Measures and Adjustments of Pattern Frequency Distributions

Tongyuan Wang

A Thesis

In the Department

of

Computer Science and Software Engineering

Prepared in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

April 2010

Your file Votre référence
ISBN: 978-0-494-67321-8
Our file Notre référence
ISBN: 978-0-494-67321-8

ABSTRACT

Measures and Adjustments of Pattern Frequency Distributions

Tongyuan Wang
Concordia University, 2010

Frequent pattern mining over large databases is fundamental to many data mining applications, where pattern frequency distribution plays a central role. Various approaches have been proposed for pattern mining with respectable computational performance. However, the appropriate evaluation of the pattern frequentness and the refinement of the mining result set are somewhat ignored. This has created a set of problems in conventional mining approaches which are identified in this thesis. Most conventional mining approaches evaluate pattern frequentness with an ill formed "support" measure, and generate patterns with full enumeration mode which produces excessive number of patterns in an application. Consequently, the mining result sets exhibit among other issues those of overfitting and underfitting, probability anomaly and bias for generated against original observations. Even worse, these results are delivered to users without any refinement. Overcoming these drawbacks is challenging, since these problems are rather philosophical than computational and hence their resolution demands a well established theory to reform the mining foundations and to pursue graceful knowledge degeneration.

Based on the problems identified, this thesis first proposes a reformulation of the frequentness measure, which effectively resolves the probability anomaly and other related issues. To deal with the profound full enumeration mode, we first explore a set of properties governing raw pattern frequency distributions, such that a number of important mining parameters can be predetermined. Based on these explorations, an approach to adjust the raw pattern frequency distributions is established and its theoretical merits are justified. This refinement theory shows that unconditional pattern reduction is achievable before domain constraints are imposed. The thesis then presents a maximum likelihood pattern sampling model and strategies to realize the adjustment.

Findings presented in this thesis are based on known set theory, combinatorics, and probability theory, and they are theoretically fundamental and applicable to every item based or key words based pattern mining and the improvement of mining effectiveness. We expect that these findings would pave a way to replace the full enumeration pattern generation with selective generation mode, which would then radically change the state of the art of pattern mining.

Acknowledgement

Many thanks to Dr. Bipin C. Desai for his dedicated supervision during the doctoral program and this thesis, his encouragement, time and liberal thinking.

Tongyuan Wang

April 2010

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Frequent pattern mining over large databases is fundamental to other mining especially the association rule mining, correlation mining and causation mining. Starting from the work by Agrawal [1, 5], extensive research to date has been reported on frequent pattern mining. Most of the research is focused on the computation efficiencies, including scalability and memory optimization [2]. Efficiency is certainly important in dealing with large datasets, but the most important yet least studied issue is how to refine the mining result set to improve the mining reliability and hence usability. We notice this importance in the mining domain in our earlier attempted causation mining over traditional Chinese medicine (TCM); this issue formed the starting point of our current research.

Research background

Mining causation relations from TCM is not only industrially important but also technically significant because of the complications of the TCM data sources, which calls for effective mining solution. Among other difficulties, the typical data issues in TCM mining include:

1. Synonymous entries – different literals used in the TCM data source referring to the same thing or similar thing.
2. Multiple valued states (MVS) – a data cell may contain a set of values of an attribute, which violates the "first order normal form" (1NF) requirement of the relational

database theory; e.g., a medicine may have different origins and hence characteristics.

3. Complex domains: the values of an attribute may not be exclusive to one another, while conventional database deals with simple domains. A simple domain X means:

   $\cup_i x_i = X$, and $x_j \cap x_k = \emptyset$, where x_i and x_j are values of X, $j \neq k$, and $\emptyset$ is the empty set.

   The attribute "origin" is an example of complex domain that, a herb may grow in province A or B, while another herb may grow in Mount M that stretches over but only partially covers A and B.

4. Layered values: This is related to the above issue but more concerned with the scope of the denotations or connotation of the values. For instance, the scope of the "origin": some herbs might grow throughout a country, but some others might grow in a small area of a province only.
5. Data reliability: Due to the long historical evolution of TCM and due to the imprecise records, data reliability issue is the primary concern for reliable mining results, even though we have managed to ensure the database to be filled with as accurate as possible data drawn from the data source [52].

The above five issues mostly emerge together among most of the attributes of the TCM database we have constructed. We have used the "origin" attribute as a commonly understandable example of the data complications for non-medical readers. For other more important attributes of TCM, such as the "efficacy", the difficulties engendered by the above five issues cannot be overlooked. In short, data complication is the first challenge for us to propose an effective mining approach. However, reported data

mining approaches as summarized in [2] are unable to deal with the above introduced data complications together, though some of them have concerned with individual data characteristics.

Taking all of these issues into consideration, we have designed and implemented a fuzzy "Hyper Knowledge Discovery System" (HKDS) to deal with TCM data mining. The first set of functionalities of the HKDS is on fuzzy information retrieval, pattern mining and association rule mining, as summarized in [52].

What we want to point out here is that, after many efforts in dealing with the data complications as described above, we found that there are even more profound issues preventing us from pursuing convincing mining results over many datasets including the TCM data. Such issues include overfitting and underfitting, probability anomaly, bias for generated against original observations, etc. as summarized in Chapter 2. These issues are generally incurred but not well addressed by conventional pattern mining approaches. It is these issues that led us from our original efforts to mine TCM data to concentrating on the current research topic towards refining mining concepts and theories. This work has culminated in the following contributions:

Our contributions

Our main contribution is in investigation and reformulation of some concepts and theories underlying the conventional pattern mining approaches, such that the state of the art of pattern mining would be radically changed. The contribution can be described in the following aspects:

a) The investigation and identification of the fundamental drawbacks embodied in conventional mining approaches. This is a contribution because detecting a problem is of first importance in a research, then is its solution, and because the problems revealed in this thesis are fundamental to an effective pattern mining.

b) The reformulation of frequentness measure. The change is simple but effective especially in the resolution of probability anomaly and overfitting issues.

c) The explorations of a set of laws governing raw pattern frequency distributions, and hence lays a foundation for frequency adjustments. These laws can be used in data property analysis and can serve as checkpoints to validate mining algorithms.

d) A theory on the adjustment of the raw pattern frequency distributions, which forms an unconditional pattern refinement framework and promises a development of selective pattern generation mode.

e) An optimization model and the related strategies to adjust individual pattern's frequentness based on the above adjustment framework.

These contributions are embodied in the following context.

Structure of this thesis

In the second chapter, we present a brief literature review on frequent pattern mining and the profound issues and drawbacks that conventional mining approaches unintentionally embrace, including overfitting and underfitting, probability anomaly, bias for generated against original observations, and bias for shorter against longer patterns. Chapter 3

proposes a new measure to gauge the pattern frequentness and radically resolve the probability anomaly issue. Chapter 4 analyzes the properties underlying the raw pattern frequency distributions based on full enumeration pattern generation regime. Chapter 5 then points out the need and presents a theory to reduce the number of excessively generated patterns and adjust their frequency distributions in a collective mode. The adjustment theory is established on a set of mathematical properties, such that the merits of the full enumeration mode could be maintained while its drawbacks are handled effectively. In other words, dimension reduction and noise diminishment are naturally embodied in the adjustment functions. Chapter 6 presents empirical verifications of the theories and properties presented in the previous two chapters. Finally, Chapter 7 proposes a maximum likelihood model to optimize the pattern sampling and realize the proposed adjustment theory; this is followed by our conclusion.

Chapter 2. Related work and open issues

Frequent pattern mining is a broad area. According to the data types, it can be divided into qualitative or quantitative mining [53, 54]. This thesis focuses on the former one concerning non-continuous data sources. However, the principles discussed herein could be applicable to the latter one as well. According to the application domains, pattern mining has developed from the early days' market-basket itemset mining to today's temporal pattern mining, spatial pattern mining, sequential pattern mining, medical data mining, genomic pattern mining, and so on. Nevertheless, the fundamental problem is the same in all of these tasks. Below is a running example to illustrate the basic problem.

2.1 The problem and terminology

Table 1 represents a database DBo of u rows and two columns. Column TID represents the key attribute and VID represents an application domain Ω of n distinct elements. Each row is a tuple, where T_i (i = 1, 2, ..., u) is a tuple ID; and each cell of column VID contains a value V (or a set of values) of that domain. For example, in a market-basket problem, a TID could represent a transaction ID, and a value of VID, V_i (i = 1, 2, ..., n), is an "item" from the domain Ω of merchandise. Particularly, a combination of k distinct Vs is termed as a pattern $Z_k = (V_i V_j ... V_s)$ of length k, or

Table 1. A Database (DBo)

TID	VID
T_1	V_1, V_4, V_7
T_2	V_2, V_4, V_7, V_8
T_3	V_2, V_6
T_4	V_1, V_6, V_8
T_5	$V_1, V_2, V_3, V_4, V_7, V_8$
T_6	V_5
T_7	V_4, V_7
T_8	V_5
T_9	V_1, V_2
T_{10}	V_1, V_2, V_3, V_8

k-itemset in market-basket problem [1, 5]. A formation of such a pattern is termed a pattern *generation*. By convention, the number of occurrences or absolute frequency *S* of a pattern Z is termed as its (absolute) support S_z over the database. The relative support is a ratio s_z [2]:

$$s_z = s(Z) = count(Z) / |DBo| = S(Z) / u = S_z / u, \qquad (2\text{-}1)$$

where u = |DBo| is the total number of tuples, i.e., the cardinality u of DBo. The multiple notations of the above support measure are used for convenience in the later part of this thesis.

Obviously, (2-1) comes from classical frequency based probability concept, and s_z should be taken as the first link between probability theory and pattern mining. In statistics terminology, the dataset DBo is a sample of the real world application at hand. The cardinality u of DBo is the sample size; and a record (tuple) is a realized *event* of the sampling [16], and hence a subset of Ω. In data mining language, we term each original tuple (event) an *original pattern*, or an *original observation*. A TID can be taken as a sample label or trial ID, and the column VID refers to the set of *events* [6]. Based on these notations, the fundamental data mining problem can be stated as follows:

Problem 2-1 (conventional problem 2-1): Given a dataset DBo as shown in Table 1 involving the universe Ω of n distinct elements of domain VID, output all patterns of the elements in any length, such that the s_z of a pattern Z satisfies $s_z \geq s_{min}$, where s_{min} is a user predefined minimum support; such satisfactory patterns are termed as qualified patterns.

Problem 2-1 itself is not too difficult to comprehend. A main issue for most of the research is the computation complexity, since there are up to the power set (2^n) of possible patterns over the n-element domain Ω. Note that, in this thesis we do not take the empty set ($\emptyset$) as a pattern, and $F(\emptyset) = 0$ in case its frequency needs to be considered. Then the largest number of possible patterns is $2^n - 1$. The power set complexity demands not only a great amount of computation time but also large memory space to store the candidate patterns and other information. The next section briefly summarizes how researchers have attempted to handle this problem.

2.2 Related work

There have been many proposals for pattern mining, and they are quite often embodied in association rule mining. Examples of these proposals include the early days *Apriori* algorithm [2, 5], the FP-growth method [7], their variations and extensions, and many other approaches.

To understand the pursuance of these proposals, let us look at the primitive mining approach over Table 1, where eight elements are involved, which means there will be $2^8 - 1 = 255$ possible combinations (patterns). The primitive approach is then enumerating each of the possible patterns and counting how many tuples support that pattern from Table 1. This would be the best understanding of the origin of the notation "support". For instance, we can get $S(V_1) = 4$, $S(V_1V_2) = 3$, ..., $S(V_1V_5) = 0$, and so on. This approach is simple, complete, but inefficient in a couple of aspects. Firstly, notice the dataset involved is normally too large to be fully loaded into the main memory to carry out the required enumeration-counting processes. This then demands multiple IOs,

which is expensive. Secondly, each turn of pattern enumeration – database access – querying and counting is again costly. Thirdly, there could be zero occurrences of many combinations, and enumerating them entails a (computation) resource waste. For instance, in Table 1, more than 70% of the 255 combinations are of zero frequency (refer to Table 2 in next section). Fourthly, similar waste can occur when enumerating those combinations whose occurrences are under s_{min}. Lastly but not least, the enumeration demands large memory space for storing these patterns.

It is because of these inefficiencies of the primitive approach, most mining proposals concentrate on efficient generation of patterns and the determination of their frequencies.

An easiest way to eliminate the third waste listed above is by replacing the full pattern enumeration from the whole element space Ω by full enumeration from each data tuple. This is the origin of the "pattern generation". This approach, however, introduces a side effect and complication that, for each tuple, one needs to check whether a combination of the elements of that tuple has already been enumerated or not. As can be imagined, such checking is expensive. Accordingly, strategies for pattern search are introduced, including "breadth first" and "depth first" search methods. In breadth first search the patterns are generated and examined from each record of $(T_i: \{V_j\})$ in horizontal data format [2]. In depth first search, the original dataset DBo are transformed into a "vertical" dataset $(V_j: \{T_k\})$, then patterns and their related frequencies are determined [9, 10].

Many proposals have been made to reduce the fourth waste. For instance, the *Apriori* algorithm [5] features pruning infrequent itemset as early as possible to achieve

computation efficiency. The pruning strategy is based on the *intuition* that any super pattern of an infrequent pattern cannot be frequent. This intuition is proved (as can be seen in Chapter 4) to be true against conventional mining conception and now referred to as "anti-monotonicity" or "downward closure" property [2]. The *Apriori* is a level-wise mining approach. The algorithm starts from k = 1, and for each loop k, enumerate (generate) all the patterns of length k (candidate patterns) to determine their frequencies; prune away those Z_k with $s(Z_k) < s_{min}$. The retained $\{Z_k\}$ are the qualified patterns of length k, and use the retained $\{Z_k\}$ as the seeds to generate patterns of length k + 1, since the super patterns of those pruned patterns could not be frequent based on the downward closure property. Repeat these operations until no more patterns could be generated and counted [5].

The *Apriori* approach, however, requires large memory space to generate and store a bundle of candidate patterns at each level of the pattern length, and it needs to repeatedly scan the database to obtain the candidate pattern frequencies. As a result, a number of variations and extensions of the *Apriori* approach have been proposed to improve the shortcomings. For instance, the "incremental mining" [43], the "dynamical itemset counting" [44], the "parallel and distributed mining" [45], the "hash-based" [46] and the "partitioning" [47] algorithms, are a few example proposals, which we do not summarize in depth here to save space.

Unlike the *Apriori* approach, the frequent pattern growth (FP-growth) approach [7] tries to avoid candidate pattern generation and hence reduce memory space requirement and IO cost. We do not present a complete explanation of this approach here due to space

required to adequately cover the terminologies and the techniques used in the paper. In short, this approach involves two database accesses and two types of trees to build. The first database access counts the frequencies of all the individual elements that are then listed in a descending order. The second database access is to build a FP-tree, starting from the most frequent elements obtained in the first step. The FP-tree is a prefix tree and acts as a compression of the original database, such that a data tuple is embodied in a branch of the tree. Then pattern mining from the original database is converted to mining from the FP-tree. The mining starts from length-1 patterns as initial suffix of other patterns, by constructing "conditional trees" from the FP-tree. A conditional tree represents a sub database, called "conditional base", consisting of the prefix paths of the patterns that are co-occurring with the suffix.

The advantage of the FP tree is that, once the tree is built, it can be repeatedly used by later mining activities without further database access and hence improve mining efficiency. However, the cost to build the tree is substantial, and more importantly, the FP tree is not guaranteed to fit completely into the main memory, and hence the mining efficiency will be compromised. To overcome these drawbacks, many extensions and variations of this approach have also been reported, for example, the "hyper structure mining" approach [48], the "bottom up and top down" tree building approach [49, 50], the array based data structure to implement the prefix tree [51], and the like.

There is also a proposal to avoid multiple IOs and to reduce the candidate pattern generations by statistical estimation over database scanning [8]. The basic idea is to randomly sample the original dataset and use it to mine the patterns against a lower s_{min};

the resulting pattern set is the final mining result set, which can be verified by the rest of the dataset. The paper claims that, in normal case, one pass of sampling and hence just one set of IOs could produce the whole result set, and if not, the missed patterns can be found in at most another pass of sampling. Nevertheless, as the paper admitted, this approach could not guarantee to output the qualified patterns precisely and completely.

The above are just a few examples that researchers have attempted for mining patterns efficiently. We would not summarize many other proposals here because of space limitation; furthermore, efficiency is not the main focus of this thesis. Rather, we are more concerned with the refinement of the mining results and on how to reduce the number of meaningless patterns that conventional mining approaches produce. In this regard, we have not found many articles on mining refinement to reduce the meaningless patterns, although a lot of research has been done on how to present the mining result set in a reduced form, such as follows.

The "constrained" pattern mining [27] reduces the mining result set size by user constraints. For instance, a user may want to mine patterns with V_1 and V_2 only, or with other constraint(s), from Table 1. A number of categories of constraints and the related mining approaches have been studied. One of them is the "monotonic" (or anti-monotonic) constraint, which features, if C is a constraint, then any of its superset $S \supseteq C$ is also a constraint. This property can be seen as a mirror to the "downward closure" property used in the *Apriori* approach, and it then can be used in a mining algorithm in a similar way as the *Apriori* approach to prune patterns that do not meet these constraints [23]. Similarly, others, such as the "succinct" constraint [23], the "convertible"

constraint [26], and the "block" constraint [11], have been studied. We do not discuss these constraints further here, while the purpose of these studies is the same as that for "monotonic" constraint, namely, how to adopt the properties of these constraints into the related mining strategies and algorithms.

Another school of reduction approaches is the "concise" ("condensed" [4], or "compressed" [2]) representation of the patterns, which means they use a small subset of the frequent patterns to represent the whole mining result set. For example, the "free sets" [29] or "generators" [30] are concise sets to represent the whole result set in an application, where a generator can be understood as a set of elements G, such that there is no $G' \subset G$ with $support(G') = support(G)$. The generator possesses the anti-monotonic property as well, that is, if G is not a generator, then $G' \subset G$ is neither. From here, we see the anti-monotonic property has been widely used in pattern mining. Similarly, other concise sets, e.g., "disjunction-free" [31], or "non-derivable" sets [28], have been proposed, while the "closed" [13], and the "maximal" [14] (or "hybrid clique" [15]) approaches have attracted more attentions. A pattern is closed if none of its proper super-pattern takes the same frequency [2, 13]. A pattern is maximal if none of its proper super-pattern is frequent against a s_{min} [2, 14]. The "closed set" representation is a lossless compression of the results set, in the sense that all of the patterns and their supports can be derived from the closed set; while the "maximal" expression is a lossy compression [2].

Similar to a lossy compression, there are approximation approaches to represent the mining results, for instance, the "top-k most frequent closed" [32], and the "pattern profile" [33] approaches.

The former proposes to mine desired number k of top frequent closed itemsets of length no less than a predefined min_l. The related mining algorithm TFP is developed from the FP growth approach. This approach does not take the s_{min} threshold into consideration, since it is not easy for a user to define a s_{min}. This is a notable point, but on the other hand, this approach [32] faces the same problem, namely, how to determine the number k and min_l, and by whom? The paper [32] did not touch this problem, and we could only surmise, similar to s_{min}, that k and min_l can only be defined *ad hoc* in an application.

The latter approach (pattern profile [33]) is based on an observation that, rather than efficiency, how to interpret the result patterns in an application is the main issue of pattern mining. The paper [33] then proposes a statistical model to summarize the large number of patterns with a small set of k representative patterns. The paper presents algorithms to seek an optimized k such that the frequent patterns can be recovered from the representative set including their supports with a small error. The authors claim that the summarization solves the interpretability problem, but it is not clear from the paper as to how. Similar to other concise or approximation approaches, the pattern profile summarization is still a compressed representation approach by using a small set of patterns to recover (or represent) but not to (semantically) interpret the whole pattern set.

The above approaches demonstrate the efforts that researchers have made in dealing with the big number of patterns in an application, yet from the above introductions, it is easy

to see that the representation approaches do not touch on the issues of refinement and hence on how to improve the meaningfulness of the mining result set. At least, these approaches did not claim, for instance, whether patterns within the concise set would all be meaningful. Additionally, these approaches did not address other more fundamental issues as presented in the next section.

2.3 The open fundamental issues

In this section, we examine some fundamental mining problems from shallow phenomena to deeper theoretic issues, so that we could identify the problems clearly and develop our solution rationally.

2.3.1 Meaningless but overwhelming number of resulted patterns

The first goal of pattern mining should be on the meaningfulness of the mined results. However, this is a far reaching issue. A few years ago, people noticed that, in an application especially if s_{min} is low, thousands or millions of frequent patterns may be produced from a fairly large database [12], but many of them are meaningless, some of them being even "counter intuitive" [55]. These problems essentially remain unchanged today, though notably many thoughtful resolutions have been proposed, for instance, the "concise representation" pattern mining approaches summarized above. These approaches try to use a small set of patterns to represent the whole result pattern set, but no article has claimed that its approach reduces the number of meaningless patterns. Furthermore, no article has claimed the related concise or representative sets contain none or fewer meaningless patterns.

2.3.2 Overfitting issues

The above mentioned issue of huge number but meaningless frequent patterns is theoretically an "overfitting" problem. Overfitting here simply means spurious patterns falsely taken to be significant ones. Conversely, a true frequent pattern but falsely taken to be infrequent is termed as underfitting. In our study, the overfitting problem is dominant and widely incurred in previously proposed pattern mining approaches. The overfitting or underfitting problem is important since it determines the reliability of the mined results.

Reliability is a widely used and discussed criterion in data mining community. Article [56] is an example wherein the problem of enhancing data mining reliability is addressed. However, formal and concise definition of data mining reliability is not readily available. In general, data mining reliability is determined by the effectiveness of a mining approach, in addition to other factors, such as the data quality, data size, and data complexity. Deriving reliability matrix is still an open issue, but criteria used in classic statistical tests can be availed of, including the stability of the mined results against data size change or data source change, and more importantly the degree of closeness of the mined results to the real values or structures embodied in the real world. For an unknown world, the said closeness can only depend on the soundness of the mining technology. The minimum requirement of the soundness should be the compliance of mining results with commonsense; a higher requirement should be the conformability of the mining principles with other related established theories.

Overfitting or underfitting certainly deviates a mining result set from the real structure, and causes instability of the mining results as we would soon see. Unfortunately, we have not found, from data mining literature, a substantial and generally accepted criterion to measure the degree of overfitting embodied in a mining result set. The absence of such measurement on one hand reflects the difficulty of the overfitting problem, and on the other hand reflects the lack of known "right" mining solution to compare with. In other words, researchers have not found a proper answer to the cause and mechanism of the overfitting problem. The following subsection reveals some of the sources of the overfitting issue.

2.3.3 Probability anomaly

As we know, pattern mining is a probability and statistics based technology, and some people even take it to belong to the area of statistics [18]. However, under our investigation, the radical problem of pattern mining – the overfitting issue – is exactly rooted from its improperly designated probabilistic criterion, the use of "support" s_z as defined in (2-1). It is obvious to see in today's data mining literature that s_z is indeed used to mean the frequentness (relative frequency) of a pattern, whether it is called "support" or not. Note that the "frequentness" is a synonym to "probability" in frequency based probability theory. Then, the accumulative frequentness of all the patterns in a question must be equal to 1. However, the use of s_z directly leads to a common problem in pattern mining that the accumulated probability (the sum of s_z) of the mining results is much larger than 1, which seriously violates the fundamental probability concept. We term this issue a "probability anomaly".

Table 2. "Statistics" from the data of Table 1 and formula (2-1)

K	Example patterns Z_k	# of Z_k	$\sum S_z$	$\sum s_z$	# freq. Z_k (s_{min} =20%)	$\sum S_z$	$\sum s_z$	#' freq. Z_k (s_{min} =20%)
A	T	B	D	E	F	D'	E'	F'
1	V_1, V_2, ..., V_8	8	24	2.56	7	28	2.8	8
2	V_1V_2, ..., V_7V_8	19	30	3.33	10	36	3.6	13
3	$V_1V_2V_3$, ..., $V_4V_7V_8$	21	26	2.9	5	30	3.0	9
4	$V_1V_2V_3V_4$, ..., $V_3V_4V_7V_8$	15	16	1.8	1	17	1.7	2
5	$V_1V_2V_3V_4V_7$..., $V_2V_3V_4V_7V_8$	6	6	0.67	0	6	0.6	0
6	$V_1V_2V_3V_4V_7V_8$	1	1	0.11	0	1	0.1	0
$\sum$		69	103	11.4	23	118	11.8	32

Note, the last 3 columns are resulted from the whole Table 1, columns B, D, E and F are from its first 9 tuples.

Example 2-1: A concrete example is given in Table 2 based on Table 1, where column D, E, and F represent statistics derived from the first 9 tuples of Table 1, while column D', E', and F' show corresponding results from the whole 10 tuples. Column B is the subtotal of the patterns of same length; column D is the accumulated frequency (subtotal of occurrences) of the patterns of same length; column E is the accumulated relative supports and hence accumulated probability of patterns of same length, from which we see that the grand accumulated probability $\sum s_z = 11.4 >> 1$. Column F is the number of frequent patterns of same length with a not-low threshold $s_{min} = 20\%$, in total of 23. In a common sense, it is difficult to believe so many frequent patterns can be derived from such a small dataset (of 9 tuples). Even worse, if a new record $T_{10} = \{V_1, V_2, V_3, V_8\}$ is added to Table 1, then as shown in the last column of Table 2, the total number of frequent patterns increases from 23 to 32, for a 40% increase at $s_{min} = 20\%$. This

illustrates that the result set is very unstable. The large number of frequent patterns and the instability of the result set are the typical symptoms of overfitting. The cause of the probability anomaly is the definition of s_z that legalizes overfitiing in current data mining methods, since s_z is generally used in the proposed pattern mining approaches to date. In this sense, we can safely conclude that overfitting is naturally presented in available pattern mining approaches. Even worse, the degree of overfitting and the probability anomaly are increasing exponentially to the typical tuple size (length) of the dataset in question. This is because, by combinatorics: a tuple of x items will generate 2^x (to be more precise, $2^x - 1$) combinations (patterns), and hence one such tuple added to the database will increase the accumulative frequency w by 2^x.

2.3.4 A further insight into the summation issue of the supports

It might be argued that the "supports" s_z should not be summed together as described in previous subsection. A typical argument is that, for instance, pattern A and B may not be disjoint, and hence, s(A) and s(B) are not directly additive at all. We will discuss this joint-disjoint issue in the next chapter further. Here we can simply answer that, if s(A) and s(B) are not directly additive, then it means they are not directly comparable either. In this sense, s_z is disqualified to properly compare the frequentness of different patterns again.

The reason we consider all s_z to be additive is based on the fact that conventional mining approaches generate all the patterns from every data tuple based on uniform distribution assumption. This is analogous to the classical event based probability theory, where all events can be drawn from a universe based on uniform distribution assumption, and the

probabilities of all the drawn events are additive and accumulated to 1. Secondly, in real applications, we need the s_z summations. For instance in the running example, suppose there are two milk-based products: yogurt represented by V_1, and bottled milk by V_2. Then a question could naturally be asked: what is the total percentage (total support) of the milk-based product patterns? If those supports are not additive, then what should the solution be? On the other hand, if they are additive, as can be quickly seen from Table 1, the sum of their supports is certainly larger than 1. This is another dilemma due to the use of conventional "support".

