k.u-tokyo.ac.jp/Y2H/

Kang, K., Lin, K., Zhou, C., & Guo, F. (2007). Domain-Specific Information Retrieval Based on Improved Language Model. *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, (pp. 374-378). Haikou.

Kotlyar, M., & Jurisica, I. (2006). Predicting protein-protein interactions by association mining. *Information Systems Frontiers* , 37-47.

Lee, Y.-K., Kim, W.-Y., Cai, Y., & Han, J. (2003). CoMine: efficient mining of correlated patterns. *Proceeding of the 2003 international conference on data mining (ICDM'03),*, (pp. 581–584). Melbourne, FL.

Lenca P., Vaillant B., Meyer P. & Lallich S. (2007) Association rule interestingness measures: experimental and theoretical studies, Quality Measures in Data Mining (eds. Guillet F. and Hamilton H. J.), Studies in Computational Intelligence, 43, Springer-Verlag Berlin Heidelberg, pp. 51-76.

Liben-Nowell, D., & Kleinberg, D. (2003). The link prediction problem for social networks. *Proceedings of the Twelfth International Conference on Information and Knowledge Management,*, (pp. 556-559).

Lin, C., Jiang, D., & Zhang, A. (2006). Prediction of Protein Function Using Common-Neighbors in Protein-Protein Interaction network. *Proceedings of the Sixth IEEE Symposium on BionInformatics and BioEngineering* . Arlington, VA : IEEE Computer Society.

Liu, B., Hsu, W., Chen, S., & Ma, Y. (2000). Analyzing the subjective interestingness of association rules. *(5)47--55* (15).

LIU, B., HSU, W., MUN, L.-F., & LEE, H.-Y. (1999). Finding interesting patterns using user expectations. *IEEE Trans. Knowl. Data Eng*, (pp. 11, 6, 817–832).

Martinez, R., Pasquier, N., & Pasquier, C. (2008). GenMiner: Mining non-redundant association rules from integrated gene expression data and annotations. *Bioinformatics* .

McIntosh, T., & Chawla, S. (2007). High Confidence Rule Mining for Microarray Analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* (pp. 611-623). Los Alamitos, CA, USA: IEEE Computer Society Press .

McIntosh-, T., & Chawla, S. (2005). On discovery of maximal confident rules without support pruning in microarray data. *Proceedings of the 5th international workshop on Bioinformatics* (pp. 37-45). New York, NY, USA: ACM.

Ming, T. (1993). Cost-sensitive learning of classification knowledge and its application in. *Machine Learning , 13* (1), 7-33.

Mitchell, T. M. (1997). *Machine Learning.* McGraw-Hill Science/Engineering/Math.

Omiecinski, E. (2003). Alternative interestmeasures formining associations. *IEEE Trans Knowl and data engineering* , 15:57–69.

Oyama- Takuya, K. K. (2002). Extraction of knowledge on protein-protein interaction by assciation rule descovery. *Bioinformatics , 18* (5), 705-714.

Oyama, T., Kitano, K., Satou, K., & Ito, T. (2000). Mining Association Rules Related to Protein-Protein Interactions. *Genome Informatics , 11*, 358–359.

Oyama, T., Yoshida, M., Satoshi, K., Kitano, K., Miura, F., Kawaguchi, N., et al. (2003). Automatic Extraction of Expression-Related Features Shared by a Given Group of Genes. *GENOME INFORMATICS SERIES* , 312-313.

Pan, F., Cong, G., Tung, A. K., Yang, J., & Zaki, M. J. (2003). Carpenter: finding closed patterns in long biological datasets. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* . Washington, D.C: ACM.

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. *Knowledge Discovery in Databases* , 229-248.

Popescul, A., & Ungar, L. H. (2003). Statistical relational learning for link prediction. *IJCAI Workshop on Learning Statistical Models.*

Qi, Y., Klein-seetharaman, J., & Bar-joseph, Z. (2007). A mixture of feature experts approach for protein-protein interaction prediction. *BMC Bioinformatics , 8(Suppl 10):S6.*

Quinlan, J. R. (1993). *C4.5: programs for machine learning.* San Francisco, CA: Morgan Kaufmann Publishers Inc.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning , 1* (1), 81-106.

Raz, T., & Yehuda, S. (Jan. 2006). On a confidence gain measure for association rule discovery and scoring. . *The VLDB Journal 15, 1* , 40-52.

Schmitz, C., Hotho, A., Jäschke, R., & Stumme, G. (2006). Mining association rules in folksonomies. *Data Science and Classification: Proc. of the 10th IFCS Conf. Studies in Classification, Data Analysis, and Knowledge Organization* (pp. 261-270). Berlin, Heidelberg,: Springer.

Schwikowski, B., Uetz, P., & Fields, S. (2000). A networks of protein-protein interaction in yeast . *Nature Biotechnology* , 1257–1261.

Sergey, B., Rajeev, M., Jeffrey, D. U., & Shalom, T. (1997). Dynamic itemset counting and implication rules for market basket data. *SIGMOD, 6*, pp. 255-264. New York.

Shanfeng, Z., Yasushi, O., Gozoh, T., & Hiroshi, M. (2005). Mining literature co-occurrence data using a probabilistic model. *IPSJ SIG Technical Reports,* (pp. 9-16). Japan.

Shekar, B., & Natarajan, R. (2004). A Transaction-based Neighborhood-driven Approach to Quantifying Interestingness of Association Rules. *In Proc. Fourth IEEE Int. Conf. on Data Mining.*

Steven, D. (1996). Knowledge-Guided Constructive Induction. *Ph.D. thesis.* University of Illinois.

Tan, P.-N., Kumar, V., & Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. *Proceeding of the ACM SIGKDD international conference on knowledge discovery in databases (KDD’02)* (pp. 32–41). Edmonton, Canada: ACM SIGKDD.

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. *UAI.*

Taskar, B., Wong, M.-f., Abbeel, P., & Koller, D. (2003). Link prediction in relational data. *Neural Information Processing Systems Conference (NIPS03).* Vancouver, Canada.

Thakkar, D., Ruiz, C., & Ryder, E. F. (2007). Hypothesis-Driven Specialization of Gene Expression Association Rules. *Proceedings of the 2007 IEEE*

International Conference on Bioinformatics and Biomedicine. Washington, DC, USA: IEEE Computer Society.

Turanalp, M. E., & Can, T. (2008). Discovering functional interaction patterns in protein-protein interaction networks. *BMC Bioinformatics* , 9:276.

Turney, P. D. (1995). Cost-sensitive classification: empirical evaluation of a hybrid genetic. *Journal of Artificial Intelligent Research 2 , 2*, 369-409.

Tuzhilin, A., & Silberschatz, A. (1996). What makes patterns interesting in knowledge discovery systems. *IEEE Trans. on Knowledge and Data Eng*, (pp. 8, 6,).

UCLA. (2008). *Database of Interacting Proteins*. Retrieved 11 2, 2008, from http://dip.doe-mbi.ucla.edu/

Van, R. C. (1979). *Information Retrieval.* Newton, MA: Butterworth-Heinemann.

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques.* San Francisco: Morgan Kaufmann.

INDEX