Another typical argument can be put like this: It is not wrong to use either s_z or s'_z, but just a viewpoint difference, since s_z represents the probability that a randomly selected data tuple contains the pattern Z, while s'_z measures the probability of Z that is selected randomly from a randomly selected tuple of the DBo (all are based on random selection and uniform probability distribution assumption over the data tuples). We notice that, the basic task of pattern mining is the comparison of the frequentness (probabilities) of different patterns, but not the probability of the data tuple (transaction) itself. At the same time, even if we wanted to accept the argument, it still does not avoid the problem of probability anomaly. Take the first data tuple $T_1 = \{V_1, V_4, V_7\}$ of Table 1 as an example again, and notice that, from Table 1, $s(V_1) = 5/10 = 0.5$, $s(V_4) = 4/10 = 0.4$, and $s(V_7) = 0.4$, then it is easy to see their summation is already larger than 1, let alone other generated patterns' supports!

The answers to the above two arguments will become much clearer in the next chapter.

2.3.5 Other drawbacks of using s_z

In addition to being the source of probability anomaly, another related issue of the use of s_z is the inability to compare the frequentness of the mined results of similar mining tasks against different data sources. This is because, as stated above, the frequentness of a pattern generated from a database of larger tuple size in general would be much larger than that from a database of smaller tuple size, if the two databases have the same number of tuples. For instance, in a market-basket mining problem, the frequentness (support) of most of the patterns generated from a database of a supermarket could be very likely much larger than that from a database of a grocery store, if the supermarket and the grocery store sell the same set of items, and if the numbers of records of the two databases are the same. This is based on the observation that normally the number of items included in a transaction in a supermarket would be much larger than that in a grocery store.

Another big problem to be addressed is the objectiveness in determining the threshold, s_{min}, to mean if a pattern is frequent or not. The value of s_{min} is assumed to be set up by the user in most of studies, and there has been no formal proposal to establish the threshold s_{min}. This assumption is somewhat absurd. As [32] has noticed, it is hard for a user to decide the s_{min}, but the problem is much beyond that. From a more industrial practice point of view, it is the miner, not the user, to tell at what grade based on what standard a mined material is rich of something. Secondly, if the user takes the role and set an arbitrary s_{min}, it could results in an anomalous situation. For instance, with the same dataset, a pattern could be frequent for a user with a low s_{min}, but infrequent for

another user with a high s_{min}. This is certainly not objective or theoretically sound for serious data mining. There is a need of generally accepted indicator to be the frequentness threshold, or the s_{min}. A user defined s_{min} can only be taken as the threshold the user wants to look at from the mining result.

2.3.6 The absolute support S_z is no better a choice than the relative s_z

The question that remains is whether the absolute S_z is a better measure of the pattern frequentness. The answer is no, even though S_z has been used as the "support" measure in more than few research papers, e.g., [24 and 42]. Firstly, s_z is obtained from S_z, thus issues of s_z are applicable to S_z but harder to be identified since S_z does not present the "symptom" of probability anomaly. Instead, the symptom could easily be interpreted as a result of the use of too big dataset, since too many patterns (hence overfitting) could pass a fixed S_z when the data size becomes large. Such interpretation is counter to common sense that, with bigger dataset, more realistic mining results should be obtained. Consequently, there is an issue of what and how an absolute S_{min} could be set up to determine if a pattern is frequent or not; and, should the absolute S_{min} be changed if the data size changes or if the dataset changes? If yes, how? These are just a few questions among others to be answered formally.

Finally, either the absolute or the relative support is stiff, in the sense that S_z or s_z of a pattern Z is unable to reflect any change of other patterns' frequencies or their accumulative frequency. This is then another big problem to use either S_z or s_z to compare and to reflect frequentness of different patterns.

In summary, the use of s_z is a main source of the problems listed from 1.3.1 to 1.3.3. There is another reason – the full enumeration pattern generation methodology fundamentally used in conventional mining approaches that reinforces these problems. At the same time, this generation mode causes other problems too. Following subsections discuss the drawbacks induced by this mode.

2.3.7 Full enumeration mode and overfitting and underfitting

As its term implies, the full enumeration approach generates every possible combination including unrealistic patterns, resulting in an excessive number of patterns. This results in the previously mentioned exponential increase of accumulative frequency against typical data tuple size. As has stated before, a tuple of length x added to the database will increase the accumulative frequency w by 2^x; at the same time, a number of infrequent patterns and false patterns could be promoted to frequent ones. This is how the number of frequent patterns can increase non-linearly with data size increasing, and how "overfitting" problem could occur even without the use of conventional s_z. On the other hand, since the accumulative pattern frequency w increases exponentially to the size of every added data tuple under the full enumeration regime, this inflated w then causes true frequent patterns to become less frequent (this can be seen more clearly by measure (3-8a) in next Chapter), giving rise to underfitting.

Note that, most of the mining approaches including the concise representations, constrained mining, and the use of pattern pruning strategies, work over the full enumeration foundation, and hence they are generally prone to both overerfitting and underfitting problems.

2.3.8 The bias for generated patterns against the original ones

The full enumeration approach is unable to weight the frequentness of the original patterns and the generated ones differently, since no measure has been reported to differentiate the original from the generated patterns. For instance, the length-one patterns V_1 and V_5 in DBo of Table 1. In conventional pattern mining approaches, V_1 is taken to be of absolute support 5 while V_5 is of 2, meaning V_1 is more frequent than V_5. However, V_5 is an originally observed pattern and randomly sampled twice, but V_1 is just a "possible" pattern "generated" from longer patterns. This raises a question: could we really take the generated combination to be more likely a pattern than the observed one? This is a simple explanation on how the full enumeration approach is biased towards generated patterns.

2.3.9 The bias for shorter patterns

In conventional mining approaches due to the "downward closure" property [2], the longer (original) patterns are potentially less frequent than their sub-patterns and hence more likely to be excluded from the mining result set and only their sub patterns are kept against a given s_{min}. This is a drawback since it is unable to properly maintain the common observation that, only when sufficient necessary elements arise simultaneously can certain event take place. Take the pattern V_1 as an example in Table 1; it is evaluated to be frequent at $s_{min} = 20\%$, but a question is, could it really appear frequently without other elements such as V_2 or V_3? In real world, it is common that compounds (patterns) are more frequently seen than single elements. For instance in chemistry, pure copper (Cu) is much less frequently found than its compounds, e.g., copper oxide (CuO) , but in

conventional mining approaches we can only conclude that individual elements are more frequent than their combinations (patterns). This illustrates the previous problem again: the frequentness determination in pattern mining is not properly designed. At the same time, the "downward closure" property needs to be re-examined, which as can be seen later, is only valid in the conventional mining theory and the full enumeration pattern generation regime.

2.3.10 The mixture of pattern mining and element mining

This issue is related to the above problem, and it is common in conventional mining approaches that, individual elements are taken to be patterns and they are the most frequent ones compared with their super patterns (of length > 1). In some applications, such as spatial or sequential pattern mining, an individual element may form a pattern under certain structural or ordering (temporal) constraints, but in a pure frequentness based pattern mining, individual elements may not be very suitably termed as patterns but just complicate mining.

Theoretically, a single element could not be excluded as a pattern. However, if and only if such an element behaves independently, could we consider it a pattern. Otherwise pattern mining would be equal to (or at least, mixed with) "element mining". Similarly, a shorter combination (termed as "sub" patterns in literature) generated from longer pattern(s) (known as "super" pattern(s)) might be a true pattern, or might be a component (but not a pattern) of the longer pattern, depending on their behavior.

2.3.11 The ultimate question: What is a pattern and what is pattern mining?

There are myriads of papers about data mining in general, but we have not found a formal definition of the very basic object – the pattern – for the mining. There have been some definitions of pattern mining, e.g., by [13], but they are more on the mining computation issue, and the "patterns" are taken as "given". What we can conclude from the related literature about pattern is that:

A pattern is a combination whose frequentness is no less than a threshold s_{min}.

However, the above is a posterior assertion after the fact; secondly the above "definition" conflicts with a widely used term "infrequent pattern", whose frequentness is low ($< s_{min}$) but still considered as a pattern.

This might be the most critical problem – with the "pattern" being not well defined, a user then could be presented with whatever is mined as a pattern, ending up with a huge number of patterns – this in fact is the starting point of our present discussion as stated in subsection 2.3.1. Similarly, there has been no formal definition of the meaningfulness or meaningless of a pattern in the literature. One reason for this could be due to its dependence on the domain of application.

Neither are we ready to provide a formal definition of pattern and its meaningfulness, which we would have presented in the beginning of this section, since there is no other reference except the frequentness to define a pattern. Nevertheless, for a more formal discussion, we have the following remarks without referring to a specific domain:

Remark 2-1: A pattern is a configuration of the same elements significantly appearing in a dataset.

Remark 2-2: If a pattern is generated but should not have been under certain configuration rules, it is deemed to be "meaningless" or "redundant".

Note that, configuration implies semantics. A pattern is firstly a combination of some element(s), but not every arbitrary combination of elements could be a meaningful configuration and hence a pattern. The above remarks can be easily extended to more specific pattern mining problems. For instance, a sequential pattern is a significant presence of the same elements in certain ordering, wherein the ordering characterizes the configuration and semantics of the patterns; in spatial pattern mining, the configuration of a pattern is its spatial structure that reflects its semantics. For pure frequentness based pattern mining, focused on in this thesis, the semantics of frequent patterns can be generally described as follows: length-1 patterns demonstrate the individuality or strength of independency of individual elements behaving in a question; length-2 patterns manifest the ability of coexistence or partnership between two concerned elements; similarly, length-3 patterns exhibit this ability among three related elements, and so on. Although we could not generally describe the configuration of a pattern in pure frequentness based pattern mining, in many applications the configuration can be easily identified. These can be demonstrated more clearly in Example 2-2 and 2-3 below.

The second note is that "meaningfulness" might be taken as a synonym of "interestingness" used in association rules mining [40, 42], since the adaptation of interestingness also means to reduce the number of patterns, and the objective measure of interestingness is mainly on the pattern frequentness, or the significant presence of the patterns. In our understanding, interestingness implies more subjective connotations than

meaningfulness. However, this thesis is not focused on association rules mining, and we do not further discuss the difference between these two notations.

The third note is that the "redundant" or "redundancy" used in the above remarks is different from that used in the literature. For instance, the concise approaches refer to those patterns that can be represented (or recovered) by the concise subset as redundant without an association with the "meaningfulness".

2.3.12 Other examples

To understand the above listed drawbacks intuitively, let us look at the following concrete examples.

Example 2-2: Suppose $V_1, V_2, \ldots, V_8$ are dancers having participated in a dance contest of different styles, solo, "pas de deux", etc., and Table 1 is their performance records, where each T_i represents a performance, and the corresponding VID records the necessary dancers in that performance. Now, consider the following questions:

Question 1: Who are the active dancers? This is equal to a query of frequent element mining, and the raw frequentness works. For instance, V_1, V_2, V_4, V_7, V_8, would be entered into the answer set because of their higher raw frequentness than that of other elements (this can be referred from Table 8 in Chapter 4).

Question 2: Who could be the active solo dancers? This is equal to asking the frequent length-1 pattern. Indeed "solo" can be taken as a pattern in this case; and similarly "pas de deux" is another pattern. In conventional mining approaches, V_1, V_2, V_4, V_7, V_8, would still be the priori answer than other elements simply because of their higher (raw) frequentness. However, this answer is contradictory to common sense: element V_5, who

has played solo twice in this record while none of others has done so, should be the first answer! In other words, V_5 is underfitted but V_1 is overfitted at least to some degree. In this regard, one may argue, even though, for instance, there is no record for V_1 as solo, s/he might be a good solo dancer since s/he has been so active and performed much more different styles than V_5 in the contest. We do not fully have objection to this argument, but at least it is only an assumption that V_1 could be an active solo dancer; one is good at multiple person dance is not necessarily good at solo, and using their raw frequentness to mean V_1 is more likely than V_5 to be an active solo dancer is certainly not reasonable or convincing. Similarly, determining the active "pas de deux" dancers would have similar confusions.

Below is another similar example to see how the conventional mining approach could be misleading.

Example 2-3: Assume Table 1 is a criminal record of same sort (e.g., burglars) kept in a police station, and V_1, V_2, ..., V_8 are the criminals who were involved in cases T_i (i = 1, 2, ...u). Now, suppose an unsolved case is reported and the case was done by a single person among those criminals. The immediate action of the police is then to use pattern mining software to search from Table 1 to see who could be the most probable suspect(s). Again, this applies for length-1 pattern mining rather than an element mining, and by conventional approaches, V_1, V_2, V_4, V_7, V_8, would be the priori answer because of their higher (raw) frequentness. If so, the police officers would be very likely misled and overlooking the more possible suspect V_5. Similarly, the police officers would be fooled by the mining software in search cliques of other number of members. For instance, to

decide which combination, V_2V_6 and V_2V_3, could be more likely a pattern (criminal conspirers), the police offers would get a crystal clear answer, V_2V_3, from convention mining approach, since $F(V_2V_3) > F(V_2V_6)$. However, the answer would not be obvious: V_2V_6 is observed once but not V_2V_3, although $F(V_2V_3)$ is larger than $F(V_2V_6)$. More convincing answer can only be obtained after the raw frequencies have been adjusted.

The above examples illustrate that the configuration and semantics of a pattern is application dependent, for instance, in the dancer example, the correspondence between a length-1 pattern and a solo dancer. In these examples, length-1 patterns are meaningful but the related mining approach should be different from that mining frequent elements. In other applications, individual elements may not be meaningfully taken as (length-1) patterns. For example, in the known market-basket problem, it would be hard to identify an individual commodity as a pattern in a business sense. Then, in this case, frequent element mining could be meaningful but length-1 pattern mining may not. In other words, conventional mining approaches work for frequent element mining but not properly for length-1 pattern mining. These examples also illustrate the inapplicability of full repeatable (re)sampling and hence the full enumeration pattern generation regime. For example, V_1 might be a pattern with V_2, or with V_3, but s/he may not be a pattern with other agent(s) at the same time drawn from a tuple in either the dance or the criminal case.

The above examples have illustrated the following: the difference between element mining and pattern mining; the mixture of the two in the conventional mining approaches; overfitting or underfittin; bias towards generated patterns; why the reliability

of mining result by conventional approaches is questionable; and hence how important it is to adjust the raw frequentness in an application.

2.4 Challenges and motivations

We now have investigated the most fundamental issues affecting proper pattern mining in general. These issues exhibit in various aspects, but they can be traced into two main approaches: the use of the conventional frequentness measure support, and the full enumeration generations mode. These issues and their roots, however, have not been well addressed in the conventional mining approaches such as those reviewed in the first part of this chapter. Most of the conventional approaches pay attention to mining efficiency but less attention on proper measure of the frequentness measure and the refinement of the mining results.

The open problem is to resolve the addressed problems effectively. As the problems and their roots have been identified clearly, our first goal is to reformulate s_z, the frequentness measure; this is presented in Chapter 3. Our second goal is to resolve problems raised by full enumeration pattern generation regime. This is a much intricate task, and we deal with it in the remaining chapters.

We notice that, resolving the identified problems is a challenge, since the problems as studied above are more philosophical than computational. Furthermore, the challenges are not only from the philosophical problem itself, but also from the fact that there are no ready test rules, tools, or testimonies to guide our work. We have seen tests and comparisons of relative mining efficiency from most of the research proposals, for instance, the benchmarks of FIMI (Frequent Itemset Mining Implementations) [25] and

of [57]. However, we rarely see the testimonies or benchmarks for mining correctness, especially of semantic correctness. Without proper verifications and comparisons, a research proposal would not be accepted or utilized by domain users. This is why data mining in general is still far from a mature technology such that it could be confidently used in knowledge system or in business decisions. However, the lack of benchmark or testimony means the lack of well established underlying theory!

At the same time, even in terms of mining efficiency, the efforts made are hard to fully comprehend, since there are too many proposals, the correct choice is overwhelming for many users. Researchers have noticed this problem, and there is an appeal for a unified theory over the numerous and ad hoc proposed approaches based on a poll of more than ten data mining experts [19]. However, our emphasis is not on the unification of the proposed approaches, but more on the reinvestigation of the theoretical foundations of pattern mining. The issues listed above, especially the refinement of the patterns generated, have not been addressed, to our knowledge, by conventional mining approaches, and our purpose is to provide an insight into the issues and their resolutions. We aim to describe the mentioned philosophical problems and their resolutions over solid mathematical basis, so that findings presented in this thesis could serve as references, criteria for reliable benchmark and test tool buildups. From delivery point of view, our proposal would lay a refinement foundation such that the number of meaningless patterns could be substantially reduced without imposing domain constraint before the results being delivered to the user, who then may or may not refine the result set further depending on the application requirement.

Chapter 3. Resolution of probability anomaly and s_z

In this chapter, we present a primary resolution of the probability anomaly through a reformulation of the measure s_z. The first reason to reformulate s_z is its drawbacks analyzed in the previous chapter. The other reason is that, a (frequent) pattern mining, as its name implies, can be simply defined as a "frequentness based mining". It differs from other data mining, such as classification or clustering that uses other characteristics of the elements in question for the mining. Since frequentness is the only criterion, a proper definition of frequentness measure is therefore of fundamental significance.

To resolve the probability anomaly and to reformulate s_z, we first need to see the theoretical justification of the use of s_z. In this regards, the relevant theories are the classic probability theory and the multivariate probability theory.

3.1 The classic probability theory and s_z

The classic frequency based probability space [16, 22] is defined to be a triple (Ω, ω, P):

1. The sample space Ω, is a nonempty set whose distinct elements $V_1, V_2, ..., V_j$ (j = 1, 2, ..., n), as stored in the second column of DBo (Table 1), are known as *outcomes* or *states* of the domain in question [16, 17]. The total number of elements of Ω is noted as $|\Omega| = n$.
2. The event set ω[3], is a power set (2^n) of Ω in general, or a subset of such power set in a particular application. An *event* Z is a combination of any number of elements in Ω. Data stored in Table 1 are examples of the events labelled with T_i.

[3] Formally, ω is an σ-algebra, which is not discussed in this thesis.

3. The probability measure P is a function from ω to the real numbers in [0, 1] that assigns each event Z a probability between 0 and 1. The probability function P (Z) satisfies the probability axioms:

   1) $P(Z) \geq 0$, (3-1)

   2) $P(\omega) = 1$, (3-2)

   3) $P(\bigcup_j Z) = \sum_j P(Z^j)$, (3-3)

   if $\{Z^j\} \subseteq \omega$ is a countable collection of pairwise disjoint sets.

Compared with the above theory, the concept of pattern in pattern mining is exactly the same as that of an event. Then it is natural to expect the theory of pattern mining to be established on the classic probability theory. The related problem is how to determine the probability of every event (pattern) involved in an application. For simplicity, we use an example of two elements A and B only, and an experiment (sample) given in Table 3. In the classic approach, we follow the following conventions:

1. Each data tuple records one observed event. In relational database theory, this corresponds to the first normal form (INF) requiring each cell of a domain to store an atomic value only.
2. Each observation is used once and only once in frequency count.
3. Consequently, the accumulative frequency w is equal to the sample size (data size) u.
4. The probability of an event is taken to be its frequentness – the relative frequency f(Z)/w or f(Z)/u, *per se* the experiment output given. That is,

$$p(Z) = f(Z)/w = f(Z)/u. \qquad (3\text{-}3)$$

Example 3-1, From Table 3, we can get, p(A) = f(A)/u = 2/5 = 0.4, p(AB) = f(AB)/u = 2/5 = 0.4, and p(B) = f(B)/u = 1/5 = 0.2.

Obviously, the above probabilities are additive and they sum to 1. As we can see, in this case, there is no joint or disjoint issue involved in the related probability determinations, simply because the events and their frequencies are all from observations. Note that, although AB is a combination of A and B, but AB itself is an event different from A or B.

However, the above observation does not prevent one from analyzing the joint relation between two events, for instance A and B. This is done using the conditional probability theory,

$$p(B/A) = p'(AB) / p'(A).$$

Note that the denominator p'(A), called "absolute probability" of A [16], is different from p(A) given above, and,

$$p'(Z) = counts(Z) / u, \qquad (3\text{-}4)$$

which means:

$$p'(Z) = S(Z) / u = s(Z). \qquad (3\text{-}5)$$

Note that, the absolute probability p'(Z) cannot be compared with one another and hence not additive directly with one another, and so cannot be s(Z), as seen in following example.

Table 3. Original sample

TID	VID
T1	A
T2	AB
T3	AB
T4	A
T5	B

Example 3-2, From Table 3, we can get, p'(A) = S(A)/u = 4/5 = 0.8, p'(B) = S(B)/u = 3/5 = 0.6, and p(AB) = S(AB)/u = 2/5 = 0.4.

From the above we see, p'(A) and p'(B) are increased from p(A) and p(B) in Example 3-1. The reason is that Example 3-1 corresponds to three events, A, B and AB, while Example 3-2 corresponds to two events A and B only. In this case, we only know p'(A) + p'(B) – p'(AB) = 1. From this only condition we could not know exactly what the respective "net" probabilities of these two events are, and p'(A) and p'(B) are certainly not comparable. This incomparability will become even clearer in the next section.

At the same time, the joint relation can be and should be applied to event A and AB:

$$p((AB)/A) = p(A \cap (AB)) / p'(A) = p(AB) / p'(A) = p(B/A) .$$

That is, the joint relation between B and A is the same as that between AB and A. This is very rational and understandable. It implies then, the full enumeration based pattern generation of B (or A) from AB is questionable, since the relations between B and A and between AB and A are the same.

From (3-4) and (3-5), we see that the support s(Z) defined in (2-1) and used in pattern mining finds its equivalence in classic probability theory, s(Z) = p'(Z). And, it is in this property that one indicates that the supports cannot be additive as mentioned in Section 2.3.4. However, if s(Z) is not additive, two other issues arise:

1) s(Z) cannot be used as the frequentness measure, simply because, for instance, p'(A) and p'(B) cannot be compared with each other.
2) Consequently the comparable measures we can use are p(A) and p(B), which, however, lead us to go back to the results of the original problem as seen in Example 3-1; at the same time, it implies no pattern generation to render.

We assume that the above two consequences are not what conventional pattern mining approaches wanted. To understand these two issues and to see the comparability between s(Z) and p'(Z) further, let us look at them alternatively through multivariate joint probability distribution theory, presented in the next section.

3.2 The multivariate probability theory and s_z

The "bitmap indexing" method used in some articles [41] could be seen as a link between pattern mining and the multivariate probability theory.

Under the multivariate probability theory paradigm and in bitmap indexing, an element in Table 3 is taken to be a variable of two random values only: 1, if the element is present in a tuple, or 0 otherwise. Then, Table 3 is transformed into Table 4. From this table, we can get the corresponding joint probability distribution contingency table (Table 5).

Example 3-3: In Table 5, the middle two columns and rows represent the joint probability distribution of the two variables. For instance p(A = 1, B = 0) = 2/5 = 0.4, which is equal to p(A) in Example 3-1.

Table 4. Bitmap indexing

TID	A	B
T1	1	0
T2	1	1
T3	1	1
T4	1	0
T5	0	1

The last column and the last row in Table 4 present marginal probability P_A and P_B respectively, where the expression $P_A(A = 1, \bullet)$ reads the marginal probability of A when A is valued at 1, and the dot indicates that the other variables can be valued at any value. That is:

$$p_A(A = 1, \bullet) = \text{counts}(A = 1, \bullet) / u. \qquad (3\text{-}6)$$

That is, to get the marginal probability $P_A(A = 1, \bullet)$, it counts whenever A = 1. This means exactly the same as that to derive s(A) defined in (2-1). Then from the results of Example 3-2 and the marginal probabilities from table 5, the following relations hold:

$p'(A) = p_A(A = 1, \bullet) = s(A)$, and

$p'(B) = p_B(B = 1, \bullet) = s(B)$.

That is, the measure support s(Z) used in pattern mining is exactly equal to the marginal probability of Z with Z positive. Note that, marginal probability is a synonym of absolute probability [16].

A side effect here is that the equality between s(A) and $P_A(A = 1, \bullet)$ provides the simplest way to prove the downward closure property numerically. This property is mentioned in Section 2.2, meaning the frequentness of a pattern B is no less than its supper pattern AB for instance. That is: $s(B) \geq s(AB)$. This is because:

$s(B) = p_A(B = 1, \bullet)$ and

$s(AB) = p_{AB}(A = 1, B = 1, \bullet)$.

Since the former is less constrained than the latter, it is then obvious that $s(B) \geq s(AB)$.

Table 5. The contingency table

A \ B	0	1	P_A
0	0	1/5	$P_A(A = 0, \bullet) = 1/5$
1	2/5	2/5	$P_A(A = 1, \bullet) = 4/5 = s(A)$
P_B	$P_B(B = 0, \bullet) = 2/5$	$P_B(B = 1, \bullet) = 3/5 = s(B)$	

However, a very important observation here is that, the above proof is only numerical; it violates the comparison rule embodied in the probability theory! It is so, since

$s(Z) = p_Z(Z = 1, \bullet)$, then s(Z) should keep the same properties of the marginal probability $p_Z(Z = 1, \bullet)$. However, as known, marginal probabilities can only be comparable within the same marginal distribution. That is, $P_B(B = 1, \bullet)$ can only be compared with $P_B(B = 0, \bullet)$. Meanwhile, such comparison is trivial in the bitmap index situation, since,

$$P_B(B = 0, \bullet) = 1 - P_B(B = 1, \bullet).$$

It follows that, the measure support s(B) cannot be compared with s(A), or with s(AB), etc., but can only be compared with $s(\neg B)$. Such comparison, however, is not only trivial, but also meaningless in two aspects. Firstly, $s(\neg B)$ is not independent of s(B). Secondly, the original dataset does not produce $s(\neg B)$ at all, simply because a database (e.g., Table 3) does not record unobserved objects. Similarly, in the joint probability distribution, Table 5 gives $p(A = 0, B = 0)$, but Table 3 does not produce its equivalence $p(\neg A \neg B)$.

The above means that the bitmap indexing is not a lossless transformation of the original data mining problem. Indeed, there is a question whether the transformation is appropriate, since the elements A or B presented in the original problem and Table 3 are values of a domain (a variable) VID, but the transformation makes each element (value) as a variable! However, we do not discuss further the pros and cons of this indexing approach, since it is not the focus of this thesis.

What we can see from the above example is that, for instance, s(B) is mistakenly taken to be the frequentness measure might have been because of some concept confusions and notation illusions, for instance between P(B) and P'(B). Furthermore, in marginal

probability terminology, s(B) is exactly to mean s(B = 1, •). However, the dot is easy to be ignored since it means the values of other variables than B do not matter; meanwhile, B = 1 is simplified as B only, then s(B = 1, •) is simplified as s(B), which then falls into misunderstanding again.

Now, we can conclude from the above, the measure support s(Z) in pattern mining can find its equivalence in either the classic or the multivariate probability theory, but neither of the theories justifies its use as the pattern frequentness measure. This creates a serious dilemma for the conventional pattern mining approaches: If s(Z) is not taken to be the frequentness measure, then frequentness based pattern mining would become baseless, since there is no other frequentness measure established yet. On the other hand, if s(Z) is taken to be the measure of pattern frequentness, then s(Z) must be comparable with one another and hence additive, but in this case the probability anomaly arises, which cannot be ignorable.

The above described dilemma reveals a theoretical fallacy of pattern mining. How to fix this fallacy becomes important in pursuing effective pattern mining. We suggest that any remedy should satisfy the following requirements:

1. Allow pattern generations from the original dataset.
2. Maintain the occurrences of a pattern as the base of its frequentness.
3. Pursue conformability of the pattern frequentness measure with the recognized theories, particularly the classic and the multivariate probability theory.

The first two terms above attempts to recognize the mining operations already exercised to avoid discontinuity; the third attempts to correct the identified inappropriateness to approach theoretical soundness of the mining operations.

3.3 The multi valued state viewpoint and the resolution of s_z

The previous two sections have demonstrated that s_z cannot be justified by either classic or multivariate probability theory. Here we look further into why and how this problem arises. The problem is indeed induced from pattern generations from each data tuple. Such generation violates the conventions listed in Section 3.1, wherein each data tuple contains one event only and is used only once in frequency calculations and the accumulative frequency w is equal to the data size u. In other words, if we consider the elements of each tuple in DBo (Table 1 or 3) as an assembly, i.e., a single pattern, then the column VID is "single value stated", and we term such patterns as "original patterns". From this point of view, the accumulative frequency w of the patterns is the same as the cardinality u of the DBo, and the probability anomaly issue does not arise. However, when other patterns are "generated" or "enumerated" from the same tuple, the interpretation of the values of the column VID (of Table 1, for instance) has changed. That is, in the miner eyes, each cell does not hold only one but multiple values in database language, multi events in probability theory, or multi patterns in data mining terminology. Let us look at the operation of pattern generation again.

Assume V_1, V_4, and V_7 stands for Bread, Coffee, and Milk respectively in a market transaction T_1 of Table 1, and the customer who made this transaction indeed wanted to combine Bread with Milk as one menu (one pattern), and Coffee with Milk as another

pattern. Then T_1 is truly composed of at least two patterns. The purpose of pattern generation is then to recover those patterns merged in a tuple. It is in this interpretation that the generation is justifiable (however, the full enumeration generation over every tuple may not be justifiable. This will be clarified in section 5.1).

The above interpretation can be adopted to the primitive mining approach too (refer to Section 2.2), which enumerate patterns from the element space Ω, but such enumeration is equivalent to the enumeration from each data tuple. Then, based on classic probability theory, one observation describes one event (or pattern); then multiple patterns correspond to multiple observations. That is, the primitive mining approach assumes that a single data tuple of the original dataset to embody multiple observations (tuples) of single patterns.

The above would be the most favorable explanation of the legality of the pattern generation approach. From this point of view, column VID of Table 1 is "multi value stated". It is this multi value stated problem that breaks the conventions listed in Section 3.1established for applications of classic probability theory, and hence s_z defined in (2-1) is no longer compliant with the classic approach and leads to probability anomaly. Based on the understanding of these problems, it is then natural to expect a resolution of s_z by probability measures over multi valued state situations. However, there has been no such multi-value state

Table 6. DBv

TID'	Patterns
t_1	V_1
t_2	V_4
...	
t_j	$V_1V_4V_7$
...	
t_k	V_1
t_s	V_1V_2
...	
T_{118}	$V_1V_2V_6V_8$

probability theory available. Instead, the above analysis has already signified a resolution based on classical probability theory.

In classical statistics and probability theory terminology, a pattern generation is equal to a "re-sampling" over a data tuple of the original dataset, since an event stored in a tuple is a subset of the sample space Ω according to the classic probability theory [16] (refer to section 3.1). Then, the full enumeration pattern generation approach means to re-sample (enumerate) every possible pattern from every tuple contained in the DBo based on uniform distribution assumption. We now put all the re-sampled patterns into a (virtual) pattern database, DBv, as shown in Table 6. Then the sample size w can be defined as:

$$w = |DBv|. \qquad (3\text{-}7)$$

We note here that to get w is not much harder than to obtain the dataset size u. We present the solution for w in Chapter 4.

Now the definition of the probabilities of the patterns becomes straightforward. If we continue the use of the support, we can keep it in its similar format as defined in (2-1) but the denominator u must be changed into the cardinality of DBv, w. That is:

$$S'_z = S_z/w, \qquad (3\text{-}8)$$

where, S_z is the occurrences or "raw frequency" *F* of Z.

In other words:

$$S'_z = F(Z)/w = p(Z), \qquad (3\text{-}8a)$$

where p(Z) is the probability of pattern Z as defined in the classic case (3-3).

The above reformulation can be intuitively further understood from the following problem.

Problem 3-1: Given a database of u tuples, from which a total of m patterns have been generated with their accumulative frequency w. Suppose an individual pattern Z has an absolute (raw) frequency S_z, then how should its frequentness or relative frequency be expressed?

The answer to Problem 3-1 would be unarguably S_z/w (using 3-8) but not S_z/u (2-1). With the above resolution, problem 2-1 can be simplified as a typical sampling or probability problem as seen in most text books, and can be reformulated as:

Problem 3-2 (revised mining problem): Given a universe Ω of n distinct elements V_1, V_2, …, V_n, and a pattern sample of size w from Ω as stored in DBv, output all of the frequent patterns Z, such that $s'_z \geq s_{min}$.

With the reformulation of s_z, the probability anomaly issue is automatically eliminated. And, as can be seen in next two sections, the two typical symptoms of overfitting, namely, too many frequent patterns and unstable mining result set, will be greatly corrected.

3.4 Primary overfitting / underfitting quantifications

The above has described not only how s_z defined in (2-1) is reshaped, but also how the probability anomaly issue is primarily eliminated. Here, we present how the degree of overfitting or underfitting of conventional pattern mining approaches could be quantified against the reformulated s_z. We term this quantification as the primary overfitting or underfitting ratio r_s, depending on whether $r_s > 1$ or $r_s < 1$, where,

$$r_s = s_z / s'_z \qquad (3\text{-}9)$$

We then get:

$$r_s = s_z / s'_z = (Sz/u) / (Sz/w) = w/u = \lambda, \qquad (3\text{-}10)$$

where λ is the average raw frequency per tuple of the DBo.

This reflects the observation that more patterns will be generated from longer data tuples, which then causes pattern frequency to increase faster and consequently overfitting increases. That is why r_s is proportional to λ. As a result, it validates the assertion made in 2.3.5 that pattern frequency over datasets of longer data tuples will be higher and hence the overfitting issue will be severer than that over datasets of shorter data tuples.

Since λ can be very large if the length of the data tuples is large, then r_s can be very large. We noticed in the running example and Table 2 (as well as Table 7 in next sections) that, for such a small dataset with average tuple length around 3, the overfitting ratio is over 10 against the raw probability distribution.

The above explains why so many frequent but meaningless patterns result in conventional mining approaches, and justifies our assertion that overfitting is naturally embodied in previously proposed mining approaches. This is a significant finding, which strongly disqualify the extensively used conventional "support" indicator s_z.

3.5 Numerical comparisons

For a more intuitive understanding of the difference of the evaluation of the pattern frequentness in conventional and the proposed reformulated s_z, we give the related comparisons in Table 7 based on the data given in Table 1. In Table 7, the numbers and their semantics of column A, B, E, and E' are copied from Table 2. That is, column E

shows the accumulated conventional "frequentness" of patterns of same length, as well as the overfitting ratios against the reformulated s_z, based on the first 9 tuples of Table 1. Column E' shows the same but based on all 10 tuples of Table 1. Column Ea and Ea' present the accumulative reformulated frequentness $\sum s'_z$ of patterns of same length k, where the probability anomaly is eliminated. The other columns starting from column T until the last show some example patterns and their related raw frequencies and raw frequentness obtained from the conventional and the reformulated support measures respectively, where column x, y, and z are the results from the first 9 tuples, while column x', y', and z' are the results after the last tuple (V_1, V_2, V_3, V_8) has been added into Table 1.

Table 7. Comparisons of the resulted parameters based on data of Table 1

k	$\#Z_k$	$\sum s_z$	$\sum s'_z$	$\sum s_z$	$\sum s'_z$	Example	S_z	s_z	s'_z	S_z	s_z	s'_z
A	B	E	Ea	E'	Ea'	T	x	y	z	x'	y'	z'
1	8	2.56	0.33	2.8	0.24	V_1 V_2 V_4 V_7	4 4 4 4	.45 .45 .45 .45	.039 .039 .039 .039	5 5 4 4	.5 .5 .4 .4	.042 .042 .034 .034
2	18	3.33	0.36	3.6	0.31	V_4V_7 V_1V_2 V_1V_8	4 2 2	.45 .22 .22	.039 .019 .019	4 3 3	.4 .3 .3	.034 .025 .025
3	21	2.9	0.22	3.0	0.25	$V_2V_4V_7$ $V_1V_2V_3$	2 1	.22 .11	.019 .010	2 2	.2 .2	.017 .017
4	15	1.8	0.08	1.7	0.14	$V_2V_4V_7V_8$ $V_1V_2V_3V_8$	2 1	.22 .11	.019 .010	2 2	.2 .2	.017 .017
5	6	0.67	0.01	0.6	0.05	$V_1V_2V_3V_4V_7$	1	.11	.010	1	.1	.009
6	1	0.11	0.01	0.1	0.01	$V_1V_2V_3V_4V_7V_8$	1	.11	.010	1	.1	.009
$\sum$	69	11.4	1.00	11.8	1.00							

We note that, with the reformulated s_z, the number of frequent patterns has been greatly reduced. With $s_{min} = 20\%$, there is no frequent pattern in the reformulated case against 23 patterns in the conventional case. More strikingly, with $s_{min} = 10\%$, there is still no frequent pattern in the reformulated case, but all of the 69 patterns are frequent in the conventional case (examples are shown in columns T and y). Obviously, the reformulated case reflects reality that we could not mine a big portion of frequent patterns from a small dataset DBo. At the same time, the results illustrate how the two overfitting symptoms, unstable mining result set and rapid pattern frequentness growth, have been remedied with the reformulation of s_z. Using s'_z, the more frequent a pattern is, the more stable is its frequentness as the dataset size changes, as shown in column z and z'. This is what is normally to be expected, with increasing data size, the frequentness of every pattern approaches asymptotically to its natural degree. The conventional s_z in general increases faster than s'_z.

The above observations can be formalized as follows: against a data size increase, s'_z increases slower than s_z, if the added accumulated frequency produced from the added data tuple is over the average accumulated frequency per tuple. Secondly, as long as the added data tuple contains Z, s_z can always increase, while s'_z may not, and it can even decrease. Thirdly, a larger s'_z will increase slower than a smaller s'_z .

Proof: initial u, w, s_z and s'_z for a given pattern Z and its raw frequency F_z. Now suppose one data tuple added into the dataset, that is, $\Delta u = 1$, which could cause F_z to increase at most by 1, since one data tuple can generate a particular pattern once, while w

will be increased by Δw ($= 2^x - 1) > 1$ unless $x \leq 1$, as have been stated in Chapter 2, where x is the length of the added tuple. Then:

$$\Delta s_z / s_z = \Delta(F_z / u) / (F_z / u)$$

$$= ((u\Delta F_z - F_z \Delta u) / u^2) / (F_z / u) = 1/ F_z - 1/u, \qquad (3\text{-}11)$$

and $\Delta s'_z / s'_z = \Delta(F_z / w) / (F_z / w)$

$$= ((w\Delta F_z - F_z \Delta w) / w^2) / (F_z / w) = 1/ F_z - \Delta w / w.$$

Considering $w = \lambda u$, where λ is the average accumulated frequency per tuple (refer to (3-10)), the above can be reformulated as:

$$\Delta s'_z / s'_z = 1/ F_z - \Delta w / (\lambda u) = 1/ F_z - (1 /u) (\Delta w/\lambda). \qquad (3\text{-}12)$$

The above formulae (3-11) and (3-12) state that:

1) $\Delta s'_z / s'_z < \Delta s_z / s_z$, as long as $\Delta w > \lambda$, which then proves the first conclusion. At the same time, it implies importantly, to keep $\Delta s'_z / s'_z$ comparable with $\Delta s_z / s_z$ the data tuple length will ultimately decline toward 1.

2) (3-11) tells $\Delta s_z / s_z > 0$ always hold, since $F_z < u$ (if $F_z = u$, Z can be fully removed from the dataset, since every data tuple holds Z). However, (3-12) indicates that $\Delta s'_z / s'_z$ can be either positive or negative, even if an added tuple makes F_z increased (by 1). It then proves the second conclusion. Meanwhile, it brings another important implication: the data tuple length matters much in a mining problem: a fairly long tuple added in can cause all patterns' frequentness decrease, and lead to underfitting, since a long data duple can cause w dramatically increased.

3) Notice that, a smaller s'_z means a smaller F_z, and hence (3-12) proves the third conclusion as well.

Although the above added only one data tuple in the proof, the proof can be easily generalized to additions of multiple tuples.

From the above, we see that, the s_z resolution not only revaluates pattern frequentness, but more importantly, it reveals a number of interesting and intrinsic properties underlying pattern frequentness measure and the mining effectiveness in whole.

3.6 The significance and impacts of the resolution

The following summarize the proposed resolution:

1. We explain why the conventional widely used support s_z is not a qualified frequentness measure, and how the probability anomaly occurs.
2. Based on this, we have provided a resolution. This resolution while simple is effective, since it radically resolves the issues addressed in sections 2.3.1 through 2.3.6, especially the probability anomaly. At the same time, the resolution fulfills the requirements stated at the end of Section 3.2.
3. Consequently, the resolution would have the following impacts on pattern mining in general:

    a) Because of the equalization of the pattern frequentness and the probability measure of events, there is no longer any need to use a dedicative "support" concept to mean pattern frequentness. For instance, we can use 3% or 5% to be the frequentness threshold without bothering user to define s_{min}. Such thresholds are often used in various research and applications, though they are not formally defined or required [58].

b) Under the proposed s'_z regime and because of the unification of accumulative frequentness to 1, if the total number of patterns is large, the individual pattern frequentness would be much decreased from that of conventional results. The degree of decrease is described by the overfitting ratio r_s. In this sense, one cannot expect many individual patterns' frequentness to be over 5%, for instance. In a case of large number of patterns, their probability distribution is more analogous to the (continuous) probability density distribution rather than the (discrete) mass probability distribution in classic probability theory. When use a 3% threshold for infrequent patterns under s'_z regime, it refers to all those patterns whose cumulated frequentness is less than 3%. Similarly, when we refer to the top 10% frequent patterns, it means all those patterns whose accumulated probability is equal to or larger than 10%. These statements with s'_z are consistent with the conventional probability theory and notations, while s_z regime does not maintain these conventions.

c) The above impacts will propagate to other mining applications based on pattern mining, for example, association rules mining, causation mining, and the like.

4. The above insights would also correct a viewpoint on pattern mining or data mining in general. The phrase "knowledge discover from database (KDD)" usually gives us an impression that the mining is *fact based*, since what a database contains are all observed facts or experimental results. However, from the above analysis we can see it not to be so. Although we can accept what a database holds are facts, the patterns generated are largely subjective, especially by the uniformly used full enumeration generation mode without justifications. Take the first tuple $T_1 = \{V_1, V_4, V_7\}$ from

Table 1 as an example, in terms of fact, we can only see T_1 proves $V_1V_4V_7$ in whole, but T_1 itself does not state its support forV_1V_4 or V_1V_7 or the like separately and equally. To take that T_1 supports all of those separated patterns is a very subjective assessment! In this sense, pattern mining or data mining in general is at most a *mixture* of *fact based* and *assumption based*, and in many cases, the latter plays a bigger role than the former, since the generated patterns could be much more than the observed ones. The issue addressed in section 2.3.8 on the bias towards generated patterns against the observed ones is indeed a bias for subjective against objective. Then a big task in pattern mining is to reduce the subjective involvement as much as possible and improve the mining objectiveness.

5. The proposed s'_z, resolves the probability anomaly and covers other related drawbacks addressed in Section 2.3, for example, the stiffness in reflecting reference pattern frequency changes. However, we notice it is not a complete or a final resolution for those addressed issues, including the overfitting/underfitting problems, since these issues are also caused or reinforced by the full enumeration pattern generation mode. This then goes back to the same issue addressed in point 4 above. Only after the number of unrealistic patterns has been reduced or their frequentness been reduced, could the s'_z of a realistic pattern approach closer to its true value. In this sense, the first requirement of the resolution stated at the end of section 3.2 should be modified as "allow pattern generations *while minimizing the unnecessary (meaningless) generations*". The remaining chapters present our efforts toward this objective.

Chapter 4. Fundamentals of raw pattern frequency distributions

To see how to modify the full enumeration pattern generation mode, we need first to know what the properties of the mode are. In this chapter, we explore these properties by following the convention and generate all possible combinations from every tuple, and we do not distinguish the connotations of patterns and combinations to simplify the elaboration. We term each generated pattern as a "raw" pattern, and its absolute occurrences S_z as "raw" frequency $F(Z)$. The relative frequentness P(Z) of an individual pattern follows formula (3-8a):

$$P(Z) = F(Z)/w = S_z/w \qquad (4\text{-}1)$$

Below we first introduce a dualism to facilitate our study; then we derive and prove a set of properties governing raw pattern frequency distributions.

4.1 A dual problem

Dualism is quite often used when an original problem is not easy to tackle, e.g., a profit maximization problem could be studied with its dual problem of cost minimization. Here, we use the dualism to introduce and prove some basic concepts, based on which, further mathematical properties of pattern frequency distributions are derived.

Dual transformation: we first transform the original database DBo to its dual database DBd by a translation τ:

$$DBo(TID \rightarrow VID) \rightarrow DBd(VID \rightarrow TID) \quad (4\text{-}2)$$

That is, τ exchanges the roles of TID and VID such that VID in DBd acts as the key attribute, each V_i (i = 1, 2, ..., n) representing a set of T_js that holds the same V_i in the original database DBo, as seen in Table 8. For example, in DBo, V_1 is referred by T_1, T_4, T_5, T_9 and T_{10}. So, in DBd, V_1 refers to these T_js in turn. In other words, if U_t means the universe of the elements T_j (j = 1, 2, ..., u), then V_i refers a subset of U_t, i.e., $V_i \subseteq U_t$

Table 8. The DBd

VID	TID
V_1	$T_1, T_4, T_5, T_9, T_{10}$
V_2	$T_2, T_3, T_5, T_9, T_{10}$
V_3	T_5, T_{10}
V_4	T_1, T_2, T_5, T_7
V_5	T_6, T_8
V_6	T_3, T_4
V_7	T_1, T_2, T_5, T_7
V_8	T_2, T_4, T_5, T_{10}

The concept of dual transformation is obvious and we do not present an algorithm for it; and we note that the transformation is similar to that used by vertical search approaches [9, 10].

The correspondences of the DBo and DBd are:

$$|DBo| = u, |U_v| = |\Omega| = n, \quad (4\text{-}3)$$

$$|DBd| = n, |U_t| = u, \quad (4\text{-}4)$$

$$\sum_{DBo} |T_j| = \sum_{DBd} |V_i|, \quad (4\text{-}5)$$

where |X| means the number of elements (the cardinality) of X; U_v means the universes of the domain VID in DBo and U_t is the universe of TID in DBd.

Furthermore, there are two other important dual concepts between DBo and DBd: the patterns and their frequencies. In the original problem, a pattern $Z_k = (V_pV_q..V_s)$, is generated within a cell of VID of DBo. In the dual problem, the same pattern Z_k is a

combination of the Vs vertically selected from different cells of the column VID of DBd. Since a V_i in DBd is taken to be a subset (of U_t), a combination of Vs then exactly represents a new subset – the intersection of the Vs. These concept conversions lead to the dualism of the "frequency", as given below:

Definition 4-1: The elements T_i ($i \in [1, u]$) held by a combination of number k of Vs, $Z_k = (V_pV_q..V_s)$, in DBd are the "intersected content(s)" (IC) of those held by each individual V_x involved in Z_k, denoted by $I_c(Z_k)$.

For instance, in DBd (Table 8) the elements held by V_1 is $\{T_1, T_4, T_5, T_9, T_{10}\}$, held by V_4 is $\{T_1, T_2, T_5, T_7\}$. Then the elements held by the combination (intersection) V_1V_4 is their "intersected contents", $I_c(V_1V_4) = \{T_1, T_5\}$; and, $|I_c(V_1V_4)| = 2$. The duality of a pattern's frequency is stated as follows:

Proposition 4-1: The "raw frequency" F of a pattern $Z_k = (V_pV_q..V_s)$ in the original Problem 3-1 equals to the cardinality of the "intersected contents" of the combination Z_k in the dual problem:

$$F(Z_k) = |V_pV_q..V_s| = |I_c(Z_k)|. \qquad (4\text{-}6)$$

The concepts expressed in definition 4.1 and Proposition 4-1 can be traced to the formal concept analysis theory [59] and are similar to those used in vertical search approaches as in [9, 10]. However, here we are not interested in what the intersected contents are,

but only in the number of such elements. Obviously, for a pattern of length-1 in DBd ($Z = V_i$), $F(V_i) = |I_c(V_i)| = |V_i|$.

To determine the frequency distribution, we present the dual problem below:

Problem 4-1 (dual problem): Given a database DBd as in Table 8, for every combination $Z_k = (V_iV_j...V_k)$ of length k, determine the number of its intersected contents (the frequency), $|I_c(Z_k)|$.

The following sections describe various concepts and solutions for the above problem, and we use "frequency" and "intersected contents" or I_c interchangeably later on.

4.2 The inclusion-exclusion principle

To study problem 4-1, we start with patterns of length one and refer to DBd (Table 8) where the universe $U_t = \{T_1, T_2, ...T_u\}$, and $u = |U_t|$. We notice that the length-1 patterns $V_1, V_2, ..., V_n$ are a (overlapped) partition of U_t in the dual problem, since each V_i (i = 1, 2, …, n) represents a subset of U_t. e.g. $V_4 = \{T_1, T_2, T_5, T_7\}$ (refer to Table 8). By set theory, if n and u are finite, we have:

$$U_t \equiv \bigcup_{i=1}^{n} V_i = V_1 \cup V_2 ... \cup V_n, \qquad (4\text{-}7)$$

and, $|U_t| = u$. (4-8)

Expand (4-7) and (4-8), and start from a very basic set operation (n = 2):

$|V_1 \cup V_2| = |V_1| + |V_2| - |V_1V_2|$,

where V_1V_2 is a shorthand for $V_1 \cap V_2$.

In general, if n and u are finite, for $U_t \equiv V_1 \cup V_2 \ldots \cup V_n$, combining (4-7) and (4-8), we have:

$$|U_t| = (|V_1| + |V_2| + \ldots + |V_n|) - (|V_1V_2| + |V_1V_3| + \ldots + |V_{n-1}V_n|) + (|V_1V_2V_3| + \ldots + |V_{n-2}V_{n-1}V_n|) - \ldots \pm |V_1V_2 \ldots V_n|$$

$$= \sum_i |V_i| - \sum_{i,j,\, i<j} |V_iV_j| + \sum_{i,j,m,\, i<j<m} |V_iV_jV_m| - \ldots \pm |V_1V_2 \ldots V_n|$$

$$= u, \qquad (4\text{-}8a)$$

where, each $\sum$ represents a sum of the raw frequencies of a "collective" of patterns of the same length, $\sum|V_pV_q,\ldots,V_s|$. Again, $|V_pV_{q\ldots}V_s|$ is not the length of $(V_pV_{q\ldots}V_s)$ but the number of its intersected elements (I_c), which equals the frequency as stated in Proposition 4-1.

Formula (4-8a) is referred as "inclusion-exclusion principle" [20], since the alternating signs presented in the formula imply the compensations of possible excessive inclusion or exclusion of the elements involved in every $(V_pV_{q\ldots}V_s)$ during the calculation. This principle has been used, for instance, in concise representation study [4, 30], and in estimation of upper bounds of candidate patterns [24]. In this thesis, we use this principle as starting point to explore more general laws governing pattern frequency distributions under the full enumeration regime.

4.3 The raw collective frequencies

Now, to simplify expression (4-8a), we use L_k to mean a collective of patterns of length k, H_k to mean the "raw collective frequency" of L_k, and C_k to mean the number of patterns of the collective L_k, More formally:

Definition 4-2: The "raw collective of patterns of the same length k":

$L_k = \{Z^j_k\}$, where j = 1, 2, …, C_k, (4-9)

and Z_k^j is the j[th] pattern[4] within L_k.

Definition 4-2a: The "raw collective frequency" of L_k:

$$H_k = \sum_{p, q, \ldots, s,\ p<q<\ldots<s} |V_pV_q\ldots V_s|$$

$$= \sum_{L_k} |I_c(Z_k^j)| = \sum_{L_k} F(Z_k^j). \quad (4\text{-}10)$$

Then, (4-8a) can be reformulated as:

$$|U_t| = \sum_{k=1}^{n}((-1)^{k-1}\sum_{L_k} F(Z_k^j)) = \sum_{k=1}^{n}(-1)^{k-1} H_k = u. \quad (4\text{-}11)$$

The "inclusion-exclusion principle" then becomes easy to express by (4-11). An important point to note here is that, (4-8a) and hence (4-11) is originally motivated for the frequencies of patterns of length-1 V_i, but ends up with the involvement of frequencies of patterns of all longer lengths.

The above concept and formulae are fundamental for the rest of this thesis, and we shall explore a number of their interesting properties. To avoid confusion, we summarize the concepts below:

A pattern Z_k is a subset of k elements of Ω, $Z_k = (V_pV_q\ldots V_s)$ in the original problem, and k is termed as the length of Z_k. In the dual problem, the combination $V_pV_q\ldots V_s$ becomes a label of a subset of U_t, and $|V_pV_q\ldots V_s|$ means the number of elements of U_t held by such subset, and termed as $|I_c(V_pV_q\ldots V_s)|$ or $|I_c(Z_k)|$, which is equal to the raw frequency of Z_k, or $|I_c(Z_k)| = F(Z_k)$. L_k is a collective (not a union) of all patterns

[4] j is for enumeration in (4-9) and (4-10) – j is a cardinal but not an ordinal number.

of the same length k. The total number of patterns within an L_k is C_k, and the collective raw frequency of L_k is H_k.

4.4 Fundamental propositions

As mentioned, the known "anti-monotonic" or "downward closure" property [2, 5], was originally taken as an intuition in [5]. Here we see how this property can be formally proved using dualism.

Proposition 4-2: The frequency of a pattern $Z_k = (V_pV_q \ldots V_rV_s)$ is no greater than that of any of its sub-pattern, say, $Z_d = (V_pV_q \ldots V_r)$, where $0 < d \leq k$. That is, $F(Z_k) \leq F(Z_d)$.

It is not very straightforward to prove the above proposition from the original problem. However, it could be much easier examined with the dual problem, the I_c notation and formula (4-6). The above proposition could be restated as:

Proposition 4-2': in the dual problem (4-1), $|I_c(Z_k)| \leq |I_c(Z_d)|$, where Z_k is the intersection of k Vs, $(V_p,V_q, \ldots,V_r, V_s)$, such that $Z_k = (V_pV_q \ldots V_rV_s)$, and Z_d is the intersection of d Vs, $(V_p, V_q, \ldots,V_r,)$ involved in Z_k, $(0 < d \leq k)$.

Proof: As given, $Z_k = Z_d \cap V_s$, and hence $Z_k \subseteq Z_d$, then, $I_c(Z_k) \subseteq I_c(Z_d)$, and $|I_c(Z_k)| \leq |I_c(Z_d)|$. By Proposition 4-1 and formula (4-6), it means, $F(Z_k) \leq F(Z_d)$. Proposition 4-2' and hence 4-2 is then fully proved.

Proposition 4-2 has been proved, for instance in [10] based on Galois lattice theory and [59], which, however, may be a bit difficult to comprehend for many readers. Here we proved it with basic set theory and dualism.

Additionally, as reader might have noticed, the above property is "monotonic", since by [21]:

If a measure m is monotonic, and if $X \subseteq Y$, *then,* $m(X) \leq m(Y)$.

The proof of Proposition 4-2' (or 4-2) is then conformable with the above definition, and Proposition 4-2should be reflected as a monotonic property. However, conventionally it has been referred as anti-monotonic! This is another mirror effect of dualism.

Following is another proposition that has not been explicitly addressed in other research.

Proposition 4-3: Given a pattern A and its sub-pattern B, the necessary and sufficient condition for the raw frequencies of A and B to be the same is that they are generated (re-sampled) from the same tuple(s) of the original dataset.

This might not be easy to prove by the original Problem 3-1, but again it becomes rather simple using the dual problem and the I_c notation. The proposition can be stated mathematically in the dual problem as:

Proposition 4-3': Given combinations A, B, and $A \subset B$ (or $B \subset A$), then, $|A| = |B|$, iff $I_c(A) = I_c(B)$.

The proof is obvious: if $I_c(A) = I_c(B)$, then $|A| = |B|$. On the other hand, if $|A| = |B|$, and because $A \subset B$ (similarly to $B \subset A$), only if $I_c(A) = I_c(B)$.

Proposition 4-3 is important and lays the foundation for pattern de-sampling to be seen in the remaining of this thesis. On the other hand, we need to address that, though proved as

above the three propositions are all based on the "full enumeration" pattern generation rule. This rule is very primitive, and when the rule is changed such that only partial but not full number of patterns is generated from a tuple, these propositions shall not be generally held any more.

Based on the above concepts, propositions and notations, we can now perform the calculations of the fundamental measures in pattern frequency distribution theory, the accumulative frequency w and each of the collective frequencies, H_k as defined in (4-10).

4.5 Formulae for w and H_k

During a scan of a database to obtain the total number of records u, it is easy to determine the length b_j of each record. The accumulative raw frequency w of all patterns then can be obtained precisely before the pattern generation as follows:

$$w = \sum_{j=1}^{j=u} \sum_{i=1}^{i=b_j} C_{b_j}^{i} = \sum_{j=1}^{j=u} (2^{b_j} - 1) = \sum_{j=1}^{j=u} 2^{b_j} - u, \qquad (4\text{-}12)$$

where $b_j = |T_j|$, is the number of elements held by a tuple T_j in the original dataset DBo. And, u = |DBo|.

Now, define g_k as the number of tuples holding a number of k elements in the original datasets DBo, and hence:

$$\sum_{k=1}^{k=\alpha} g_k = u. \qquad (4\text{-}13)$$

(4-12) then can be further simplified as:

$$w = \sum_{j=1}^{j=u} \sum_{i=1}^{i=b_j} C_{b_j}^{i} = \sum_{k=1}^{k=\alpha} (g_k \sum_{i=1}^{i=k} C_k^{i})$$

$$= \sum_{k=1}^{k=a} g_k(2^k - 1)$$

$$= \sum_{k=1}^{k=a} g_k T_k = \sum_{k=1}^{k=a} 2^k g_k - \mathrm{u}, \tag{4-14}$$

$$\text{where, } T_k = 2^k - 1, \tag{4-15}$$

which represents the number of patterns and hence the sum of their frequencies enumerated from a tuple of length k.

The simplification of (4-12) to (4-14) reduces the number of exponent operations from u to α, and as we know, commonly, α << u. Furthermore, we can completely avoid the exponent operations, starting by reformulating (4-14):

$$\mathrm{w} = \sum_{k=1}^{k=\alpha} (g_k \sum_{i=1}^{i=k} C_k^i) = \sum_{k=1}^{k=\alpha} \sum_{i=k}^{i=a} g_i C_i^k. \tag{4-16}$$

(4-14) and (4-16) produce the same w, but they represent different pattern generation strategies. (4-14) describes the case that, in a loop k, patterns of different lengths ≤ k are generated from tuples of same length k, but (4-16) states that, in a loop k, patterns of same length k are generated from all tuples of length ≥ k., and the result is then the H_k. That is:

$$H_k = \sum_{i=k}^{i=a} g_i C_i^k, \tag{4-17}$$

$$\text{and} \quad \mathrm{w} = \sum_{k=1}^{k=\alpha} \sum_{i=k}^{i=a} g_i C_i^k = \sum_{k=1}^{k=\alpha} H_k. \tag{4-18}$$

(4-17) and (4-18) can also be in vector and matrix expressions. Firstly, we define:

$$\mathbf{G_k} = (g_k, g_{k+1}, ..., g_\alpha) \tag{4-19}$$

as a "gathering vector" of dimension ($\alpha - k + 1$);

$$\mathbf{\Theta_k} = (C_k^k, C_{k+1}^k, ..., C_\alpha^k) \tag{4-20}$$

as a "setup vector" of dimension ($\alpha - k + 1$). In particular, when $k = 1$, Θ_1 is termed as "initial setup vector", and

$$\mathbf{\Theta_1} = (1, 2, ..., \alpha). \tag{4-21}$$

In addition, we define a "product vector" $\mathbf{E_k}$, each element e_i of it being the product $g_i C_i^k$ ($i = k, k + 1, \ldots, \alpha$). That is,

$$\mathbf{E_k} = (e_k, e_{k+1}, \ldots, e_\alpha) = (g_k C_k^k, g_{k+1} C_{k+1}^k, ..., g_\alpha C_\alpha^k). \tag{4-22}$$

Then, (4-17) can be expressed as:

$$H_k = \sum_{i=k}^{i=a} g_i C_i^k = \mathbf{G_k} \bullet \mathbf{\Theta_k}. \tag{4-23}$$

Or, $$H_k = \sum_{i=k}^{i=a} g_i C_i^k = \mathbf{B_k} \bullet \mathbf{E_k}, \tag{4-24}$$

where, $\mathbf{B_k}$ is termed as a "base vector" of dimension ($\alpha - k + 1$), with all elements being 1:

$$\mathbf{B_k} = (1, 1, \ldots, 1). \tag{4-25}$$

In the following matrix expression, we use non-bold G_k and Θ_k to mean the corresponding matrixes:

$$H_k = \sum_{i=k}^{i=a} g_i C_i^k = G_k I_k \Theta_k. \qquad (4\text{-}26)$$

where, G_k is an 1 * (α – k + 1) "gathering matrix", starting from g_k: $G_k = (g_k \; g_{k+1} \; \cdots \; g_\alpha)$; Θ_k is an (α – k + 1) * 1 "setup matrix", and $\Theta_k = (C_k^k \; C_{k+1}^k \; \cdots \; C_\alpha^k)^T$. In particular, when k = 1, Θ_1 is an α * 1 "initial setup matrix", and $\Theta_1 = (1, 2, ..., \alpha)^T$; I_k is an (α – k + 1) * (α – k + 1) idempotent matrix, with all elements of the main diagonal being 1 while the rest being 0.

Among other significances, a use of the above vector and matrix formulae is its facility to obtain H_ks recursively without involving any exponent operation.

Using (4-17), we can directly get:

$$H_{k+1} = \sum_{i=k+1}^{i=a} g_i C_i^{k+1} \qquad (4\text{-}27)$$

Since $C_i^{k+1} = \frac{i-k}{k+1} C_i^k$ (from the combinatorics), (4-28)

then (4-27) becomes:

$$H_{k+1} = \sum_{i=k+1}^{i=a} g_i C_i^{k+1} = \sum_{i=k+1}^{i=a} \frac{i-k}{k+1} g_i C_i^k$$

$$= \frac{1}{k+1} \sum_{i=k+1}^{i=a} (i-k)\, g_i C_i^k$$

$$= \frac{1}{k+1} \mathbf{D_k} \bullet \mathbf{E'_k}. \qquad (4\text{-}29)$$

Where $\mathbf{E'_k}$ is a vector of all the elements of $\mathbf{E_k}$ except the first element being cut off, for instance, if $\mathbf{E_k}$ = (2, 3, 5), then $\mathbf{E'_k}$ = (3, 5); $\mathbf{D_k}$ is a "deductive vector", each of its element d_k^i being (i – k), and $\mathbf{D_k}$ = (1, 2, …, α – k). In fact, $\mathbf{D_k}$ can be seen as the first section of the $\mathbf{\Theta_1}$ vector (4-21) up to element α – k. The properties of $\mathbf{D_k}$ and $\mathbf{E'_k}$ make the H_k computation pretty easy as would be seen soon.

To process the recursive operations on all H_k s, we only need to know $\mathbf{E_{k+1}}$ and the initial vector $\mathbf{E_1}$. According to (4-24), we have:

$$H_{k+1} = \mathbf{B_{k+1}} \bullet \mathbf{E_{k+1}}, \qquad (4\text{-}29a)$$

Since $\mathbf{B_{k+1}}$ is a base vector, comparing (4-29) and (4-29a), we can easily see that, any element e^i_{k+1} of $\mathbf{E_{k+1}}$ is the computation result from (4-29):

$$\mathbf{E_{k+1}}: \{ e_{k+1}^i = \frac{1}{k+1} * i * e'^i_k \,,\ i = k+1, k+2, \ldots, \alpha\} \qquad (4\text{-}30)$$

where, i is the value of the i^{th} element of $\mathbf{D_k}$ and e'^i_k is the i^{th} element of $\mathbf{E'_k}$.

In addition, it is easy to see from (4-22):

$$E_1 = \mathbf{G_1} \bullet \mathbf{\Theta_1}, \qquad (4\text{-}31)$$

where $\mathbf{G_1}$ is the vector of the whole series of g_k, and $\mathbf{\Theta_1} = (1, 2, \ldots, \alpha)$.

(4-31) and (4-30) form the recursive computation of $\mathbf{E_k}$.

Table 9 is an example which uses the above formulae to compute all H_ks recursively and w over dataset DBo shown in Table 1. The first row of Table 9 lists the element of $\mathbf{\Theta_1}$, which is just an enumeration from 1 to α (here α = 6); and the second row lists the elements of $\mathbf{G_1}$ (the series g_i). These two lists are the only inputs. The bold numbers are the elemental results of (4-29), and they together form an upper triangular matrix, each row of it forms an $\mathbf{E_k}$ (k = 1, 2, …, α). For instance, Row 3 is the results of $\mathbf{E_1}$ (refer to 4-31) and hence H_1, by multiplying the corresponding elements of $\mathbf{G_1}$ and $\mathbf{\Theta_1}$. In row

4, to compute $\mathbf{E_2}$ and H_2, right shift $\mathbf{\Theta_1}$ by one column and get $\mathbf{D_1}$; or similarly but left shift $\mathbf{E_1}$ by one column and we get $\mathbf{E'_1}$. Then according to (4-30), the first element of $\mathbf{E_2}$, $e_2^1 = ½ (1 * 4) = 2$.

Indeed, the main diagonal elements are just a copy of the $\mathbf{G_1}$ (the second row)! The remaining computations and the results would be easy to follow.

The above clearly demonstrated the programming and computation advantages of using formulae (4-29) to (4-31). Table 9 also shows the potential improvement of the computation efficiency. Furthermore, all the intermediate results are fully reused, and, compared with the initial formula (4-12), where the computation complexity of w is more than linear (to the data size u), here the computation cost is (nearly) constant for a

relatively large data size. This is because, when the data size becomes large, the maximum pattern length α would be relatively stable, and hence the numbers of columns and rows of Table 9 become stable, and the computation cost becomes constant. In other words, our approach realizes a full scalability of the calculation of H_k and w. It would be even more significant if this approach could develop a way to reach the scalability of pattern mining in general, which is recognized as a critical issue in pattern mining [25]. Furthermore, when data size changes, for instance, some data tuples added in or deleted, Table 9 can be updated with only those columns and rows affected by the changed g_k (this becomes clearer with Corollary 4.3 stated in next subsection). Finally, the tabular approach also eliminates the exponent operations required in T_k (see 3-15), and Table 9 presents the relationship between formulae (4-14) and (4-17), where the vertical (column) summation represents the mechanism of (4-14), while the horizontal (row) summation represents that of (4-17).

Table 9. The recursive computation of H_ks

Θ_1	1	2	3	4	5	6	
K \ G_1	2	3	2	2	0	1	H_k
1	**2**	**6**	**6**	**8**	**0**	**6**	28
2		**3**	**6**	**12**	**0**	**15**	36
3			**2**	**8**	**0**	**20**	30
4				**2**	**0**	**15**	17
5					**0**	**6**	6
6						**1**	1
$g_k T_k$	2	9	14	30	0	63	w = 118

4.6 The odd and even length pattern frequencies

Additionally, there are interesting relations between the summations of the frequencies of odd length and even length patterns. Manipulating (4-11) such that all negative signed terms are moved to the right hand side:

$$\sum_{k=1}^{n/2} H_{2k-1} = \sum_{k=1}^{n/2} H_{2k} + \mathrm{u}, \tag{4-32}$$

where, the upper bound "n/2" of the left hand side should be replaced by (n+1)/2 if n is odd. We use H_{odd} and H_{even} to mean the raw frequencies of patterns of odd lengths and even lengths respectively:

$$H_{odd} = \sum_{k=1}^{n/2} H_{2k-1}, \text{ and } H_{even} = \sum_{k=1}^{n/2} H_{2k}. \tag{4-33}$$

Then, (4-32) becomes:

$$\mathrm{H_{odd}} = \mathrm{H_{even}} + \mathrm{u}. \tag{4-34}$$

Adding H_{odd} to both sides of (4-34), and noting $H_{odd} + H_{even} = \mathrm{w}$, we get:

$$2H_{odd} = H_{odd} + H_{even} + \mathrm{u} = \mathrm{w} + \mathrm{u}.$$

That is, for the sum of frequencies of all odd length patterns:

$$H_{odd} = (\mathrm{w} + \mathrm{u})/2. \tag{4-35}$$

And similarly, for the sum of frequencies of all even length patterns:

$$H_{even} = (\mathrm{w} - \mathrm{u})/2. \tag{4-36}$$

As measures of frequencies, H_{odd} and H_{even} each must be an integer. We have a proposition that guarantees it:

Proposition 4-4: w + u or w – u is always even numbered; and w is even or odd follows u is even or odd.

Proof: Using (4-12), and let $y = \sum_{j=1}^{j=u} 2^{b_j}$. Since $b_j = |T_j| > 0$, it follows that y is always even. Then, w + u = y – u + u = y, and w – u = y – u – u = y – 2u. In both cases, the results are even, and the first part of Proposition 4-4 is proved. At the same time, it is easy to see that, if u is even (or odd), w is then even (or odd); and the second part of the proposition is proved.

The above results can be seen from Table 9.

Following, we introduce significant laws governing all of the H_k distributions.

4.7 The H_k-curve and its properties

If we plot the H_k distribution (k, H_k) and link all of the H_k value points together as shown in Fig. 1, we get a curve of "raw collective frequency distribution" (or "H_k-curve", see Fig. 1). Interestingly, the curve can be expressed as a relation between every adjacent H_k and H_{k+1} as what follows:

Theorem 4-1: in pattern mining problem, the H_k-curve can be expressed as:

$$H_{k+1} = R_k \frac{\alpha - k}{k+1} H_k, \quad (0 < k < \alpha \leq n) \tag{4-37}$$

and, $H_k = 0$, $(k < 1$, or $k > \alpha)$ (4-37a)

where, α is the maximum length of the patterns; R_k is a "collective frequency reducer", or abbreviated as "reducer":

$$0 < R_k \leq 1, \quad (0 < k < \alpha). \tag{4-38}$$

To be more understandable, we prove the theorem qualitatively starting with a preliminary case where all the u subsample spaces (T_j) of Ω stored in the dataset DBo are of same size α, such that every collective frequency is multiplied by u:

$$H_k = u * C_\alpha^k . \tag{4-39}$$

According to (4-28), $C_\alpha^{k+1} = \frac{\alpha - k}{k+1} C_\alpha^k$, we then have: $H_{k+1} = \frac{\alpha - k}{k+1} H_k$. (4-40)

In this case, H_k possesses properties identical to C_α^k. For instance, H_k is symmetric since $C_\alpha^{\alpha-k} = C_\alpha^k$; when α is an even number, H_k is strictly quasi-concave and reaches its maximum value at $k = \frac{\alpha}{2}$; when α is an odd number, H_k gets its two maximum values at k $= \frac{\alpha - 1}{2}$ and $k = \frac{\alpha + 1}{2}$. The preliminary H_k model (4-40) is seen as the dashed curve in Fig. 1.

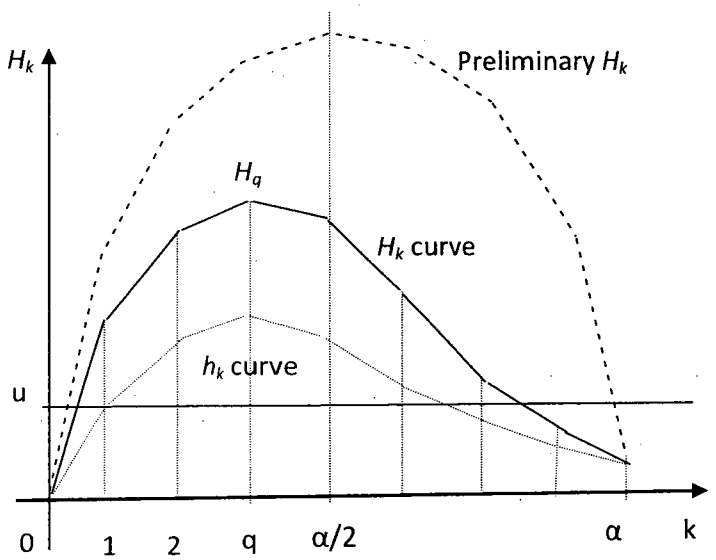

Fig. 1. The H_k and h_k curves

In reality, where many re-sampling subspaces are smaller than α, the simplified model (4-40) must be tuned using the following factors:

i). Different frequencies of individual patterns within a collective. Since we are interested in L_k as a collection

rather than its individual patterns, we consider H_k as a whole. In this way, the individual frequency differences within a collective do not matter.

ii). Inter-collective frequency reduction. Based on the Proposition 4-2, on average, the frequency of a pattern of length k + 1 will be less than its sub pattern of length k. We use R_k to mean such frequency reduction to adjust (4-40), and R_k has the following properties:

$$0 < R_k \leq 1, \quad (1 \leq k < \alpha)$$

which explains how (4-38) comes, and note obviously, $R_k \neq 0$ for $k \in [1, \alpha]$.

iii). Pattern contraction. This reflects the fact that, many subsample spaces are smaller than α, which means that not every pattern of length k will extend to pattern of length k+1. However, this phenomenon is the cause of point ii) above: a longer (super) pattern's frequency is less than that of its sub pattern, because a shorter pattern does not always extend to be a longer one. Hence, this observation reinforces that $R_k \leq 1$, but has no more effect on (4-40).

In summary, R_k defined in point ii) fully captures the tuning mechanism for (4-40). And, since R_k is a model of the proved Proposition 4-2, R_k is thus well established such that:

$H_{k+1} = R_k \frac{\alpha - k}{k+1} H_k$, as declared in the theorem.

The above delivers a qualitative proof of (4-37 and 4-38). To be more convincing, we prove (4-38) qualitatively below:

Following (4-26), $H_k = \sum_{i=k}^{i=\alpha} g_i C_i^k = G_k \, I_k \, \Theta_k$, we have

$$H_{k+1} = \sum_{i=k+1}^{i=a} g_i C_i^{k+1} = G_{k+1}\, I_{k+1}\, \Theta_{k+1}. \qquad (4\text{-}41)$$

On the other hand, by (4-28),

$$H_{k+1} = \sum_{i=k+1}^{i=a} g_i C_i^{k+1} = \sum_{i=k+1}^{i=a} \frac{i-k}{k+1} g_i C_i^k$$

$$= \frac{\alpha-k}{k+1} \sum_{i=k+1}^{i=a} g_i \frac{i-k}{\alpha-k} C_i^k$$

$$= \frac{\alpha-k}{k+1}\, G_{k+1}\, A_{k+1}\, \Theta'_k, \qquad (4\text{-}42)$$

where, Θ'_k is a sub-matrix of Θ_k without the first row (and first column); similarly G_{k+1} is a copy of G_k without the first element. A_{k+1} is a diagonal matrix of dimension (α-k) * (α-k), and termed an "adoptive matrix". Its main diagonal elements $a_{i,i} = \frac{i-k}{\alpha-k} < 1$ for all i, ($k < i \leq \alpha$).

Now, we define a diagonal matrix A'_k of dimension (α-k+1) * (α-k+1), with its first element $a_{1,1} = 0$, while its sub matrix of dimension (α-k) * (α-k) being exactly the same as A_{k+1}. To be more understandable, following are examples of the matrixes related to the running example with k = 3 (G_k and Θ_k can be referred from Table 9), then:

$$I_K = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}, \quad A_{k+1} = \begin{bmatrix} 1/3 & & \\ & 2/3 & \\ & & 1 \end{bmatrix}, \quad A'_k = \begin{bmatrix} 0 & & \\ & 1/3 & \\ & & 2/3 \\ & & & 1 \end{bmatrix}$$

With the above results, (4-42) can be reformulated as:

$$H_{k+1} = \frac{\alpha - k}{k+1} G_{k+1} A_{k+1} \Theta_k' = \frac{\alpha - k}{k+1} G_k A'_k \Theta_k. \quad (4\text{-}43)$$

Now (4-26) and (4-43) becomes comparable. Since ever element (except the last one) of A'_k is less than that of I_k, and if not all of the other elements except the last one of G_k is zero (note if the last element of G_k is zero, then the longest pattern length will be less than α!), then,

$$0 < G_k A'_k \Theta_k < G_k I_k \Theta_k. \quad (4\text{-}44)$$

Or, $$0 < G_k A'_k \Theta_k = R_k * G_k I_k \Theta_k = R_k * H_k \quad (4\text{-}44a)$$

where $0 < R_k < 1$ must be held to satisfy (4-44), and finally from (4-44a and 3-43),

$$H_{k+1} = \frac{\alpha - k}{k+1} G_k A'_k \Theta_k = R_k \frac{\alpha - k}{k+1} H_k,$$

which proves (4-37).

Now, an interesting question: is there any case where $R_k = 1$ would hold? The answer is yes, but the case is rare, and we have the following:

Corollary 4-1: the necessary and sufficient condition for $R_k = 1$ is that every element except the last one of G_k equals to zero.

Corollary 4-2: If $R_k = 1$, then all $R_s = 1$, where $k \leq s < \alpha$ (note R_k series is ended at $R_{\alpha-1}$).

The two corollaries above are easy to derive from the above proof and from (4-26) and (4-44a). If and only if the condition of Corollary 4-1 holds, both $G_k A'_k \Theta_k$ and $G_k I_k \Theta_k$ degrade to a scalar value $g_\alpha C_\alpha^k$ and the corollary becomes true. In fact, Corollary 4-2

comes directly from Corollary 4-1; and it can be simply noted as, if there are m zeros in the g_i distribution from i = α – 1 backwardly, then there are m ones in the right section of R_k series.

(4-38) is now formally proved. At the same time we see from these two corollaries, when $R_1 = 1$, which means all of the original data tuples are of same length α, and we get the preliminary case depicted in (4-40). This is where our qualitative and quantitative proofs converge.

After seeing the above interesting results, we have another property of R_k:

Corollary 4-3: The distribution of original data tuples of lengths less than k, g_j (j < k), does not have effect on R_s (s ≥ k).

Proof: It can be easily seen from (4-43) and A'$_k$.

This corollary converges again with the qualitative proof referred in the former part where R_k is tuned because not all (shorter) data tuples extend to longer ones, which is equal to say shorter data tuples have no effect on the frequencies of longer patterns as stated by corollary 4-3. This corollary can be stated alternatively that R_k is determined by all and only the g_j (j ≥ k),

The above now has fully proved (4-38) and (4-37), as well as other R_k properties, and hence Theorem 4-1 is fully proved. As follows, Theorem 4-2 below presents an important property of the H_k-curve.

Theorem 4-2 (H_k quasi-concavity theorem): If R_k is non-decreasing, then, the H_k-curve expressed in (4-37) is strictly quasi-concave downward over $0 < k \leq \alpha$, and it reaches its apex value at $k = q \leq \alpha/2$ (refer to Fig. 1).

"Quasi concavity" is used in real valued function study [22]. If a function f(**Z**) is strictly quasi concave within a domain E, then there exists a **Z*** (**Z*** $\in$ E) such that f(**Z**) is increasing for **Z** < **Z*** and f(**Z**) is decreasing for **Z** > **Z*** [22], where **Z** can be a vector of multidimensional variables. We use this concept not only for better understanding but also for formal applications of the properties of the H_k distributions. The only difference here is that the "quasi-concavity" property applies to discrete H_k values only.

If $R_k = 1$, the concerned problem is trivial according to corollaries 4-2 and 4-3, so we prove the quasi concavity property for $R_k < 1$ only.

Let us first look at the slope of the H_k-curve, $\Delta H_{k\cdot}/\Delta k$, with $\Delta k = 1$ which is the smallest interval:

$$\Delta H_{k\cdot}/\Delta k = (H_{k+1} - H_k)/\Delta k = H_{k+1} - H_k$$

$$= (R_k \frac{\alpha - k}{k+1} - 1)\, H_{k\cdot} \qquad (4\text{-}45)$$

Since $H_{k\cdot} > 0$, the sign of the slope $\Delta H_{k\cdot}/\Delta k$ is determined by $R_k \frac{\alpha - k}{k+1} - 1$. We notice that,

i). $\frac{\alpha - k}{k+1}$ is a strictly decreasing function of k, since:

$$\Delta(\frac{\alpha-k}{k+1})/\Delta k = -\frac{\alpha+1}{(k+1)(k+2)} < 0. \qquad (4\text{-}46)$$

ii). Without the effect of R_k, ($\frac{\alpha-k}{k+1}$ – 1) will be positive initially and cause H_k to increase and reach its maximum value at k = $\frac{\alpha}{2}$ if α is even, or, k = $\frac{\alpha-1}{2}$ and k = $\frac{\alpha+1}{2}$ if α is odd, as stated in the proof of Theorem 4-1 such that ($\frac{\alpha-k}{k+1}$ – 1) = 0. After that ($\frac{\alpha-k}{k+1}$ – 1) becomes negative and leads to a decrease of H_k.

iii). With the effect of R_k, , at the early stage (k << n), $\frac{\alpha-k}{k+1}$ could dominate and keep $R_k\frac{\alpha-k}{k+1} > 1$. That means H_k will be increasing with k but at a reduced rate because $R_k < 1$. Consequently ($R_k\frac{\alpha-k}{k+1}$-1) → 0 at a point q such that H_k reaches its apex value H_q, but q would be no larger than $\frac{\alpha}{2}$ due to the reduction effect of R_k; and the H_q itself would become much smaller than that without the effect of R_k (as seen in Fig. 1). The condition that R_k is not decreasing as given in the theorem guarantee H_k is always increasing until q, although the increase rate is diminishing. Once the H_q has been reached, the slop factor ($R_k\frac{\alpha-k}{k+1}$ – 1) becomes negative and keeps decreasing with k increasing, this is because, although R_k is increasing (or not decreasing), the decreasing rate of $\frac{\alpha-k}{k+1}$ is more dominant. H_k thus keeps decreasing until k → α, and there is no chance for H_k to get another apex value H_q regardless of α being odd or even.

In summary, H_k has one and only one apex at q, and H_k is strictly increasing for k < q but strictly decreasing for k > q, with the given condition of the theorem. H_k is thus strictly quasi-concave, and the theorem is fully proved.

Note, when α is not very large, H_k might take the maximum value at k = 1, but this does not affect the soundness of the theorem. Another point to note is that, theorem 4-2 stated a sufficient condition of R_k to keep an H_k curve quasi concave, while following are the more precise description of R_k against this condition:

Corollary 4-4: If the R_k series is not decreasing, and q is the apex point of the H_k curve, then:

$$\frac{k+1}{\alpha-k} < R_k \leq 1 \text{ for } 0 < k < q \leq \alpha/2, \qquad (4\text{-}47)$$

and, $$\frac{q}{\alpha-q+1} < R_k \leq 1 \text{ for } q \leq k < \alpha. \qquad (4\text{-}47a)$$

Proof: Both formulae are based on the fact of non decrease of R_k and its upper boundary defined in (4-38). The former formulae can be derived from (4-37) directly; the latter one is obtained by applying k = q – 1 into (4-47) and taking R_{q-1} as the base of comparison with other R_k (with k ≥ q). These two formulae can also be verified from the data of Table 9.

Quasi concavity is a nice property for an H_k curve. At this point, a question may arise: could this property be typical? Or, would the condition of no-decreasing R_k hold in most frequent pattern mining applications? The following theorem answers it.

Theorem 4-3: For an ordinary g_k distribution, the R_k series is no decreasing, and, the smaller the k relative to α, the stronger the condition $R_{k+1} \geq R_k$.

The implication of this theorem is that, once a $R_{k+1} < R_k$ happens, it would reflect an abnormal g_k distribution such that either, at the right tail of the distribution a nearly longest transaction (referring g_k for k → α) is more frequent than a shorter one, or, at the left tail, there is a jump (a sharp increase or decrease) from g_k to g_j (j > k). Note that the requirement of "ordinary" g_k distribution means the distribution is denser around the middle of α, diminishing towards the two ends. However, this requirement is easy to meet. It does not require the classic N(μ, σ) normal distribution or β distribution. It even does not require the distribution of a single mode, and a scattered g_k distribution is allowed, as long as the extra mode does not appear in the right tail of the distribution.

This theorem can be proved through any of the basic H_k expression (4-17), the vector expression (4-23), or the matrix expression (4-26). However, in any expression mode, a fully formal proof of this theorem would be lengthy. Here below we present the framework of the proof through the basic expression (4-17).

Starting from (4-17), $H_k .= \sum_{i=k}^{i=a} g_i C_i^k$, then,

$$H_{k+1} = \sum_{i=k+1}^{i=a} g_i C_i^{k+1} = \sum_{i=k+1}^{i=a} \frac{i-k}{k+1} g_i C_i^k$$

$$= \frac{1}{k+1} \sum_{i=k+1}^{i=a} (i-k)\, g_i C_i^k \ ,$$

and, $H_{k+2} = \sum_{i=k+2}^{i=a} g_i C_i^{k+2} = \sum_{i=k+2}^{i=a} \frac{(i-k)(i-k-1)}{(k+1)(k+2)} g_i C_i^k .$

Then, $R_k = \frac{k+1}{\alpha-k} H_{k+1} / H_k = \frac{\sum_{i=k+1}^{\alpha} \frac{i-k}{\alpha-k} g_i C_i^k}{\sum_{i=k}^{\alpha} g_i C_i^k} = \frac{\frac{k+1}{\alpha-k} g_{k+1} + \sum_{i=k+2}^{\alpha} \frac{i-k}{\alpha-k} g_i C_i^k}{g_k + (k+1) g_{k+1} + \sum_{i=k+2}^{\alpha} g_i C_i^k}$

$$= \frac{\frac{k+1}{\alpha-k} g_{k+1} + X_{k+2}^1}{g_k + (k+1) g_{k+1} + Y_{k+2}^1}, \qquad (4\text{-}48)$$

where, $X_{k+2}^1 = \sum_{i=k+2}^{\alpha} \frac{i-k}{\alpha-k} g_i C_i^k,$ (4-48a)

and, $Y_{k+2}^1 = \sum_{i=k+2}^{\alpha} g_i C_i^k .$ (4-48b)

$$R_{k+1} = \frac{k+2}{\alpha-k-1} H_{k+2} / H_{k+1}$$

$$= \frac{\sum_{i=k+2}^{\alpha} (i-k) \frac{i-k-1}{\alpha-k-1} g_i C_i^k}{\sum_{i=k+1}^{\alpha} (i-k) g_i C_i^k} = \frac{\sum_{i=k+2}^{\alpha} (i-k) \frac{i-k-1}{\alpha-k-1} g_i C_i^k}{(k+1) g_{k+1} + \sum_{i=k+2}^{\alpha} (i-k) g_i C_i^k}$$

$$= \frac{X_{k+2}^2}{(k+1) g_{k+1} + Y_{k+2}^2}, \qquad (4\text{-}49)$$

where $X^2_{k+2} = \sum_{i=k+2}^{\alpha} (i-k)\frac{i-k-1}{\alpha-k-1} g_i C_i^k$, (4-49a)

and, $Y^2_{k+2} = \sum_{i=k+2}^{\alpha} (i-k) g_i C_i^k$. (4-49b)

Then, the general condition of $R_k < R_{k+1}$ is substituted by the following inequality:

$$\frac{\frac{k+1}{\alpha-k} g_{k+1} + X^1_{k+2}}{g_k + (k+1) g_{k+1} + Y^1_{k+2}} < \frac{X^2_{k+2}}{(k+1) g_{k+1} + Y^2_{k+2}}, \quad (4\text{-}50)$$

For simplicity, the following proof uses X and Y without superscripts and subscripts to mean either (4-48a, b) or (4-49a, b), since these Xs and Ys are linear combinations of the same $g_i C_i^k$ series.

The advantage of (4-50) is that it simplifies the problem and involves three terms only, g_k, g_{k+1}, and the remains g_i (i > k + 1) which are wrapped into the respective X and Y.

Note in the above formulae, the following always hold:

$i \le \alpha$, $k < \alpha - 1$, $k < i - 1$, and hence in general, $i - k < \alpha - k$; $i - k - 1 < \alpha - k - 1$; and

$$\frac{i-k}{\alpha-k} > \frac{i-k-1}{\alpha-k-1}; \quad (4\text{-}51)$$

$$X^1_{k+2} < Y^1_{k+2}, \text{ and } X^2_{k+2} < Y^2_{k+2}, \quad (4\text{-}51a)$$

and, $X^1_{k+2} < X^2_{k+2}$, and $Y^1_{k+2} < Y^2_{k+2}$. (4-51b)

(4-48), (4-49) and (4-51a) above explain again why R_k and $R_{k+1} < 1$. In (4-51), the difference of the two sides is not large in general. Furthermore, given k, the difference gets its maximum at the smallest possible i then diminishing with i increasing. For instance, let $\alpha = 6$ as given in the running example, and suppose $k = 2$, then, $\frac{i-k}{\alpha-k} - \frac{i-k-1}{\alpha-k-1}|_{i=4} = 1/6$, $\frac{i-k}{\alpha-k} - \frac{i-k-1}{\alpha-k-1}|_{i=5} = 1/12$; and in general, for any feasible k, $\lim_{i\to\alpha} \frac{i-k-1}{\alpha-k-1} = \lim_{i\to\alpha} \frac{i-k}{\alpha-k} = 1$. Consequently, for any $k < \alpha - 2$, we have:

$$X^1_{k+2}/Y^1_{k+2} > X^2_{k+2}/Y^2_{k+2}, \quad (4\text{-}51c)$$

and $\lim_{k\to\alpha-2} X^1_{k+2} = Y^1_{k+2}$, and $X^2_{k+2} = Y^2_{k+2}$; (4-51d)

$$\lim_{k\to\alpha-2} X^2_{k+2}/Y^2_{k+2} = X^1_{k+2}/Y^1_{k+2} = 1. \quad (4\text{-}51e)$$

Now, look back (4-50). $R_k = \dfrac{\frac{k+1}{\alpha-k} g_{k+1} + X^1_{k+2}}{g_k + (k+1) g_{k+1} + Y^1_{k+2}}$ can be taken as an expansion of X^1_{k+2}/Y^1_{k+2}, and $R_{k+1} = \dfrac{X^2_{k+2}}{(k+1) g_{k+1} + Y^2_{k+2}}$ an expansion of X^2_{k+2}/Y^2_{k+2}.

Comparing (4-50) and (4-51c), we see the expansions should reverse the relation of (4-51c). According to Theorem 4-3, this reversion is dominant in most of the applications. Following are the analysis on what the forces are to maintain or reverse the inequality $R_k \le R_{k+1}$.

Unlike (4-51), the "greater than" relation of (4-51c) is not monotonic against k but depends on the distribution of g_i involved in the two ratios X^2_{k+2} / Y^2_{k+2} and X^1_{k+2} / Y^1_{k+2}. If k is relatively large and close to α, then only few g_is would be involved and a relatively large g_i could cause a significant difference of the two ratios, which may lead to $R_{k+1} < R_k$. This is one reason why $R_{k+1} < R_k$ would be more sensitive at the right tail of the distribution. The other possible reason for $R_{k+1} < R$ is the relative effect of g_k and g_{k+1} expressed in (4-50). In general, if g_k and g_{k+1} are significantly smaller (e.g., zero valued) than the remaining $g_i s$, then the relation of (4-51c) may be maintained. In particular, if $g_{k+1} > g_k$ and the two X/Y ratios do not differ much, which would lead $R_{k+1} < R_k$. This is because the effect of $\frac{k+1}{\alpha - k}$ could be large for a large k. In summary, as k approaches α, if R_{k+1} is less than R_k, then g_i would be greater than g_k (i > k). This means that in the right tail longer transactions occur more frequently than shorter ones, and reflects an abnormal g_i distribution.

For a smaller k, the number of $g_i s$ involved in X and Y is increased, stretching from the right to the left tail of the distribution. Then, a single g_i could significantly influence the X/Y ratios only if g_i is comparatively very large and i is smallest possible (e.g., i = k + 2). Secondly, the factor $\frac{k+1}{\alpha - k}$ is decreasing quickly when k decreases compared with

the relatively small decrease from X^1_{k+2}/Y^1_{k+2} to X^2_{k+2}/Y^2_{k+2}. Thirdly, notice that, from (4-51b), $X^1_{k+2} < X^2_{k+2}$. These are the forces to support (4-50). In other words, to reverse the relation of (4-50) such that $R_{k+1} < R$ holds, one possibility is $g_j >> g_k$ (j > k, and typically $g_{k+1} >> g_k$), and reflects the other type of "abnormal" g_i distribution at the left tail of the g_i distributions as mentioned before. The theorem is then proved.

Table 10 demonstrates the above conclusions and hence Theorem 4-3. Here the second row refers to the original data of Table 1, which gives an increasing R_k series and a quasi concave H_k curve. Column 1 of the table indicates the value of k, for which the relation $R_{k+1} < R_k$ would occur. Column 2 lists the g_i distributions, and the bold numbers are the minimum g_i s that would cause an $R_{k+1} < R_k$, which illustrates clearly that the smaller the i, the more significant g_i is required. Column 3 lists the R_k series and the bold numbers are those in the inequalities of $R_{k+1} < R_k$. Column 4 gives the two ratios of X/Y corresponding to those R_k and R_{k+1}, and this column demonstrates the properties given from (4-51a) to (4-51e); meanwhile, row 3 shows the case of $R_{k+1} < R_k$ mainly due to X^1_{k+2}/Y^1_{k+2} being significantly larger than X^2_{k+2}/Y^2_{k+2}, while row 2 and 4 show the cases that $R_{k+1} < R_k$ is caused mainly by an abnormally large g_{k+1} over g_k, when the two X/Y ratios do not differ much (less than 3% in these examples). Column 5 is the corresponding H_k series, from which we can see how resilient the quasi concavity property is. Except the last row, all of the other cases result in strict quasi concave H_ks.

A basic reason for the observation is that the related R_k fluctuation ($R_{k+1} < R_k$) is not significant, as just noted that the abnormal g_i is the minimum (to alternate the relation $R_{k+1} > R_k$ into $R_{k+1} < R_k$). Only when such a g_i becomes even larger, and R_{k+1} becomes significantly smaller than R_k, could the corresponding H_k curve becomes non quasi concave. The last row of Table 10 provides a case of no quasi concave H_k curve for α = 10, from which we notice the extreme g_i distribution. Interested reader can verify that in this case, R_k fluctuates greatly.

Table 10. Demonstrations of the H_k and R_k properties

k	g_is	R_k series	X/Y ratios	H_k series
	2, 3, 2, 2, 0, 1	0.514, 0.625, 0.756, 0.882, 1		28, **36**, 30, 17, 6, 1
1	2, **50**, 2, 2, 0, 1	**0.272, 0.271,** 0.756, 0.882, 1	X^1_{k+2}/Y^1_{k+2} = 0.642 X^2_{k+2}/Y^2_{k+2} = 0.656	**78**, 61, 30, 17, 6, 1
1	2, 3, **12,** 2, 0, 1	**0.4551, 0.4545,** 0,567, 0.882, 1	X^1_{k+2}/Y^1_{k+2} = 0.494 X^2_{k+2}/Y^2_{k+2} = 0.455	58, **66**, 40, 17, 6, 1
3	2, 3, 2, 2, **3,** 1	0.614, 0.682, **0.711, 0.703, 0.667**	X^1_{k+2}/Y^1_{k+2} = 1 X^2_{k+2}/Y^2_{k+2} = 1	43, **66**, 60, 32, 9, 1
	50, 100, 300, 0, 0, 0, 0, 0, 6, 3			1234, **1351**, 1164, 1386, **1506**, 1134, 576, 189, 36, 3

In conclusion, we observe from Table 10 that R_k distribution is much less scattered than its underlying g_i distribution. In other words, R_k function would mostly transform a rather scattered g_i distribution into a nice monotonic R_k series. At the same time, we notice that a single data tuple can be seen as the simplest preliminary case (of u = 1),

which produces a quasi concave curve of frequencies of patterns of different lengths. Then the H_k curve is an aggregation of these individual quasi concave curves. In general, a summation of a set of quasi concave curves is not necessarily quasi concave, and how to organize such set of quasi concave curves into a single quasi concave curve is an interesting topic in many applications [22]. In this sense, R_k, is a nice solution.

Finally, we see that the quasi concavity property of H_k is more resilient than the monoitonicity of R_k, and would not be affected by some minor fluctuations or decreasing of R_k over a given g_i distribution. It means then, the monotonic condition of R_k as stated in (4-38) is stronger than required, and the deeper reason for it is that the effect of $\frac{\alpha - k}{k+1}$ is more dominant than R_k in (4-37). Considering all of these aspects together, quasi concavity property for H_k is very typical.

From the above quasi concavity theorem and its proof, we see that the different lengths of data tuples do not degrade but indeed improve the quasi-concavity of the left section of the H_k curve compared with the preliminary case (4-40). The reason is that, the shorter data tuples slow down the rapid frequency increase with k increasing in the preliminary case, which then causes the H_k curve to become more rounded. Then a natural question would arise: is it possible for an H_k curve to become not just quasi but genuine concave? The answer is given in the following theorem.

Theorem 4-4: An H_k-curve can be strictly concave downward within an interval E = [a, b], if the following condition holds:

$$R_{k+1} R_k \frac{\alpha-k}{k+1} \frac{\alpha-k-1}{k+2} - 2R_k \frac{\alpha-k}{k+1} + 1 < 0, \qquad (4\text{-}52)$$

where α is the maximum length of all the patterns, and,

the minimum a = 1, the maximum b = ½ * (α+2 + (α+2)^{½}). (4-53)

We prove the theorem starting from the definition of "concavity": if a function f(**z**) is strictly concave over an interval E, then for any three points $\mathbf{z}_1$, $\mathbf{z}_2$, $\mathbf{z}_3$ within E, such that $\mathbf{z}_2 = \lambda\mathbf{z}_1 + (1-\lambda)\mathbf{z}_3$, where $\lambda \in (0, 1)$, and **z** can be a vector of multidimensional variables, then the following relation holds [22]:

$$\lambda f(\mathbf{z}_1) + (1-\lambda) f(\mathbf{z}_3) < f(\mathbf{z}_2). \qquad (4\text{-}54)$$

Alternatively, set λ = ½, the necessary and sufficient condition for f(**z**) to be strictly concave is [22]:

$$½ (f(\mathbf{z}_1) + f(\mathbf{z}_3)) < f(\mathbf{z}_2). \qquad (4\text{-}54a)$$

where $\mathbf{z}_2$ is in the middle of $\mathbf{z}_1$ and $\mathbf{z}_3$: $\mathbf{z}_2 = ½(\mathbf{z}_1 + \mathbf{z}_3)$.

To facilitate the proof, we use (4-54a) and choose any three consecutive points k, k+1 and k+2 of the domain of the H_k-curve and check whether they satisfy (4-54a). In this case, λ = ½, and k+1 = ½ (k + (k+3)). Then the related H_k values must satisfy: ½ (H_k + H_{k+2}) < H_{k+1}. By (4-37), it means:

$$½ (H_k + R_{k+1} \frac{\alpha-k-1}{k+2} H_{k+1}) < R_k \frac{\alpha-k}{k+1} H_k,$$

Or, $\frac{1}{2}(H_k + R_{k+1} R_k \frac{\alpha-k}{k+1} \frac{\alpha-k-1}{k+2} H_k) < R_k \frac{\alpha-k}{k+1} H_k$.

Manipulating the above and removing H_k since $H_k > 0$, we get the condition for H_k concavity:

$$R_{k+1} R_k \frac{\alpha-k}{k+1} \frac{\alpha-k-1}{k+2} - 2R_k \frac{\alpha-k}{k+1} + 1 < 0, \qquad (4\text{-}52)$$

which is the necessary and sufficient condition specified in Theorem 4-4.

Now, we look at the maximal interval E = [a, b] over which (4-54a) could hold. According to the rationale that the first section of the H_k curve can be augmented to concavity is due to the effect of shorter data tuples such that R_k is less than 1 and the growth of H_k slows down before H_k reaches its apex value. After that, to maximally maintain the concavity section, R_k should keep as large as possible to prevent H_k from decreasing quickly. It means then that the right end of the concave interval will be the same in the preliminary case where all data tuples are of same length α, and R_k is in its maximum 1. We then first solve (4-52) in the preliminary case, where (4-52) becomes:

$$\frac{\alpha-k}{k+1} \frac{\alpha-k-1}{k+2} - 2\frac{\alpha-k}{k+1} + 1 < 0. \qquad (4\text{-}56)$$

Solution (of k) to the above inequality is:

$$s\ [=\text{ceiling}\ (\tfrac{1}{2} * (\alpha - 2 - (\alpha+2)^{1/2}))] < k < [\text{floor}\ (\tfrac{1}{2} * (\alpha - 2 + (\alpha+2)^{1/2}))] = t, \qquad (4\text{-}57)$$

where ceiling(y) is a minimum integer s ≥ y; and floor(y) is a maximum integer t ≤ y. Here, *s* is the left end but *t* is not the ultimate right end of the concavity section of

H_k curve in the preliminary case, since, based on the above formulations, if k = t is a solution to (4-56), then k+1 and k+2 will be included in the concave interval as well. That is:

$$b = t + 2 = ½ * (\alpha - 2 + (\alpha+2)^{½}) + 2 = ½ * (\alpha+2 + (\alpha+2)^{½}), \qquad (4\text{-}57a)$$

which is then the ultimate right end of the concave interval as declared in the second part of (4-53) of Theorem 4-4, since after that, there is no force to entail $R_k > 1$ to slowdown the decrease of the H_k curve and augment its right quasi concave tail into genuine concavity.

It is easy to find out from (4-57) and (4-57a), the two end points, *s* and *b*, are symmetric against α/2 (the middle of the g_i distribution), as we have introduced before, the preliminary H_k curve is symmetrical. This means that, in the preliminary case, only the middle section of the H_k curve is concave, but (for $\alpha > 4$) its right and left "tails" are quasi-concave only. Since s or b is an increasing function of α, then a larger α implies longer quasi-concave tails, but the middle concave interval decreases relatively against α. This is because, $(b - s) / \alpha = ((\alpha+2)^{½} + 2) / \alpha$ decreases against α. If α is large, for instance, α = 100, then in the preliminary case, $(b - s)/\alpha \approx 12\%$, a small portion.

Now, in a general case where the uniformed data tuple length is no longer held, and R_k can be as small as possible in the left section of the g_i distribution (as long as $R_k > 0$ holds). In other words, the left end of the interval can be stretched as left as possible, and ultimately s = 1. This then proves the first part of (4-53).

Continuing the above example (where $\alpha = 100$) in the general case, the concave interval could be increased from [s, b] to as large as [1, b]; and the percentage of the whole concavity section [1, b] is increased to $(b - 1)/\alpha = 56\%$, a big increase! A more concrete example is given here with $\alpha = 10$, and g_i distributions = {2, 3, 52, 10, 8, 6, 5, 3, 2, 3}. In this case, the related H_k curve gets its maximum concavity interval [1, 7], against [3, 7] in the preliminary case.

Theorem 4-4 and its implications have now been fully proved. However, an important notice here is that the maximum concave interval is only possible and may not be seen often in empirical cases. This is because, as implied above, the left extended concavity interval corresponds to a left skewed g_i distribution. At the same time, the concavity interval can otherwise be smaller than that in the preliminary case. How large the concavity interval could be obtained from an application is determined by the underlying g_i distributions. The value of Theorem 4-4 and the previous ones is in that they formally describe the properties of the raw frequency distributions, the shapes of the H_k curves, and their relations with the underlying g_i distributions.

In this chapter, we have introduced and proved the H_k concave and quasi-concave laws, and also delivered a number of accompanied interesting implications and properties embodied in the corollaries. Hereafter we will not distinguish quasi concavity and concavity unless required, since a concavity implies quasi concavity, though not vice versa. The H_k quasi concavity property would be applied to many pattern mining

applications, just as those quasi-concave functions are widely used in modern economics, operation research and other related domains. For instance, if an H_k is known, then H_{k-1} and H_{k+1} could be predicted as well, according to the H_k expressions. This property and other properties, such as the relation between odd and even length pattern frequencies, could be taken as check points of the correctness of a mining algorithm. They can also be used as a reference for the determination of the boundaries of supports or frequentness of patterns in concise representations [4, 28, 34, 35], or for the estimates of the number of patterns of different levels [24]. However, our emphasis is not on the application of the raw pattern based studies. The more significant use of the H_k concavity property is on how to refine the raw pattern frequency distributions. For this, we present below a prime property of the quasi-concave function. Other properties of it and their applications shall be studied in future work.

The prime property of a quasi-concave curve f(**Z**) is that, in the real value situation, the domain of **Z** covered by f(**Z**) is a convex upper contour set, where **Z** can be a vector of multidimensional variables [22]. Here, we can imagine that the domain **Z** as a *hyper polyhedron*; and the convex domain means the polyhedron is dense – without internal hole and the surfaces of it are convex. Intuitively, given a straight line between two separate points $\mathbf{Z}_i$ and $\mathbf{Z}_j$ $(i \neq j)$ within the convex polyhedron, all of the points of the line will be within the polyhedron [22]. Since each $\mathbf{Z}_i$ is a multidimensional hyper point, we term this domain density as "dense by point". The domain covered by a quasi-concave H_k curve is the integer k, from 1 to α; and this domain is dense meaning no integer m ($1 \leq m \leq \alpha$) is not been covered by the H_k curve. This is an easy but superficial understanding

of the prime property of the H_k concavity. There are two important things to note. Firstly, it is not appropriate to consider the discrete integer domain density the same as the continuous density as in the real value situation. This is because, theoretically, for a given domain (interval) [a, b], the number of points within it is infinitive in the continuous case but finite in the integer case; and intuitively, the discrete integers within the domain could not form a continuous line. Secondly and more importantly, if we took k as a one-dimension integer only, then we totally ignored the semantics of k. Under the H_k regime, each k value represents a collective of patterns of length k. That is the insight, and k implies a function to collect all of such length patterns:

$$k = k(\mathbf{Z}).$$

Then, $H_k = H(k) = H(k(\mathbf{Z})) = f(\mathbf{Z})$.

In this case, since each k represents a collective of hyper points **Z**, then reasonably we can take k as a "hyper plane", and the domain density under the concave H_k curve can be termed as "dense by (hyper) plane" or "dense by collective", compared with the "dense by point" in the conventional real valued case. To understand the "dense by plane", we take a simple example of a pattern domain **Z** of elements (values) A, B, C, which can be illustrated as a 3-D polyhedron graphically as plotted in Fig. 2, where each of the three values is a hyper point represented by a unit vector in the polyhedron. And, the patterns generated from them can also be plotted as vertices (points) of the polyhedron. Then, connecting the patterns of same lengths k forms (hyper) planes L_1, L_2, and L_3 respectively, where L_3 shrinks into a (hyper) point in this simple example. In this way, the concept of hyper plane in this space corresponds to a collection of patterns. Thus we can analogue "dense by plane" to "dense by point": for a given straight line connecting

two non-adjacent hyper planes, L_1 and L_3, for instance, the line must pierce through all the hyper planes (here only L_2) sandwiched between these two planes (L_1 and L_3). This then gives the semantics of the quasi-concavity of H_k curve, under which no more patterns could be enclosed into the domain covered by the H_k curve. On the other hand,

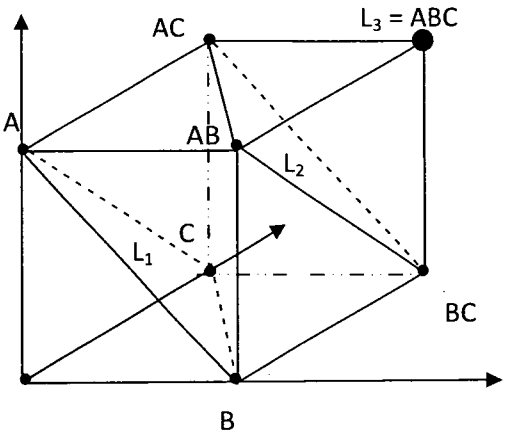

Fig. 2. A convex domain of H_k

such domain can be too dense in an application. For instance, when a new data tuple is added, then the hyper planes in Fig. 2 could overlap. This corresponds to the already mentioned case where the full enumeration based pattern generation produces excessive number of patterns. Considering these phenomena together, we term the domain convexity (density) a **saturation** property in lieu of H_k concavity for the full enumeration pattern generation regime.

In the following chapter, we will see how the quasi convexity (saturation) property could be used and maintained in pattern frequency adjustment.

Chapter 5. The adjusted pattern frequency distributions

The previous chapter has introduced a set of interesting properties governing raw pattern frequency distributions under full enumeration regime. A merit of complete enumeration approach is the fullness of patterns produced, which is reflected by the saturation (concavity) property of the associated H_k curve. However, as discussed in section 2.3, the full enumeration means meaningless patterns will be generated with other drawbacks, notably the overfitting and underfitting issues, bias for generated vs. original patterns, and favoring shorter against longer patterns. Ideally, we want to overcome these drawbacks and at the same time keep the advantage of the full enumeration approach. To do so, we first look at why full enumeration mode has been used by conventional mining approaches, and then we present the theory on how to adjust this approach.

5.1 The assumptions underlying the full enumeration mode

In a sense, the drawbacks induced by full enumeration mode and examined in Section 2.3 are the surface problems, and one may ask, why the full enumeration mode is used, since it has so many drawbacks? We can trace these problems to the following underlying assumptions that researchers have adopted but not formally reported in the relevant literature:

1) The assumption of full repeatable random sampling. From statistical point of view, the concerned pattern generation is exactly a *re-sampling* over every original data tuple. The full enumeration then exactly corresponds to the full repeatability assumption of (re)sampling, as explained below.
2) The assumption of uniform probability distribution of the patterns to be generated.

3) The assumption of every generated pattern is effective, or in other words, no pattern generated is meaningless or a "random walk".

Only with the identification of the above assumptions could we explain why and how a miner generates the patterns. That is because: the miner could not consider an original tuple as a pattern. For instance, without prior knowledge, a miner could not assume that the first tuple of the DBo in Table 1, ($V_1V_4V_7$), is a true pattern, but can only postulate that combinations of the three elements are equally possibly patterns. Here the miner does not only take the uniform probability distribution assumption, but also assume the elements can be repetitively drawn to form different patterns with each other. In this sense, the full repeatability is the base of full enumeration.

The question is then whether this full repeatability assumption would hold. In a sense, this assumption could be justified for the original data tuples. For instance, in a market-basket problem, every element can be drawn repeatedly by customers to form transactions (data tuples), since every element (product) can be always refilled by the supplier. Such transactions are the originally sampled events. In other words, if each original tuple is taken to be a single observation, then we can take it as an outcome of pattern generation based on full repetition (or full replacement) assumption. Regarding a sampled event (tuple), e.g. $V_1V_4V_7$, the full repeatability of re-sampling may not be justifiable: After V_1 and V_4 had been drawn to form a pattern, for instance, would V_1 be used again to form another pattern with or without other element(s) together from that tuple, since V_1 is already used? There is no predetermined answer regarding a single data tuple, since in pattern mining, only the presence of an item matters, not its quantity.

At this point, a question may arise, since we have stated in Section 2.2 that the full enumeration mode over every single tuple is equivalent to the full enumeration from the whole element space, but here we do not suppose full enumeration of pattern over a single tuple is justifiable, is it a contradiction in the context? The answer is, although full repeatability is the base of full enumeration, it does not necessarily imply that the repeatable draws will produce every possible combination of the elements, let alone every possible pattern. This can be verified practically, even in a market-basket problem, the number of patterns is normally far less than the power set. That is, the above paragraphs do not mean full pattern enumeration from the whole element space is a natural consequence of the assumed full replacement sampling mode. The logic is clear then, full pattern enumeration from a single data tuple is not justifiable, but it is fundamentally exercised by conventional mining approaches.

It is because of the full repetitive (re)sampling assumption and because of the equalization of this assumption with the full pattern enumeration, the drawbacks of 2.3.7 through 2.3.9 listed in section 2.3 take place. Since, for instance, according to combinatorics described in Chapter 4, the frequencies of shorter patterns will certainly be larger than the longer ones, and hence causes drawback 2.3.9.

Similarly, the third assumption is reflected from the fact that the occurrence S_Z of a pattern Z is incremented by every generation of Z without any deduction or adjustment in conventional pattern mining approaches. The deeper reason for this is that, with the absence of domain knowledge, a miner could not assume a single data tuple is a random

walk over the whole dataset, or an element is a random walk towards a single data tuple. Subsequently, the patterns are generated from that tuple and they can only be taken to be equally meaningful. From this we notice how a contradictory consequence is created in pattern mining: in the generating process every pattern is meaningful but the process ends up with too many meaningless patterns and hence an overfitting!

Now, a question may arise: whether approaches, such as the constrained, the concise representation approaches, including the "closed" and the "maximal" approaches, would solve the above full enumeration related problems, since, as reported, they produce a set of greatly reduced number of patterns [2, 25]? Our answer is negative. Basically, the mission of these approaches is not to address and resolve these problems. Respectively, the constrained approach is mechanical and ad hoc, since it does not exactly "reduce" the mining result set but rather, "take over" only a subset of the result satisfying the constraints, and the constraints are application/user determined. Here the takeover means that, even though the delivered result set is smaller, the listed problems are delivered without a radical remedy. The concise representation approaches, as the name implied, do not exactly reduce the mining result set either, but just use a subset to "represent" the whole result set. The only reduction is the memory space to store a subset of the result instead of the whole. The computation cost can be reduced and mining efficiency be improved [34, 35, 36], but post work is needed to get an exact pattern and its frequentness. A more subtle issue is that, since the concise approaches do not necessary deliver the substantiated patterns [35] to the user but just wait and answer the user's query as to a pattern is frequent through the mining program, the user may not be alerted

of some surprising patterns! In general, this mode may not be very plausible to the user, who expects to get full knowledge of the mining results from the miner instead of querying the patterns that the user may not know yet. Finally, a common evidence of no pattern reduction from these approaches is that the pattern frequentness, S_z/u, as defined in (2-1), is commonly used by all of the approaches mentioned above, and it does not matter how small the result set is delivered, the included individual pattern frequentness is the same as that in the complete result set.

5.2 The principle of pattern frequency adjustment

The above section reveals a methodological weakness of the full enumeration based pattern generation – it is single data tuple based, since the resulted pattern set is just a simple collection of the separated generations over every original data tuples. Such tuple based generation could only follow those underlying assumptions listed in section 5.1 even though they are not very plausible, since no better assumptions could be adopted. This methodological weakness then leads to a contradictory consequence of the generation, in the sense that every combination generated from each data tuple is equally taken to produce an effective pattern, but their assemble ends up with too many meaningless patterns. What we need then is how to organically adjust the tuple based pattern generation with the information embodied in the dataset in whole, such that the drawbacks identified in section 2.3 could be resolved, and a reduction of the meaningless patterns could be pursued before domain specific constraints have been imposed. We term this kind of reduction an "unconditional pattern de-sampling", and the reduction is realizable based on the following two observations.

Observation 5-1: For a given set T_i of m (m > 1) elements over the original dataset (e.g., the DBo of Table 1), if T_i in whole is not a pattern, then at least one of its elements is a random walk to T_i.

Observation 5-2: For the same T_i in observation 5-1, if T_i in whole is a pattern, then at least one of its immediate proper sub patterns (of m – 1 elements) must be redundant, and hence meaningless to generate.

Observation 5-1 is philosophically unarguable; note that an element is a random walk to a single data tuple does not necessarily mean it is a random walk towards the whole dataset. Observation 5-2 is also unarguable based on remark 2-1. For instance, if T_i = {A, B, C} is a pattern that means the three elements present steadily in the dataset, and if AB and AC also present steadily, then BC must present steadily and hence be a pattern without a generation. Indeed, in this particular example if we know any one of the length-2 combinations is a pattern, then we know the other two are. That is, two of them are redundant. This is why the phrase "at least" is used in observation 5-2. Furthermore, the first observation is even stronger than the second one in the implication that we can safely de-sample at least one immediate sub pattern from an original element set T_i.

The problem is, regarding a single T_i = {A, B, C}, we do not know which of the three length-2 patterns is (are) meaningless to generate and should be de-sampled. Following elaborates our approach.

Since a reduction of pattern generation (a pattern de-sampling) means a decrement of the sum of the pattern frequencies, our approach to realize the pattern de-sampling is then

through the adjustment of the frequency distribution based on a principle as manifested from the above observations:

Principle of pattern frequency adjustment: the accumulated frequency of the immediate proper sub-patterns generated from a set T_i of m elements should be safely reduced by the frequency of T_i.

Since T_i represents m elements, where m is any feasible number ($1 \leq m \leq \alpha$), repeatedly applying this adjustment principle to all pattern levels from α to 1, it is easy to see the frequency of patterns of length k should be affected by that of all of their super patterns. This is well consistent with the spirit of the inclusion-exclusion principle stated in Section 4.3, where we have noticed that the determination of the total frequency of length-1 patterns involves that of patterns of length > 1:

$$\sum_{k=1}^{n} (-1)^{k-1} H_k = u, \qquad \text{(refer to 4-11)}$$

This inclusion-exclusion principle then can be naturally extended to represent the above adjustment principle in whole from level 1 to α, formalized as the following proposition:

Proposition 5-1: Extended implications of the "inclusion-exclusion principle":

i). The raw frequencies of patterns of length l must be adjusted (reduced) by that of their super-patterns of length > l.

ii). Some patterns might be de-sampled.

iii). The degree of pattern de-sampling and frequency reduction are defined implicitly by (4-11).

iv). The adjustments are to be done in an alternating mode.

Elaboration of the above proposition is as follows:

Recall that the frequency of a pattern is equal to the number of intersected element "I_c" held by a pattern (refer to Proposition 4-1), then I_c counted by a super pattern should not be recounted by its sub-patterns. This is the essence of the proposition as manifested in part i). Part ii) is a corollary of part i). When a sub-pattern Z and its super-pattern hold the same I_c, then by reduction, Z's frequency will become zero, which means Z get de-sampled. This is an important mechanism, and it allows the expected de-sampling. Part iii) states that the pattern de-sampling and frequency reduction must be done properly, neither less nor more than the formula requires. We will see what is exactly required soon. The reason for part iv) is the same as given in section 4.2.

The next section introduces the mathematical model to adjust the collective pattern frequencies based on the above adjustment principle and proposition 5-1.

5.3 The adjusted H_k, h_k

Definition 4-1: The "adjusted collective frequency" (of all patterns of length k) is:

$$h_k = H_k - \sum_{j=k+1}^{n} (-1)^{j-k-1} H_j , \qquad (k \in [1, \alpha]) \qquad (5\text{-}1)$$

which can be simplified to:

$$h_k = H_k - h_{k+1}, \qquad (k \in [1, \alpha]) \qquad (5\text{-}2)$$

with $h_k = 0.$ $(k < 1, \text{ or } k > \alpha)$ (5-3)

To be a measure of frequency, h_k must be nonnegative. In fact, we have the following theorem to guarantee it:

Theorem 5-1: If the underlying H_k–curve is strictly quasi concave, then h_k defined in (5-1, or 5-2) is always positive; and it fits the boundary conditions at k = 1 and k = α seamlessly.

Following we prove the theorem by induction. Since H_k is strictly quasi-concave, we examine the problem in two intervals, [1, q] and (q, α] respectively, where q is the maximum point of H_k and α is the longest pattern length.

For k ∈ [1, q], take the initial case k = 1, we see (5-1) is exactly (4-11) itself, thus

$$h_1 = \mathrm{u}. \tag{5-4}$$

It means the theorem holds at k = 1, since $h_1 > 0$. At the same time, (5-4) tells that (5-1) or (5-2) automatically fits the boundary condition at k = 1.

At k = 2 and by (5-2),

$$h_2 = H_1 - h_1 = H_1 - \mathrm{u} > 0, \tag{5-5}$$

which is obvious, since $H_1 = \sum_u |T_i| > u$ (unless $|T_i| \leq 1$ (i = 1, 2, ...u), but it is too trivial a case, and more importantly the H_k curve shrinks into a point H_1 only in this case) , where T_i is the i^{th} tuple of DBo.

Now, suppose the theorem holds at k= t (1 < t < q), such that $h_t > 0$, we check if $h_{t+1} > 0$ would still hold.

Since the theorem holds at k = t, it thus holds at k = t – 1 as well, that is, $h_{t-1} > 0$; and notice $H_t - H_{t-1} > 0$ for t ∈ [1, q] (because of the H_k concavity as given). Then, by (5-2), it means:

$$h_{t+1} = H_t - h_t = H_t - (H_{t-1} - h_{t-1}) = (H_t - H_{t-1}) + h_{t-1} > 0, \qquad (5\text{-}6)$$

and the theorem is proved for k ∈ [1, q].

For k ∈ (q, α], we start at k = α as the initial case and prove the theorem in a reverse direction. In this case, note that there is no pattern of length k > α. Thus, $h_t = H_t = F(\emptyset) = 0$ (for t > α, and Ø is the empty set). It thus means,

$$h_a = H_a - h_{a+1} = H_a > 0. \qquad (5\text{-}7)$$

It again demonstrates how (5-1) or (5-2) nicely fit the boundary condition at k = α, and $h_k > 0$ at the initial case k = α, such that the theorem holds.

At k = α-1, and notice that by (4-37),

$$H_{\alpha-1} = \left(\frac{1}{R_k}\frac{k+1}{\alpha-k}H_{k+1}\right)\Big|_{k=\alpha-1} = (\alpha / R_{\alpha-1})\, H_\alpha, \quad \text{then:}$$

$$h_{\alpha-1} = H_{\alpha-1} - h_\alpha = H_{\alpha-1} - H_\alpha = (\alpha / R_{\alpha-1} - 1)\, H_\alpha > 0, \qquad (5\text{-}8)$$

which results from $\alpha > 1$; $0 < R_{\alpha-1} \leq 1$; and $H_\alpha > 0$.

Now, suppose the theorem holds at k = t (q < t < α), such that $h_t > 0$, we check if $h_{t-1} > 0$ would still hold.

Since the theorem holds at k = t, it thus holds at k = t + 1 as well, that is, $h_{t+1} > 0$; and notice that $H_{t-1} - H_t > 0$ for t ∈ (q, α]. Then, by (5-2), it means:

$$h_{t-1} = H_{t-1} - h_t = H_{t-1} - (H_t - h_{t+1}) = (H_{t-1} - H_t) + h_{t+1} > 0. \qquad (5\text{-}9)$$

That is, the theorem holds for $k \in (q, \alpha]$ as well, hence the theorem is fully proved.

Theorem 5-1 justifies the necessary conditions for h_k to be a frequency function. The following points justify the sufficient conditions for h_k to be the expected adjustments covering the issues more than that addressed in from subsection 2.3.7 through 2.3.9:

a) h_k is truly succinct, thus overfitting is suppressed. This can be seen clearly from (5-2) which indicates that the sum of frequencies of shorter patterns is subtracted by that of longer patterns, and hence there is no double counted inter-collective frequencies in the accumulative frequency.

b) h_k well addresses the combinational effects in pattern formation, and it takes care of longer patterns that should not be less weighted. In particular, as shown by (5-7), the longest patterns' raw frequencies would not be reduced.

c) h_k also takes care of those original patterns from the original datasets as well, since the longer or especially the longest patterns are the original ones. This is a noticeable point that all the original patterns could be maintained in the final results set. Contrarily, they cannot all be recovered from conventional mining result set.

d) On the average, the adjustment is relatively evenly distributed over different collectives. That is, the collectives of larger number of frequencies would be adjusted more than those of smaller frequencies.

e) More importantly, the adjustment would not excessively de-sample the patterns generated from the full enumeration, since the adjustment maintains the quasi

concavity property. Below we prove this assertion first and the other statements will become clearer.

5.4 The h_k-curve and its properties

Similar to the H_k-curve, by connecting all of the h_k values together, we get an h_k-curve, and its quasi-concavity property can be presented by a theorem below:

Theorem 5-2 (h_k quasi-concavity theorem): If $\sum_u |T_i| > 2u$, where T_i is the i^{th} tuple of the original database DBo, then if the corresponding H_k-curve is strictly quasi-concave downward, then the h_k-curve is strictly quasi concave downward as well, and it reaches its apex value at k = q', where q' = q or q+1; and q is the apex point of the corresponding H_k-curve.

Note the condition of $\sum_u |T_i| > 2u$, or equally average $|T_i| > 2$, is symbolic only, since any complex mining problem would satisfy it. Another implication of this condition is the maximum pattern length $\alpha = \max(|T_i|) \geq 3$.

To prove the above theorem, we first look at the following relations:

$$h_{k+1} - h_k = (H_{k+1} - h_{k+2}) - (H_k - h_{k+1}),$$

or, $$(h_{k+1} - h_k) + (h_{k+2} - h_{k+1}) = H_{k+1} - H_k, \quad (5\text{-}10)$$

or, $$h_{k+2} - h_k = H_{k+1} - H_k. \quad (5\text{-}10a)$$

On the other hand, we have:

$$h_{k+1} - h_k = (H_k - h_k) - (H_{k-1} - h_{k-1}),$$

or, $$h_{k+1} - h_{k-1} = H_k - H_{k-1}. \quad (5\text{-}11)$$

We notice that, for k $\in$ [1, q), based on theorem 4-2, $H_k - H_{k-1} > 0$, (5-10a) and (5-11) are consistent and we get:

$h_{k+2} > h_k$, (from (5-10a) (Case I)

and, $h_{k+1} > h_{k-1}$, (from (5-11) (Case II)

and refer to (5-5), and notice $H_1 = \sum_u |T_i| > 2u$, we have:

$$h_2 > h_1, \tag{5-5a}$$

Since k can be any number within [1, q), the only way to have (5-5a) and both cases I and II to always hold within this interval is:

$$h_{k+2} > h_{k+1} > h_k > h_{k-1}. \tag{5-12}$$

Similarly, for k $\in$ (q, α], $H_{k+1} - H_k < 0$, (5-10a) and (5-11) are consistent; shifting back k by 1, we get:

$h_{k+1} < h_{k-1}$, (Case I')

and, $h_k < h_{k-2}$, (Case II')

and, refer to (5-8): $h_{\alpha-1} = H_{\alpha-1} - h_\alpha = (\alpha / R_{\alpha-1} - 1)\, h_\alpha$; and notice $\alpha \geq 3$, $0 < R_{\alpha-1} \leq 1$, then:

$$h_{\alpha-1} > h_\alpha, \tag{5-8a}$$

That is, within (q, α], the only way to always maintain (5-8a) and the cases I' and II' is:

$$h_{k+1} < h_k < h_{k-1} < h_{k-2}. \tag{5-12a}$$

However, at point q, (5-10a) and (5-11) represent things differently as follows:

$h_{q+2} < h_q$, (from (5-10a) (5-13)

and, $h_{q+1} > h_{q-1}$. (from (5-11) (5-13a)

Then, the relation between h_q and h_{q+1} is not fixed and can only be case determined. In other words, the apex value of h_k in an application can be reached at either q or q+1, and we simply refer to it as q'. Here, we do not examine whether $h_q = h_{q+1}$ would happen, since, no other integer exists between q and q+1, then even if $h_q = h_{q+1}$ takes place, it does not affect the quasi concavity property.

We can now conclude that, h_k-curve reaches its apex value at q', and it is strictly increasing within [1, q'] (based on (5-12)) and strictly decreasing within [q', α] (based on (5-12a)), h_k-curve is thus strictly quasi-concave, and the theorem is fully proved.

Since h_k-curve keeps the quasi concavity property, it means under the proposed adjustment regime, sufficient number of patterns is still maintained. However, the degree of quasi concavity has been reduced from that of the H_k-curve. Accordingly, we use **sufficiency** instead of "saturation" to mean the h_k concavity.

The h_k-curve is shown in Fig. 1, which depicts how h_k compresses H_k, as h_k curve is fully underneath the H_k curve.

From the above proofs, and assuming the condition of theorem 5-2 holds, we can induce other implications as follows:

The calculus function of h_k: From equation (5-10):

$$(h_{k+1} - h_k) + (h_{k+2} - h_{k+1}) = H_{k+1} - H_k,$$

or, $\Delta h_k + \Delta h_{k+1} = \Delta H_k,$ (5-10b)

or, $\Delta h_k / \Delta H_k + \Delta h_{k+1} / \Delta H_k = 1.$ (5-10c)

(5-10c) is taken to be the *calculus function* of h_k over H_k.

Corollary 5-1: If an H_k-curve gets its apex value at k = 1, then the related h_k-curve will definitely reaches its apex value at k = 2.

This is obvious, since $h_2 > h_1$ is always true (refer to (5-5a)) and q' = q + 1 applies.

Corollary 5-2: The difference function between the H_k-curve and the h_k-curve is also quasi-concave.

In general, a difference of two quasi-concave functions may not necessarily be quasi-concave. Corollary 5-2 then represents a special "quasi-concavity invariant" property of the difference function between the H_k and the h_k curves. Indeed, the proof of this corollary is rather straightforward: the difference function is the shifted h_k-curve itself, since from (5-2):

$$H_k - h_k = h_{k+1}. \qquad (5\text{-}14)$$

This corollary and (5-14) indicates that the larger the H_k, the larger the adjustment. In other words, the adjustments are relatively evenly distributed over different collectives and correspond to point d) claimed in the end of subsection 5.2. At the same time, it reflects point b) that the adjustments correct the bias for shorter patterns. In this regard, following corollary depicts more precisely.

Corollary 5-3: The adjustments correct and redistribute the frequencies from shorter patterns towards longer ones, such that:

$$h_k < \tfrac{1}{2} H_k, \quad k \in [1, q') \tag{5-15}$$

$$h_k > \tfrac{1}{2} H_k, \quad k \in (q', \alpha] \tag{5-15b}$$

and, $$h_{q'} \approx \tfrac{1}{2} H_q. \tag{5-15c}$$

Proof: According to the h_k quasi-concavity property and for $k \in [1, q')$, $h_k < h_{k+1}$, then by (5-14), $H_k = h_k + h_{k+1}$, it means $H_k > 2h_k$. (5-15) is then proved. Similarly we can prove (5-15b). And (5-15c) is a natural consequence of the former two.

In addition to the above, in the case of $q' = q + 1$, the redistribution toward longer patterns is even obvious because of the shifted apex point from H_k to h_k. Even without the apex point shifting, the relation $h_{q+1} > h_{q-1}$ as seen in (5-13a) is generally held, which is another sign of the redistribution and a characteristic of the h_k -curve.

Corollary 5-3 then formally demonstrates that h_k realizes the desired correction of under evaluation of longer patterns than that of shorter ones by conventional mining approaches, and h_k thus takes care of combinational effects in pattern formation as addressed in subsection 2.3 and the end of subsection 5.2. Corollary 5-3 also implies that about half of the raw frequencies would be squeezed accumulatively. This is to be seen formally in the next subsection.

5.5 The aggregative relations between the H_k and the h_k measures

From (5-2) to (5-6), we have:

$$h_1 = H_1 - h_2$$

$$h_2 = H_2 - h_3$$

...

$$h_{a-1} = H_{a-1} - h_a$$

$$h_a = H_a$$

and, $-\mathrm{u} = \quad -h_1$ (refer to (5-4))

Summarize the above equations together, we get:

$$\sum_{k=1}^{a} h_k - \mathrm{u} = \sum_{k=1}^{a} H_k - \sum_{k=1}^{a} h_k ,$$

or, $$2\sum_{k=1}^{a} h_k - \mathrm{u} = \sum_{k=1}^{a} H_k . \qquad (5\text{-}16)$$

Note that $H_k = h_k = 0$ for k > a, thus: $\sum_{k=1}^{a} H_k = \sum_{k=1}^{k=n} H_k = \mathrm{w}$ (the raw accumulative frequency).

Set the "adjusted accumulative frequency" as $\mathrm{w_a}$, then from (5-16) we get:

$$\mathrm{w_a} = \sum_{k=1}^{a} h_k = (\mathrm{w} + \mathrm{u})/2, \qquad (5\text{-}17)$$

which is a coincidence with (4-35), and hence:

$$\mathrm{w_a} = \sum_{k=1}^{a} h_k = H_{odd.} \qquad (5\text{-}18)$$

(5-17) tells how the adjustment compresses the number of accumulated raw frequencies. Since normally u << w, (5-17) implies $\mathrm{w_a} \approx \mathrm{w}/2$. We term it a "law of half".

Meanwhile, since every increment of frequency of a pattern implies a generation of that pattern from a data tuple, then the number of patterns deducted from the adjustment

process would be proportional to the number of frequencies reduced if such a reduction leads a pattern de-sampled. That is, the estimate of the lower boundary of the total number of patterns after the adjustment process would follow the law of half. This is another important implication in an application.

The above relations thus enable the adjusted accumulative frequency to be easily predetermined. This is what many approaches pursue but no final finding has been reported to our knowledge.

5.6 The concavity of h_k-curve

Similar to the H_k-curve, after we have proved the h_k quasi concavity and other accompanied properties, we have the h_k genuine concavity property as well.

Theorem 5-3: If an H_k-curve is strict concave downward over an interval E = [a, b], then the corresponding h_k-curve would maintain the concavity over the interval $[a_1, b_1]$, subject to the only condition of:

$$(H_{k-1} + H_k) > 2(h_{k-1} + h_{k+1}), \qquad (5\text{-}19)$$

where a_1 and $b_1$can be either greater or less than a, b respectively.

Proof: According to the definition of concavity (refer to the proof of Theorem 4-4 and formula 4-54a), if an h_k distribution curve is strictly concave over an interval E, then for any three consecutive integers k-1, k, k+1 $\in$ E, the following relation must hold:

$$h_{(k-1)/2 + (k+1)/2} = h_k > \tfrac{1}{2}\,(h_{k-1} + h_{k+1}).$$

The above can be expressed as: $X_k = 2h_k - (h_{k-1} + h_{k+1}) > 0.$ (5-20)

Our task is then to prove how (5-20) could hold over the interval $[a_1, b_1]$ stated in the theorem.

Because of the strict concavity of H_k-curve as given by the theorem, then there exists:

$$H_k - \tfrac{1}{2}(H_{k-1} + H_{k+1}) > 0.$$

Or, $A = 2H_k - (H_{k-1} + H_{k+1}) > 0.$ (5-21)

From the definition of h_k, we know $H_k = h_k + h_{k+1}$, then (5-21) becomes:

$$A = 2H_k - (H_{k-1} + H_{k+1})$$

$$= 2(h_k + h_{k+1}) - ((h_{k-1} + h_k) + (h_{k+1} + h_{k+2}))$$

$$= (2h_k - (h_{k-1} + h_{k+1})) + (2h_{k+1} - (h_k + h_{k+2}))$$

$$= X_k + X_{k+1} > 0, \quad \text{(5-21a)}$$

where, $X_{k+1} = 2h_{k+1} - (h_k + h_{k+2})$. (5-20a)

Note that, X_{k+1} is a forwardly shifted X_k over triple (k, k+1, k+2), which can be always feasible for $\alpha > 5$, since $\max(k) = b - 1 < \alpha - 2$ from the boundary of the concavity interval of an H_k curve stated in (4-57) of Theorem 4-4.

From (5-21a), $A = X_k + X_{k+1} > 0$, we see it is not possible for both $X_k < 0$ and $X_{k+1} < 0$ to hold. On the other hand, if both $X_k > 0$ and $X_{k+1} > 0$ hold, it means conditions of (5-20) and (5-20a) are both satisfied and the concavity maintenance of h_k is fulfilled. Because

of the similarity of X_k and X_{k+1} in their formulation, we take X_k as the target to discuss the condition to keep its positivity. This can be done by the following manipulation:

$$X_k = 2h_k - (h_{k-1} + h_{k+1}) = (h_k - h_{k-1}) + (h_k - h_{k+1})$$

$$= ((H_{k-1} - h_{k-1}) - h_{k-1}) + ((H_k - h_{k+1}) - h_{k+1}) = (H_{k-1} - 2h_{k-1}) + (H_k - 2h_{k+1})$$

$$= (H_{k-1} + H_k) - 2(h_{k-1} + h_{k+1}) > 0,$$

which is the condition (5-19) stated in the theorem; and if $X_k > 0$ for every k within (a, b) is maintained, then h_k maintains the concavity.

On the other hand, from the above context, we see the definition of X_k does not restricted to (a, b), so the interval of h_k concavity can be either larger or smaller than [a, b], while it is harder to entail the concavity before than after the apex point q of the H_k curve. This can be seen from (5-19) again, which is reformulated as:

$$X_k = (H_{k-1} - 2h_{k-1}) + (H_k - 2h_{k+1}) > 0. \qquad (5\text{-}19a)$$

According to Corollary 5-3, we know $H_{k-1} > 2h_{k-1}$ for $k < q$, which support the above inequality. For the second term, although $H_k - 2h_k$ is ensured, but $H_k - 2h_{k+1} > 0$ is not, because h_k is increasing against k before q. Therefore, the ultimate sign of the left hand side of (5-19a) is not predetermined.

However, for $k > q$, although $H_{k-1} - 2h_{k-1}$ becomes negative (refer to Corollary 5-3), $H_k - 2h_{k+1}$ would be most likely positive and compensate the loss of $H_{k-1} - 2h_{k-1}$, because h_k is decreasing against k after q, and $(H_k - 2h_{k+1}) > (H_k - 2h_k) > (H_{k-1} - 2h_{k-1})$ could hold and (5-19a) to be reached. Additionally, as we can see from (5-21a), if $k = b - 1$, and if both

$X_k > 0$ and $X_{k+1} > 0$ hold, it means the right end of the h_k concavity interval will be extended from b to b + 1. That is, $b_1 > b$ takes place.

Theorem 5-3 is now fully proved. As have done in proof of Theorem 4-4, we continue the example given there with $\alpha = 10$, and g_i distributions = {2, 3, 52, 10, 8, 6, 5, 3, 2, 3}, the maximum H_k concavity interval is [1, 7]. Here we can find the corresponding h_k maintains this concavity interval without a change. In next chapter, we will see the cases that a_1 or/and b_1 differ from a or/and b in empirical datasets.

As mentioned before, the concavity and quasi concavity of h_k-curve is very important. It lays a theoretical foundation for the reduction of the number of raw patterns generated from full enumeration, and at the same time the sufficiency property is maintained, which ensures non-excessive reduction to happen. The concavity and quasi concavity are essential features of our theory, and a justification of the appropriateness of the proposed adjustment functions.

5.7 Further semantic justification of the adjustment of H_k to h_k

Having presented the above h_k properties, we now prove that the h_k adjustment exactly represents the principle of pattern de-sampling and the semantics of the two observations stated in section 5.2.

Theorem 5-4 (the equivalence theorem): The adjustment of H_k to h_k is effectually equivalent to an alternated pattern generation (combination) by a reduced order. That is, compared with $H_k = \sum_{i=k}^{i=a} g_i C_i^k$,

$$h_k = \sum_{i=k}^{i=a} g_i C_{i-1}^{k-1} . \qquad (5\text{-}22)$$

Proof: Notice that H_k and h_k both are summations of combination operations over individual tuples of the underlying dataset. Theoretically, there is no restriction of the data size to the use of H_k and h_k. For simplicity, we prove the theorem assuming u = 1, i.e., only one tuple contained in the dataset; and the length of the tuple is m. That is, for $H_k = C_m^k$, we only need to prove

$$h_k = C_{m-1}^{k-1}. \qquad (5\text{-}22a)$$

We prove (5-22a) in induction again.

Starting from k = m (the highest level):

$h_m = H_m = C_m^m = 1 = C_{m-1}^{m-1} = C_{m-1}^{k-1}$, which satisfy (5-22a).

For k = m – 1:

$h_{m-1} = H_{m-1} - h_m = C_m^{m-1} - 1 =$ m – 1

$= C_{m-1}^1 = C_{m-1}^{(m-1)-1} = C_{m-1}^{k-1}$,

which means (5-22a) holds.

For k = m – 2:

$h_{m-2} = H_{m-2} - h_{m-1}$

$= C_m^{m-2} -$ (m – 1) = ½ (m * (m – 1)) – (m – 1) = ½ (m – 1)(m-2)

$= C_{m-1}^2 = C_{m-1}^{(m-1)-2} = C_{m-1}^{(m-2)-1} = C_{m-1}^{k-1}$,

which means (5-22a) holds again.

Now, suppose k = t + 1 (0 < t < m) the formula (5-22a) holds, that is

$$h_{t+1} = C^{t}_{m-1}. \qquad (5\text{-}22b)$$

Then, for k = t,

$$h_t = H_t - h_{t+1} = C^{t}_{m} - C^{t}_{m-1}$$

$$= \frac{m!}{t!(m-t)!} - \frac{(m-1)!}{t!(m-t-1)!} = \frac{(m-1)!(m-(m-t))}{t!(m-t)!}$$

$$= \frac{t*(m-1)!}{t!(m-t)!} = \frac{(m-1)!}{(t-1)!(m-t)!} = C^{t-1}_{m-1}.$$

Note that the formula $C^{t}_{m} - C^{t}_{m-1} = C^{t-1}_{m-1}$ can be found from many mathematic textbooks.

In conclusion, when k = t, (5-22a) still holds, and it is fully proved.

Now, for the general case with g_i distribution and u > 1, as already known, $H_k = \sum_{i=k}^{i=a} g_i C^{k}_{i}$; and note h_{k+1} is a weighted (by g_i) summation of (5-22b), such that:

$$h_{k+1} = \sum_{i=k+1}^{i=a} g_i C^{k}_{i-1}, \text{ then}$$

$$h_k = H_k - h_{k+1} = \sum_{i=k}^{i=a} g_i C^{k}_{i} - \sum_{i=k+1}^{i=a} g_i C^{k}_{i-1} = \left(\sum_{i=k+1}^{i=a} g_i C^{k}_{i} + g_k\right) - \sum_{i=k+1}^{i=a} g_i C^{k}_{i-1}$$

$$= \sum_{i=k+1}^{i=a} g_i (C^{k}_{i} - C^{k}_{i-1}) + g_k = \sum_{i=k+1}^{i=a} g_i C^{k-1}_{i-1} + g_k C^{k-1}_{k-1} \qquad (5\text{-}22c)$$

$$= \sum_{i=k}^{i=a} g_i C^{k-1}_{i-1},$$

which is (5-22), and hence Theorem 5-4 is fully proved.

Additionally, follows are other implications of Theorem 5-4.

Firstly, set k' = k – 1, and i' = i – 1, then (5-22) becomes:

$$h_k = \sum_{i=k}^{i=a} g_i C_{i-1}^{k-1} = \sum_{i'=k'}^{i'=a-1} g_{i'+1} C_{i'}^{k'}, \qquad (5\text{-}22d)$$

which paves a way to prove the h_k quasi concavity property as done for theorem 5-2.

Since the h_k equivalence (5-22d) shown above has the same formulation as H_k, h_k should possess the same properties as H_k, and hence the concavity. We did not use this approach to prove theorem 5-2 is because it would be harder to derive other properties and corollaries presented in section 5.3.

Secondly, notice $C_n^0 = 1$, where $n \geq 0$. Then at k = 1, in a general case, (5-22) becomes:

$$h_1 = \sum_{i=k}^{i=a} g_i C_{i-1}^{k-1} |_{k=1} = \sum_{i=1}^{i=a} g_i C_{i-1}^{0} = \sum_{i=1}^{i=a} g_i = u, \qquad (5\text{-}23)$$

which means the conformity with formula (5-4) on one hand; on the other hand, it notifies us of attention to the details in calculation of w and w_a. When we calculate w, the accumulated H_ks, according to (4-12), k starts from 1 to α, for w_a, the accumulated h_ks (by 5-22d or 5-23), k starts from 0 to α -1. For instance in the particular case (5-22a), α = m and u = 1, then

$$w = \sum_{k=1}^{m} C_m^k = 2^m - 1;$$

and, $$w_a = \sum_{k=0}^{m-1} C_{m-1}^k = 2^{m-1} = ((2^m - 1) + 1)/2 = (w + u)/2.$$

This is another method that can be used to prove the "law of half".

Now, what can we infer from the above equivalence theorem? The inference is that, for a given tuple of m elements in DBo, different from the full enumeration regime (H_k) that generates patterns from the m elements, h_k takes (m – 1) elements as the generation base. Meanwhile, notice that the original tuple of m elements is kept as it is, as shown in the above proof. Considering these two aspects together, we can see that the adjustment h_k means that, either the m elements in whole (the original tuple) is a pattern, or (an inclusive or), at least one element is a "random walk", and hence the patterns can only be embodied in its subset of (m – 1) elements, which reflects exactly the semantics of observation 5-1 and 5-2, and hence well represent the principle of pattern frequency adjustment and proposition 5-1 presented in section 5-2. The above interpretation then illustrates that the proposed adjustment h_k is philosophically sound in three senses: It reduces meaningless pattern; it is safe; and it is rational. In other words, Theorem 5-4 presents a perfect semantic justification of the h_k adjustment, and it lays a foundation for dimension reduction and noise diminishment in pattern generation, in addition to what have been stated in the end of subsection 5-3.

Finally, we present the approach to calculate h_ks. Although it is the most efficient way to get h_ks from H_ks if they are available, there would be a need to know h_ks before H_ks or H_ks could not be precisely obtained. Because of the similar formulation of H_k and h_k equivalence, h_k can be calculated without knowing H_ks by the tabular approach used in Table 9. Table 11 gives an example. The only differences between Table 11 and 9 are: because of the reduced order combinations, each value of k is decreased by 1 (Table 11 keeps the original k values in the brackets to refer to the subscript of h_k); and according to

(5-23) the elements of the third row (k = 0(1)) would be the same as that of the second row (the G_1 series). Starting from row 4 (k = 1(2)), Θ_1 (first row) should be right shifted by 1 column. The remaining operations are the same as the ones described for Table 9 (see Section 4-5).

Table 11. The recursive computation of h_ks

Θ_1	1	2	3	4	5	6	
K \ G_1	2	3	2	2	0	1	h_k
0 (1)	**2**	**3**	**2**	**2**	**0**	**1**	10
1 (2)		**3**	**4**	**6**	**0**	**5**	18
2 (3)			**2**	**6**	**0**	**10**	18
3 (4)				**2**	**0**	**10**	12
4 (5)					**0**	**5**	5
5 (6)						**1**	1
	2	6	8	16	0	32	w_a = 64

The purpose of the above example is to show how to obtain the h_ks in an equivalent way, but their inner compositions (the elements of the bold triangular matrix of Table 11) may not necessary be kept the same under the de-sampling policies (e.g. the solution to the model presented in chapter 7).

The above presents an insight into the understanding and obtaining the h_ks based on the equivalence theorem. The application of the theorem can be further extended, as shown below.

5.8 Higher order reductions

As can be implied from the previous subsection and from the principle of pattern frequency adjustment, there would still be some meaningless patterns retained after the h_k adjustment, since the reduction is at the "least". In other words, there is a need to further

reduce the number of patterns, or, further adjust the pattern frequencies. Indeed, the approaches proposed in previous subsections can be rightly extended to feed the needs.

Since h_k has similar formulation as H_k, the reduction operations can be applied to h_k as well. For this we term h_k as the first order reduction of H_k. The second order reduction is noted as h^2_k, and

$$h^2_k = h_k - h^2_{k+1}, \tag{5-24}$$

which is analogous to (5-2).

The first order reduction is to diminish redundancy of pattern generation in general; the second order reduction reinforces the effect of the first order, and at the same time can effectually remove all generated length-one patterns (except those original tuples of one element only). This effect may be desired in some applications. As we have addressed in chapter 2, length-1 pattern could be meaningful, for instance, in spatial or sequential pattern mining, but may be less interested in some pure frequency based pattern mining applications. Following is how the length-1 patterns can be naturally de-sampled based on the equivalence Theorem 5-4 and (5-24).

Comparing the formulation of (5-22c), $h_{k+1} = \sum_{i=k+1}^{i=a} g_i C^k_{i-1}$, we can get,

$h^2_{k+1} = \sum_{i=k+1}^{i=a} g_i C^{k-1}_{i-2}$, this can be verified by its conformity with (5-24), which results: $h^2_k = \sum_{i=k}^{i=a} g_i C^{k-2}_{i-2}$.

Then, for k = 2, $h^2_2 = \sum_{i=2}^{i=a} g_i C^0_{i-2} = \sum_{i=2}^{i=a} g_i$

From (5-23):

$$h_1 = \sum_{i=1}^{i=a} g_i = \sum_{i=2}^{i=a} g_i + g_1.$$

Then, $h^2_1 = h_1 - h^2_2 = \sum_{i=2}^{i=a} g_i + g_1 - \sum_{i=2}^{i=a} g_i = g_1,$

which concludes that, at k = 1,the reduction of the number of frequencies is equal to that of all those generated but original length-one patterns. At the same time, this helps understand further why the apex point q' of h_k curve may right shift one position from that of the corresponding H_k curve, such that q' = q + 1 as stated in Theorem 5-2, which is because the one-order adjustment reduces relatively more of the frequencies of shorter patterns and hence relatively increases that of longer patterns and leads an apex shifting.

Again, due to formulation similarity, if g_1 is not very large compared with other g_i s, h^2_k will possess all of the properties of h_k; and the law of half becomes:

$$w_2 = \sum h^2_k = (w_a + g_1)/2, \qquad (5\text{-}23a)$$

where w_2 is defined as the accumulative frequency of all patterns after the second order reduction.

For completeness, we affirm that the reduction can be extended to higher orders, h^3_k, h^4_k, …, h^α_k , as defined below:

$$h^m_k = h^{m-1}_k - h^m_{k+1} = \sum_{i=k}^{i=a} g_i C^{k-m}_{i-m}, \quad 2 \le m \le k \le i \le \alpha, \qquad (5\text{-}24a)$$

and $h^m_k = g_k$, $0 < k < m.$ (5-24b)

(5-24a) starts from k = m is because its right hand side does not exist for k < m. Similar to the second order reduction, (5-24a) can be obtained directly from the original dataset by the tabular approach as given in Table 10 with some modifications.

The above formulae means, each higher order (m > 1) reduction (adjustment) will remove equally a whole level of (m – 1) generated but the original patterns. For instance $h^3{}_k$ will remove all of the generated length-two patterns effectually and end at $h^3{}_2 = g_2$. In this sense, the proposed adjustment approach is an inverse operation of the pattern generation, and we term it pattern "degeneration", a synonym of pattern de-sampling. The degeneration implies at least two applications. Firstly, it provides a way to recover the original dataset from the generated data. Such ability to recovery is an added feature of our approach: data recovery has many applications. Secondly, since full enumeration based pattern generation is excessive, the degeneration is rightly a correction, as long as it is not excessively corrected. In this regard, theoretically, the smaller the m (of $h^m{}_k$), the lower risk towards over correction, since smaller m means the recovered $g_i s$ will be within the left section of the $h^{m-1}{}_k$ series, and that section is more likely to maintain the concavity (refer to Theorem 5-4). However, a reduction of higher than the second order requires some cautious: This is because the quasi concavity property is not guaranteed to hold, because (5-24a) starts from k = m > 1, and the law of half needs to be reformulated, similar to what has been done on $h^2{}_k$ (refer to (5-23a)).

In short, for normal mining applications, the first order adjustment is generally applicable; the second order can be optional depending on the mining objectives. Higher

order reductions are not generally recommended in this thesis for pattern frequentness adjustment, but they can be used in other application purposes, for instance, data recovery, etc.. However, it could be a very interesting topic on how to determine up to what order the reduction can be safely and effectively rendered in an application. In this regard, we hypothesize that:

The order m of pattern reduction rises with the increase of sample size u:

$$m \propto u, \tag{5-25}$$

which implies $u \rightarrow \infty$, $m \rightarrow \alpha$. (5-25a)

That is, if a sample size is infinitively large, there is no need of pattern generation at all!

This hypothesis is based on the known "law of large number" in classic probability and statistics theory that, when the sample size is infinitively large, a measure obtained from the sample reaches the true value of the whole population in question [6, 16]. In this sense, whether the above hypothesis holds or not is converted into the question whether the large number theorem would hold in pattern mining problems. If the hypothesis holds, then its practical significance is obvious: when the dataset in question is sufficiently large, frequent patterns can be obtained from the dataset (the original patterns themselves) directly without pattern generation or with light generation only. As a result, the complexity of pattern mining problem will be greatly reduced. We would discuss these issues in a future work.

Finally, corresponding to the adjusted accumulative frequency w_m ($m = a, 2, 3, \ldots, \alpha$), the relative frequentness of a pattern after the adjustment is given by:

$$P(Z) = F_m(Z)/w_m. \quad (5\text{-}26)$$

where $F_m(Z)$ is the frequency of pattern Z after adjustment up to order m.

Meanwhile, the primary overfitting /underfitting ratio r_s proposed in Chapter 2 shall be modified with F_m and we get the corresponding *adjusted overfitting/underfitting ratio* r_a:

$$r_a(Z) = s(Z) /P(Z) = (F(Z) / u) / (F_m(Z)/w_m)$$

$$= F(Z) / F_m(Z) * w_m / u.$$

$$= F(Z) / F_m(Z) * w_m / w * w / u$$

$$= F(Z) / F_m(Z) * w_m / w * r_s(Z), \quad (5\text{-}27)$$

where $r_s(Z) = w / u$ is the corresponding primary overfitting/underfitting ratio.

Notice that $F(Z) \geq F_m(Z)$, and $w_m < w$. The above formula presents following properties:

1) If $F(Z) / F_m(Z) > w_m / w$, then $r_a(Z) > r_s(Z)$; otherwise $r_a(Z) \leq r_s(Z)$ would hold. Particularly, if a pattern's frequency is not changed after the refinement, i.e., $F(Z) = F_m(Z)$, then $r_a(Z) < r_s(Z)$.

2) Since w_m decreases with m increasing, then $r_a(Z)$ is monotonic decreasing against m as well. Its implication is obvious: more generation leads to more severe overfitting problem (note lower m means less degeneration and hence more generation). In other words, to reduce the overfitting problem, generate as small a number of patterns as possible. This is very understandable, and it then justifies

the designation of the two indicators, $r_s(Z)$ and $r_a(Z)$, since they properly describe the mining phenomena.

5.9 Summary

In this chapter, we have presented a theory over a set of properties to reduce the number of meaninglessly generated patterns via pattern frequency distribution adjustments. The theory lays a foundation for dimensional reduction and noise reduction in pattern generation. The theory and the adjustment functions resolve the drawbacks of conventional full enumeration based pattern mining approaches, including, the bias for generated patterns vs. original ones, the bias for shorter patterns vs. longer ones, the mixed element mining, and in overall the reduction of meaningless patterns and hence overfitting and underfitting issues. A practical significance of the theory is its potential compliance with the traditional (mineral) mining, wherein the mined materials are required to be refined before delivering to the end user. Conventional pattern mining approaches do not take this refining process and that is why users complain against the too many meaningless patterns delivered. Our proposal lays a theoretical foundation for reduction of meaningless patterns in more than one order to become possible.

At this stage, the adjustment functions are in terms of different collectives (of same pattern length k). Chapter 6 presents empirical verifications of the properties revealed in Chapter 4 and 5; Chapter 7 introduces a model to apply these collective based functions to individual pattern frequentness adjustments.

Chapter 6. Empirical verification

The properties revealed in the previous chapters have been theoretically proved. In Table 12, we present empirical verifications of these properties by 7 datasets. These datasets[5] have been used in a number of research articles, and as benchmarks used in FIMI 2003/04 [25, 39]. These datasets represent different types of data sources. For instance, the g_i distributions include two preliminary cases that all data tuples keep the same length, three datasets in ordinary distributions, and two not very ordinary (the "Accident" and the "Pummsb*" have 17 and 48 consecutive zeros in the left end of their respective g_i distributions). The datasets are empirically collected, except the last two which are generated ones. More info about these datasets can be found in [39].

Despite the dataset variations, the results from them well demonstrate the conformability with the theories developed in the previous two chapters. For instance, in H_k related properties, we see from Table 12 that, all H_k curves keep strict quasi concave property; while in the preliminary case (mushroom dataset) H_k has two adjacent apex values because of its odd u (refer to the proof of Theorem 4-1). And, the results show precisely that all the corresponding apex points satisfy $q \leq \alpha/2$; particularly, for the datasets Accident, Pumsb* and T1014D100k, their q values are significantly smaller than $\alpha/2$, a reflection of their left skewed g_i distributions (this can be seen from the comparison between the last two datasets). The results also show the intervals of the genuine

[5] We express our sincere thanks to the dataset providers.

concavity of the H_k curves as depicted in Theorem 4-4 in both preliminary and ordinary cases. In the preliminary cases, the intervals obtained from formula (4-57 and 4-57a)

Table 12. Empirical results

Item \ DBs		Mushroom	pumsb	Retail	Accident	Pumsb*	T40I10D100k	T1014D100k
u (tuples)		8124	49046	88162	340183	49046	100000	100000
n (elements)		120	7117	16470	469	7117	1000	1000
α (max length)		23	74	76	51	63	77	29
g_i distribution characteristics		Preliminary case	Preliminary case	Ordinary, 1 zero in the right	17 zeros in the left section	48 zeros in the left section	Some zeros in the left section, and 2 in the right	Ordinary; left skewed c.p. wt the left db
H_k	Quasiconca.	True	True	True	True	True	True	True
	q (≤ α/2?)	11, 12	37	38	21	28	38	11
	Concv intvl (theoretical)	[8, 15]	[32, 42]	[33, 43]	[21, 30]	[27, 36]	[34, 43]	[11, 18]
	Concv intvl (actual)	[8, 15]	[32, 42]	[33, 43]	[16, 25]	[23, 33]	[33, 43]	[7, 15]
	Concavity comparisons	Same	Same	Same	Left shifted	Left shifted	Left extended by 1	Left shifted
R_k	$0 < R_k \leq 1$?	All 1s	All 1s	$0 < R_k \leq 1$	$0 < R_k < 1$	$0 < R_k < 1$	$0 < R_k \leq 1$	$0 < R_k \leq 1$
	Monotonic	True	True	True	Decrease bt [1, 11] but rate < 1%	Decrease bt [1, 16] but rate < 1%	True	True
	Coro. 4-1	True	True	True	True	True	True	True
	Coro. 4-2	True	True	True	True	True	True	True
	Coro. 4-4	True	True	True	True	True	True	True
W (accumulative frequency)		68149043268	9.265E+26	1.0816E23	5.967E16	4.055E20	4.3158E23	6556956652
H_{odd}		34074525696	4.632E+26	5.4080E22	2.983E16	2.027E20	2.1579E23	3278528326
H_{even}		34074517572	4.632E+26	5.4080E22	2.983E16	2.027E20	2.1579E23	3278428326

Table 12. Empirical results (continued)								
Item \ DBs		Mushroom	pumsb	Retail	Accident	Pumsb*	T40I10D100k	T1014D100k
h_k	Quasi-concv	True	True	True	True	True	True	True
	q' = {q, q+1}	12	37, 38	38	21	28	38	12
	$h_{q+1} \geq h_{q-1}$?	True	True	True	True	True	True	True
	Corollary 5-3	True	True	True	True	True	True	True
	Error (%) b.t. h_q and $½H_q$	4.16	2.63	1.08	0.098	0.024	2.20	1.13
	Law of half	True	True	True	True	True	True	True
	Concave intvl (actual)	[9, 15]	[33, 42]	[33, 43]	[17, 26]	[24, 33]	[33, 43]	[8, 16]
	Compared with H_k concavity	$a_1 = a + 1 > a$	$a_1 = a + 1 > a$	Same	$a_1 = a + 1 > a$; $b_1 = b + 1 > b$	$a_1 = a + 1 > a$;	Same	$a_1 = a + 1 > a$; $b_1 = b + 1 > b$

are exactly the same as that numerically computed from the datasets. For those datasets in ordinary cases, the theoretical and actual concave intervals also conform to the conclusion of Theorem 4-4.

The R_k properties are verified too. For instance, R_k keeps 1 for all k in the two preliminary case datasets; and the number of 1s of the R_k series are equal to the number of 0s in the right tails of the g_i distributions of other datasets (refer to Appendix A), as stated in Corollaries 3-1, 3-2 and 3-4. As a reflection of the consecutive zeroed g_i s of the left section of g_i distributions in the two datasets stated above, the related R_ks are decreasing but at a slight rate (less than 1%), while R_ks are always monotonic increase in other cases.

The only parameters that cannot be fully precisely listed in this table are the w, H_{odd} and H_{even} due to their large values, except for the first and the last datasets from which we can see their relations.

On the h_k related series and their properties, Table 12 also presents a full conformity with the theorems and corollaries presented in previous chapter. For instance, the quasi concavity is well maintained in all the cases, and it is not affected by the slight fluctuations of the R_ks stated above. The results also precisely demonstrates the apex point q' of every h_k series compared with that of H_k series, such that q' equals q or q + 1. And interestingly, there is a case (for the "Pumsb*" dataset) that two adjacent apex values at q' = q and q' = q' + 1 are reached, as have been mentioned in the proof of Theorem 5-2. The results verified Corollary 5-3 that before q', $h_k \geq \frac{1}{2} h_k$, and after it, the relation is reversed, and the closeness of h_k and $\frac{1}{2} H_k$ at the apex point q'. The genuine concave h_k intervals and their relation with that of H_k are also presented in the last two rows of Table 12.

Details of the results are given in Appendix A.

Chapter 7. The optimized sampling model

In this chapter, we present an optimization model for the first order individual pattern frequency adjustment.

7.1 The model

The first order individual pattern frequency adjustment is the realization of the h_k function and the rational improvement of overall pattern generation, such that true frequent patterns would be kept frequent, false frequent ones be corrected, and random walks be de-sampled. Mathematically, these together mean an optimized sampling such that a "maximum likelihood" of the whole set of sampled patterns **Z** would be reached. That is, suppose the probability of an individual pattern Z^i is $p(Z^i) = f_w(Z^i)/w_a$, where $f_w(Z^i)$ is the adjusted frequency of pattern Z^i and w_a is the adjusted accumulative frequency, then the sampling leads to:

$$\prod_{Z} p(Z^i) \rightarrow \max \qquad (7\text{-}1)$$

Subject to: h_k, (k = 1, 2, ...α), (7-1a)

and: $0 < p(Z^i) \leq F(Z^i)/w_a$, (7-1b)

where $F(Z^i)$ is the raw frequency of pattern Z^i.

(7-1) is a very neat optimization model but it captures the spirit of all the sampling optimization mechanisms addressed above.

Unlike most other likelihood maximization problems typically concerning resolving only one or a few given parameters, it is challenging to solve (7-1) because of the need of identifying a huge number of patterns and their frequencies in an application. However, the solution can be reachable with the following strategies:

1. Pursue as few patterns as possible. This is based on the fact that every $p(Z^i) < 1$ (unless the number of patterns i = 1, but it is too trivial), and the adjusted accumulative frequency w_a is fixed by h_k function. This strategy implies pattern de-sampling. At the same time, because of the constraint of h_k, excessive de-sampling will be controlled.
2. For those patterns which could not be de-sampled but their collective frequency is larger than that required by an h_k, keep those concerned patterns' frequencies as close as possible while maintaining their frequentness orders. We term this a "trim" of frequency, which leads a correction (reduction) of the frequencies of overfitted patterns.
3. Localize the operation as much as possible, and ideally render the operation within every two adjacent levels (collectives). This is because, the maximization of the objective function (7-1) can be rearranged as:

$$\text{Max}(\prod_{Z} p(Z^i)) = \max(\prod_{k=1}^{k=\alpha} (\prod_{L_k} p(Z^i_k)) = \prod_{k=1}^{k=\alpha} (\max(\prod_{L_k} p(Z^i_k)), \text{ s.t. } h_k \quad (7\text{-}2)$$

where L_k is the collective of patterns of length k (refer to section 4.3), and the above formula indicates that global optimization is the product of local optimizations of different levels (collectives).

Though the h_k functions are level-wise defined, the de-sampling at level k must be done by referring the super patterns and their frequencies in level k+1. This implies the localization should minimally refer two levels.

Pursuing the largest $p(Z^i)$ can also be a strategy, and is embodied in the first strategy. This is because, firstly, each raw frequency $F(Z^i)$ already has a maximum value since it resulted from full enumeration as discussed in previous chapters. Secondly, since the adjusted accumulative frequency w_a is fixed and decreased from w, and since the number of retained patterns after the de-sampling is reduced to its minimum, any $p(Z^i)$ equals $F(Z^i)/w_a$ automatically reaches its maximum value as shown in (7-1b). At the same time, we see how the optimization alters the frequentness of the resulted pattern via (7-1b).

The above presents the essential features of our approach, and the strategies are guidelines to pattern de-sampling and frequency distribution adjustments. How to apply these strategies to form an operable solution of the optimization model (7-1) would be a challenge. What we are thinking includes, for example, the adaptation of the known linear programming (LP) technology or mixed integer linear programming in particular [37, 38] to solve the model, since the constraint (7-1a) of the model are in integer domain, and the object function of the model can be translated into linearity by a logarithm transformation:

$$x_i = \ln(p(Z^i)). \qquad (7\text{-}3)$$

However, this transformation would cause, for instance, a transformation of (7-1a), which would complicate the transformation. We are also considering other more effective

approach to solve the problem, and following is our trial solution for the running example (Table 1).

7.2 A sample solution

To apply the above three solution strategies to model (7-1), we use the following rules:

Rule 7.1: The de-sampling rule: if $F(Z^i_k) \leq f_w(Z^j_{k+1})$ and $Z^i_k \subset Z^j_{k+1}$, then Z^i_k can be a candidate pattern to be de-sampled, where f_w is the adjusted frequency (of pattern Z^j_{k+1}, which stays in the result set).

The above rule says if a sub pattern Z^i_k is not more frequent than its super pattern Z^j_{k+1}, then it can be de-sampled, where the superscripts i and j are respective enumerators. This rule is based on Proposition 4-3 and the frequency adjustment principle (stated in Section 5-2). Proposition 4-3 says, if $Z^i_k \subset Z^j_{k+1}$, and if $F(Z^i_k) = F(Z^j_{k+1})$, then they must be generated from the same data tuples. Since Z^j_{k+1} is already in the result set, and since Z^i_k is not more frequent than Z^j_{k+1} generated from the same data tuples, then it means Z^i_k does not bear more information than Z^j_{k+1} and hence can be de-sampled.

Rule 7.2: The competition and tie-resolution rule: when a competition takes place, that is, $F(Z^s_k) = F(Z^t_k) \leq F(Z^j_{k+1})$, where $Z^s_k \subset Z^j_{k+1}$ and $Z^t_k \subset Z^j_{k+1}$, then if there is a case that $F(Z^s_{k-m}) > F(Z^t_{k-m})$, where $Z^s_{k-m} \subset Z^s_k$ and $Z^t_{k-m} \subset Z^t_k$, $0 < m < k$, then Z^s_k should be de-sampled. In case of a tie, a random draw is used to break the tie.

The above rule says if two or more sub patterns of Z^j_{k+1} are of equal frequency and hence equally good to be de-sampled, then the sub pattern Z^s_k that has stronger (more frequent) descendent/s (sub pattern/s) should be de-sampled. Note that, according to the downward

closure property and the frequency adjustment principle stated in 5-2, there is at most one sub pattern that can be de-sampled against a pattern Z^j_{k+1}.

The above rule is based on the rationale that, although Z^s_k and Z^t_k are equally frequent at level k, Z^s_k is indeed relatively weaker than Z^t_k, since Z^s_k's sub pattern/s (at level k – m) is stronger than that of the other. The rule also implies that, when m is reached, the comparison stops. Finally, the second part of the rule refers to a tie case where no stronger sub pattern of Z^s_k is found than that of Z^t_k and vice versa until k = 1. In this case the miner can arbitrarily choose Z^s_k or Z^t_k to be de-sampled. We can prove that, in this case, a draw would not affect the output of next level. However, the proof is not given here. This no side effect propagation is a merit of the maximization model and the above defined solution rules.

Based on the above, we define adjusted pattern frequency below:

Definition 7.1: The "adjusted pattern frequency" $f_w(Z^j_k)$ (k = 1, 2, ...a-1) is defined as:

$$f_w(Z^j_k) = 0, \quad \text{(de-sampled case)} \tag{7-4}$$

$$\text{or,} \quad f_w(Z^j_k) = F(Z^j_k) - F_d, \quad \text{(trimmed case)} \tag{7-5}$$

$$\text{or,} \quad f_w(Z^j_k) = F(Z^j_k), \quad \text{(unchanged case)} \tag{7-6}$$

where, $F_d > 0$ is the frequency to be reduced, and it is determined in the following conceptual algorithm.

The above definition and the following algorithm can be simply understood as that, during the adjustment course, a pattern is de-sampled (if it is not more frequent than its

antecedent (super pattern), and if it is the least frequent one compared with its siblings), or its frequency is reduced (trimmed). If a pattern got no antecedent to compare with in a step, then its frequency is not affected (the unchanged case).

Algorithm 7-1:

For (k = a-1, a-2, ..., 2, 1) do:

For **each** Z^i_{k+1}, let y = $f_w(Z^i_{k+1}) > 0$,

If there are sub-patterns, Z^j_k, such that $F(Z^j_k) \le y$, do

While ($F(Z^j_k) \le y$)

If there is only one such Z^j_k,

then de-sample it (set $f_w(Z^j_k) = 0$), and let y = y - $F(Z^j_k)$

If more than one such Z^j_k exists,

then resolve the competition and tie case with rule 7-2.

End while

End if ... do

If there is no such sub-pattern Z^j_k that $F(Z^j_k) \le y$, then

For each Z^j_k in the sub patters set $\{ Z^s_k\}$, reset its frequency

$f_w(Z^j_k) = F(Z^j_k) - F_d$,

where F_d is determined by the following steps:

Set: $F_s = \sum_s F(Z^s_k)$, $Z^j_k \in \{ Z^s_k\}$ (7-7)

F_d = round ($y * F(Z^j_k) / F_s$). (7-8)

Then, $f_w(Z^j_k) = F(Z^j_k) - F_d$ (7-9)

end for each.

end for ... do.

The above algorithm features the following: it is a top-down approach since the h_k constraints are defined top-down; and each time the operations are localized in two

adjacent levels, k and k+1. This localization greatly simplifies the solution. However, if a competition or a tie occurs, its resolution may require multiple level searches and comparisons as stated in rule 7-2.

Table 13. A mutual element free dataset

TID	VID
T_1	V_1, V_0
T_2	V_2, V_0, V_8
T_3	V_2, V_6
T_4	V_1, V_6, V_8
T_5	V_1, V_2, V_3, V_0, V_8
T_6	V_5
T_7	V_0
T_8	V_5
T_9	V_1, V_2
T_{10}	V_1, V_2, V_3, V_8

Now, we use the running example (Table 1) to demonstrate the solution of the optimization model using algorithm 7.1. In Table 1, we notice a "mutual pattern". A "mutual pattern" Z_k is a pattern whose k compositional elements always occur simultaneously in the original dataset. We define the elements of a mutual pattern as "mutual elements"[6]. In table 1, V_4 and V_7 are mutual elements and they form a mutual pattern in this example. Discovering a mutual pattern is certainly interesting, not only because it can be seen as a true pattern in any sense, but also because it has a big impact on pattern mining, particularly in the refinement solutions – it will cause a de-sampling tie as can be imagined. This tie causation can also be formally proved, since the frequencies of all sub patterns of a mutual pattern are the same! Since a tie could greatly complicate the resolution process as we have mentioned, it is desired to eliminate it by merging the mutual elements as a single (composite) element only. This resolution does not only reduce the tie complication, but also reduce the mining dimension.

[6] Finding mutual pattern itself is a data mining problem!

Table 14. The raw patterns from Table 1

k	C_k	H_k		Raw Patterns
A	B	C	E	X_1
1	8	28	0.237	V_1(.042)/5, V_2(.042)/5, V_3(.017)/2, V_4(.034)/4, V_5(.017)/2, V_6(.017)/2, V_7(.034)/4, V_8(.034)/4
2	18	36	0.305	V_{47}(.034)/4, V_{12}(.025)/3, V_{18}(.025)/3, V_{28}(.025)/3, V_{13}(.017)/2, V_{14}(.017)/2, V_{17}(.017)/2, V_{23}(.017)/2, V_{24}(.017)/2, V_{27}(.017)/2, V_{38}(.017)/2, V_{48}(.017)/2, V_{78}(.017)/2, V_{16}(.009), V_{26}(.009), V_{34}(.009), V_{37}(.009), V_{68}(.009)
3	21	30	0.254	V_{123}(.017)/2, V_{128}(.017)/2, V_{138}(.017)/2, V_{147}(.017)/2, V_{238}(.017)/2, V_{247}(.017)/2, V_{248}(.017)/2, V_{278}(.017)/2, V_{478}(.017)/2, V_{124}(.009), V_{127}(.009), V_{137}(.009), V_{148}(.009), V_{168}(.009), V_{178}(.009), V_{134}(.009), V_{234}(.009), V_{237}(.009), V_{347}(.009), V_{348}(.009), V_{378}(.009)
4	15	17	0.144	V_{1238}(.017)/2, V_{2478}(.017)/2, V_{1234}(.009), V_{1237}(.009), V_{1247}(.009), V_{1248}(.009), V_{1278}(.009), V_{1347}(.009), V_{1348}(.009), V_{1378}(.009), V_{1478}(.009), V_{2347}(.009), V_{2348}(.009), V_{2378}(.009), V_{3478}(.009)
5	6	6	0.051	V_{12347}(.009), V_{12348}(.009), V_{12378}(.009), V_{12478}(.009), V_{13478}(.009), V_{23478}(.009)
6	1	1	0.009	V_{123478}(.009)
$\sum$	69	118	1.00	
Notes: the terns in X_i (i = 1, 2, 3), e.g, V_{123}(.017)/2 means: pattern $V_1V_2V_3$(probability s'$_z$)/frequency. If frequency not specified, it is 1.				

Merging V_4 and V_7 into V_0 gives the data set shown in Table 13, wherein the longest pattern length α is reduced from 6 to 5, and the related H_k series becomes {24, 25, 16, 6, 1}. The reduction certainly simplifies the problem. For instance, the total number of patterns generated from the reduced case (Table 13) is 37, and the accumulative frequency w = 72, compared with the 69 patterns and their accumulative frequency 118

in the original problem (seen in Table 14). The raw patterns generated from Table 13 are presented in Table 15.

Table 15. The raw patterns from Table 13 after merging V_4 and V_7

k	C_k	H_k	$\sum s'_z$	Raw Patterns
A	B	C	E	X
1	7	24	0.333	V_0(.056)/4, V_1(.069)/5, V_2(.069)/5, V_3(.028)/2, V_5(.028)/2, V_6(.028)/2, V_8(.056)/4
2	13	25	0.347	V_{01}(.028)/2, V_{02}(.028)/2, V_{08}(.028)/2, V_{03}(.014), V_{13}(.028)/2, V_{23}(.028)/2, V_{38}(.028)/2, V_{12}(.042)/3, V_{18}(.042)/3, V_{28}(.042)/3, V_{16}(.014), V_{26}(.014), V_{68}(.014)
3	11	16	0.222	V_{012}(.014), V_{013}(.014), V_{018}(.014), V_{023}(.014), V_{028}(.028)/2, V_{038}(.014), V_{123}(.028)/2, V_{128}(.028)/2, V_{138}(.028)/2, V_{238}(.028)/2, V_{168}(.014)
4	5	6	0.083	V_{0123}(.014), V_{0128}(.014), V_{0138}(.014), V_{0238}(.014), V_{1238}(.028)/2
5	1	1	0.014	V_{01238}(.014)
$\sum$	37	72	1.00	
Notes: the terns in the column X, e.g, V_{123}(.028)/2 means: pattern $V_1V_2V_3$(probability s'_z)/frequency. If frequency not specified, it is 1.				

Applying Algorithm 7-1 and starting from $k = \alpha - 1 = 4$, there is only one pattern at k = 5, V_{01238} with frequency y = 1. We see its sub patterns listed in k = 4 of Table 15 are of same frequency 1 except V_{1238}. It means then there is a competition to resolve. According to algorithm 7-1, we go down to k = 3, and find that V_{0128} and V_{0238} have equal number of stronger sub patterns than other ones. Then we keep looking into level k = 2 and find that V_{0128} has a stronger sub pattern (V_{18}) than that of V_{0238}. V_{0128} is then de-sampled. That is, at k = 4, the refined pattern set is {V_{0123}(.014), V_{0138}(.014), V_{0238}(.014),

$V_{1238}(.028)/2\}$ as shown in Table 16. In this example, from k = 4 to k = 2, the main refinement operation is pattern de-sampling, and the refined pattern set is seen in Table 16.

At k = 1, frequency trimming defined in algorithm 7-1 plays the major role, since the frequencies of all the sub patterns at k = 1 are larger than that of their respective super patterns at k = 2. Notice that at k = 2 in Table 16 and k = 1 in Table 15, there are following correspondences:

$V_{01}(2) \leftrightarrow \{V_0(4), V_1(5)\}$, $V_{08}(2) \leftrightarrow \{V_0(4), V_8(4)\}$, $V_{28}(2) \leftrightarrow \{V_2(5), V_8(4)\}$,

$V_{12}(3) \leftrightarrow \{V_1(5), V_2(5)\}$, $V_{18}(3) \leftrightarrow \{V_0(4), V_1(5)\}$, $V_{01}(2) \leftrightarrow \{V_0(4), V_1(5)\}$,

$V_{01}(2) \leftrightarrow \{V_0(4), V_1(5)\}$,

Table 16. The refined patterns from Table 15

k	C_k	h_k	$\sum s'_z$	Raw Patterns
A	B	C	E	X
1	7	10	0.244	$V0_1(.049)/2$, $V_1(.024)$, $V_2(.049)/2$, $V_3(.049)/2$, $V_5(.049)/2$, $V_6(.028)$
2	7	14	0.341	$V_{01}(.049)/2$, $V_{08}(.049)/2$, $V_{12}(.042)/3$, $V_{18}(.042)/3$, $V_{28}(.049)/2$, $V_{26}(.024)$, $V_{68}(.024)$
3	7	11	0.268	$V_{013}(.024)$, $V_{028}(.049)/2$, $V_{038}(.024)$, $V_{123}(.049)/2$, $V_{138}(.049)/2$, $V_{238}(.049)/2$, $V_{168}(.024)$
4	5	5	0.122	$V_{0123}(.024)$, $V_{0138}(.024)$, $V_{0238}(.024)$, $V_{1238}(.049)/2$
5	1	1	0.024	$V_{01238}(.024)$
$\sum$	27	41	1.00	
Notes: the terms in the column X, e.g, $V_{123}(.049)/2$ means: pattern $V_1V_2V_3$(probability s'_z)/frequency. If frequency not specified, it is 1.				

where the left hand side of ←→ is the adjusted frequency f_w of a pattern at k = 2, and the right hand side is the corresponding two sub patterns at k = 1 and their raw frequencies which are noted in the round brackets.

Applying formulae (7-7) to (7-9) to the above, we get:

$F_d(V_0) = 4 * 2 / 9 + 1 * 2 / 2 = 1.9$ → 2

$F_d(V_1) = 5 * 2 / 9 + 1 * 3 / 2 + 5 * 3 / 9 = 4.3$ → 4

$F_d(V_2) = 5 * 2 / 9 + 1 * 3 / 2 + 2 / 7 = 3$ → 3

$F_d(V_6) = 2 / 7 + 2 / 6 = 0.6$ → 1

$F_d(V_8) = 1 / 2 + 4 * 2 / 9 + 4 * 3 / 9 + 4 / 6 = 4$ → 4

That is, based on formula (7-5) the refined patterns and their frequencies obtained from Table 16 at k = 1 are: {$V_0(2)$, $V_1(1)$, $V_2(2)$, $V_3(2)$, $V_5(2)$, $V_6(1)$}.

This completes the refinement process, and the refined patterns and their adjusted frequencies are seen in Table 16. However, for a conceptual consistency in pattern length, we need to slightly reorganize the result set, since V_0 is not really a single element. For example, V_{01} stored in k = 2 of Table 16 should be moved to k = 3, since it is exactly a length-3 pattern V_{147}. The rearranged result set is presented in Table 17, wherein we substituted the element number V_{47} in lieu of V_0 to avoid confusions. At the same time, we note that there is no need to split V_{47} into V_4 and V_7 again and put their separate combinations with other elements into the result set, for instance, V_{14} or V_{234}. This is simply because V_4 and V_7 always occur together. If one sees a V_{14}, then it is exactly V_{147}! This is an important significance of finding a mutual pattern, not only

because of the pattern itself, but also because of the reduction of the pattern generation and the reduction of the mining dimension, as illustrated in the above example.

Table 17. The reorganized refined patterns

k	C_k	h_k	$\sum s'_z$	Raw Patterns
A	B	C	E	X
1	5	8	0.195	V_1(.024), V_2(.049)/2, V_3(.049)/2, V_5(.049)/2, V_6(.028)
2	7	12	0.293	V_{47}(.049)/2, V_{28}(.049)/2, V_{26}(.024), V_{68}(.024), V_{12}(.073)/3, V_{18}(.73)/3
3	6	11	0.268	V_{123}(.049)/2, V_{138}(.049)/2, V_{238}(.049)/2, V_{168}(.024), V_{147}(.049)/2, V_{478}(.049)/2
4	4	6	0.146	V_{1238}(.049)/2, V_{1347}(.024), V_{2478}(.049)/2, V_{3478}(.024)
5	3	3	0.073	V_{12347}(.024), V_{13478}(.024), V_{23478}(.024),
6	1	1	0.024	V_{123478}(.024)
$\sum$	27	41	1.00	
Notes: the terns in the column X, e.g, V_{123}(.049)/2 means: pattern $V_1V_2V_3$(probability s'_z)/frequency. If frequency not specified, it is 1.				

The result set demonstrate the essences of the adjustment model. For instance, the number of patterns is reduced (from 37 in Table 15 to 27 in Table 17, about 30% decrease; compared with that in Table 14, the reduction is from 69 to 27, a 60% decrease); the frequentness of most retained patterns (in Table 17) is enhanced compared that in Table 15. For instance, in Table 17, the probabilities of all the patterns are between 0.024 and 0.073; and there are more than 50% of the patterns that their probabilities are no less than 0.049. However, in Table 15, the probabilities of all the patterns are between 0.014 and 0.069; and there are less than 20% of the patterns that their probabilities are over 0.042. This enhancement is even more striking compared with

that in Table 14. Consequently, the maximization objective of the optimization model is then reached.

The results also reflect what we have argued in previous chapters. For instance, recall that we have interpreted the dataset Table 1 (equally Table 13 above) as the dancer example (refer to Example 2-2 in Chapter 2); there we discussed the problem of an entertainment company which wants to find out the most possible potential solo dancer/s. In conventional mining approaches (refer to Table 14, at k = 1), the answer would be $V_1(5)$,$V_2(5)$, $V_4(4)$, $V_7(4)$, and $V_8(4)$, while $V_5(2)$ is almost impossible, since it is one of the least frequent elements. We have argued that the result is counter commonsense, since V_5 is observed as a solo dancer twice, while all the rest are generated patterns, although their raw frequencies are much higher than that of V_5. Our argument is now justified, and from the result set $\{V_1(1), V_2(2), V_3(2), V_5(2), V_6(1)\}$, as shown in the row k = 1, Table 17, the entertainment company will find that V_1 is weaker than V_5, a reverse conclusion from the conventional mining approaches. Similarly, V_2 becomes only comparable with V_5 from a much stronger position in the conventional case. More strikingly, V_8 gets fully out of the scene after the refinement, but it is rather stronger than V_5 in the conventional case. At the same time, V_3 can be seen as a surprising pattern, since its raw frequentness is one of the lowest compared with other elements, but it becomes comparable with V_5 and V_2 in the refined case. Finally, it is also very understandable that V_4 and V_7 are not in the answer set, because V_4 and V_7 are mutual elements, they act always together and could not be individual players! However, in conventional mining approaches, the answer is reversed.

7.3 Significance and implications

From the above sample solution, we see that, the results from the refinement model are at least more interpretable and more coherent with commonsense rather than counter intuition exhibited in the result set of conventional mining approaches. The results from the solution also indicate that, although the optimization model aims at maximization of the probabilities in whole, it does not exclude those less frequent but potentially interesting patterns. The results support a vision that the adjustment model provides a mechanism to retain and discover surprising patterns. Using conventional approaches, we would be unable to get the above insights into the solo dancer situation. One of the reasons for these noticeable points can be seen from the adjustment algorithm: every pattern's frequency will be examined and compared as long as it has an antecedent. Furthermore, the comparisons are rendered both vertically and laterally. That is, the comparison must be done between a parent and its children, and among the siblings. This is the essence of mining. Conventional approaches do not implement these comparisons. It should also be noticed that the frequency comparisons are done "locally" with relevance. The comparisons are not extended to patterns of no relation. This is the basic reason for the less frequent patterns to be retained in the result set which is another significant difference from conventional approaches that compare pattern frequencies globally and absolutely without consideration of relevance. It then implies the inability of conventional frequent pattern mining approaches to discover and retain less frequent but meaningful patterns, or surprising patterns.

Having discussed the main characteristics of the optimization model, we notice that the merits of the model and its solutions, such as the ability to discover surprising patterns, are theoretical only at this moment. Furthermore, the example is very small, and the solution only demonstrates that the optimization model is solvable. The solution complexity could increase rapidly with the increase of the number of elements involved and the data size. A specific complexity could be due to the existence of mutual pattern(s), which is indeed a mining problem itself. More general complexities would mostly originate from decision making problems. In the above sample solution, we have set up some rules to resolve competition and tie problems, but these may not be the only ones required. In general, to solve a realistic refinement problem, it is imperative to address the following:

- Investigate and identify in what and how many situations we will face a decision making problem, and the nature of each decision making problem.
- What rule(s) to be used to make a decision; the justification of the rule and the consequences of the decision to be made, especially, whether alternative decisions would lead to substantially different final result sets.

All of these are research problems and only after these problems have been resolved, could we design and implement the appropriate algorithm(s) to solve the refinement model for practical pattern mining.

As stated earlier, the nature of the refinement is a reduction of the number of meaninglessly generated patterns, in other words, a correction of the full enumeration generation, which implies an inception of "selective pattern generation" mode. The

above proposed refinement model can be seen as an indirect selective pattern generation approach, since the selectivity is equivalent to selection of patterns from the fully generated pattern set. Our expectation is that, after the refinement strategies and the related decision making complexities elaborated above have been fully studied, they can be applied to develop a direct selective pattern generation approach. Thus, patterns could be generated directly to fit to the target refined result set without involving the use of full enumeration. This is analogous to obtaining the refined accumulative frequency w_a and h_ks directly without knowing the raw w and H_ks as presented in Section 5.6. In short, the refinement model and the development of a selective pattern generation regime could herald a radical change of pattern mining methodologies.

Chapter 8. Conclusions, contributions and future work

Pattern mining is fundamental to many data mining tasks, especially association rule mining, causation mining and the like, of which pattern frequency distribution and frequentness determination play essential roles. Conventional pattern mining approaches focus on mining efficiency but largely ignore the importance of appropriate measure of pattern frequentness and the refinement of the mining results sets. Even worse, such unrefined results are taken to be final and delivered to the users. This thesis reveals theoretical pitfalls underlying and subsequently the drawbacks imposed by conventional mining approaches, and then proposes their corresponding resolutions with theoretical proofs. This is the general contribution of our work.

Problems, such as overfitting and huge number of meaningless patterns resulted from mining have been noticed for some years, but no substantial investigation has been reported. This could be because of the difficulty of the problems that are not only computational but philosophical as well. We noticed such difficulties, and we are aware that revealing and resolving philosophical problems could not be so exciting or as easy to be recognized as developing an efficient algorithm. However, we believe the first important thing in research is the unearthing and identifying the problems.

Our study has identified that the overfitting problem is dominantly existent in previously proposed approaches, wherein overfitting implies meaningless combinations of elements are falsely taken to be meaningful patterns, or less frequent ones be frequent patterns. The overfitting and other drawbacks revealed in the second chapter of this thesis are

rooted mainly from two sources. One is the probability anomaly caused by the conventional defined and widely used "support" s_z; another is the full enumeration pattern generation regime. These discoveries are the first contribution of our work.

Based on the problem investigations, we have presented our primary reformulation of s_z, which forms our second contribution. The reformulation is simple but effective: it automatically resolves the probability anomaly and covers other issues addressed in from subsection 2.3.1 through 2.3.6; it also fulfills the requirements raised at the end of section 3.2. Along with the resolution, the degree of overfitting embodied in the conventional mining approaches can also be quantified. We have numerically illustrated the effectiveness and the striking differences of the mining result set of our proposed reformulation compared with the one obtained by conventional mining approaches.

Compared with the frequentness measure issue, it demands much more intellections in understanding and resolving the full enumeration pattern generation related drawbacks. This thesis provides an insight into the problems, and proposes a refining framework based on our third contribution in explorations of a set of intrinsic properties governing pattern frequency distributions, such that the merit of the full enumeration generation is maintained while the related drawbacks addressed in Section 2.3 are corrected.

Findings presented in this thesis are well derived from set theory and combinatiorics, and we believe they could be well exploited in applications of pattern mining. These findings, particularly the H_k and h_k curves and their properties, could form a set of reliable indicators and check points to test and guide pattern mining. For example, if an H_k curve derived from a dataset is not quasi concave, then there must be some

characteristics embodied in the related data source, or the mining approach might have not been correctly implemented. More importantly, the H_k and h_k curves and their quasi concavity properties theoretically prove the possibility to obtain a succinct mining result set while maintaining sufficiency. The succinctness means the overfitting and underfitting problems will be controlled, and the sufficiency property is to ensure only meaningless patterns to be reduced without an excessive cut-off. A salient feature of our proposal is the promise of substantial reduction of the number of meaninglessly generated patterns before domain specific constraints are imposed. Moreover, our proposed reduction can be done in more than one order, complying with the real industrial mining practice that the mined raw materials should be refined more than once before delivering to the user. In short, this thesis presents a refinement theory for dimension reduction in pattern generation and for noise reduction, so that graceful degeneration of knowledge acquisition is achievable. This is the fourth contribution of our work.

Our fifth contribution is the maximum likelihood model developed in the last chapter to realize the proposed refining theory. We have analyzed and suggested a number of strategies to solve the optimized sampling model. In the end, we have presented a sample solution to demonstrate that the optimization model is solvable. At the same time the solution supports what we have addressed on the drawbacks of the conventional approaches and their expected resolutions.

The conventional support measure, the full enumeration pattern generation mode and the "downward closure" property based on it, are the foundations of conventional mining approaches. A modification of these foundations would indicate a radical change of the

state of the art of pattern mining. This is because, our proposal means an overpass of the full enumeration pattern generation mode adopted by the conventional mining approaches. In other words, findings and the refinement model presented in this thesis would lead to a development of selective pattern generation regime. Under this regime, the downward closure property is no longer to hold, and the widely used pattern pruning and the like strategies will become invalid. Accordingly, the focus of data mining will be shifted from conventionally on how to efficiently produce all possible frequent combinations (patterns) into how to obtain meaningful patterns. This is the principle significance of our work.

The above has summarized our main contributions and the possible impacts. These impacts will certainly extend to other mining tasks based on pattern mining, for instance, association rule mining, causation mining.

Our future work can be divided into two stages. In the first stage, we will try to find the fully fledged operational approaches and algorithms to solve the proposed optimized refinement model over real datasets, which corresponds to an indirect selective pattern generation approach. The model optimally selects patterns from the fully generated pattern set. This proposed future work would mainly focus on the investigation and identification of various decision making problems, for instance, to decide which pattern(s) to be retained or degenerated, or to decide to what degree a pattern's frequency may be reduced, and so on. The next stage work will be on how to develop a direct selective pattern generation mode, such that patterns would be generated directly to fit to the target refined pattern set without the involvement of full enumeration.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases". Proceedings of the ACM SIGMOD International Conference on the Management of Data, Washington, D.C., USA, pp 207-216, May 1993.

[2] J. Han, H. Cheng, D. Xin, X. Yan, "Frequent pattern mining: current status and future directions". Data Mining and Knowledge Discovery, Volume 15, No. 1, pp55–86, 2007.

[3] T. Wang, B. C. Desai, "Issues in Pattern Mining and their Resolutions". Proceedings of Canadian Conference on Computer Science & Software Engineering, C3S2E 2009, Montreal, Quebec, Canada, 2009. ACM International Conference Proceeding Series, ACM 2009, ISBN 978-1-60558-401-0.

[4] H. Mannila, H. Toivonen, "Multiple uses of frequent sets and condensed representations". Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), pp189–194, 1996.

[5] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules". Proceedings of the 20th VLDB Conference, Santiago, Chile, pp 487-499, 1994.

[6] Gut, Allan, "Probability: A Graduate Course". Springer 2005, ISBN 0387228330.

[7] J. Han, J. Pei, Y. Yin, "Mining frequent patterns without candidate generation".

Proceeding of the 2000 ACM-SIGMOD international conference on management of data (SIGMOD'00), Dallas, TX, pp 1–12, 2000.

[8] H. Toivonen, "Sampling Large Databases for Association Rules". Proceedings of the 22nd VLDB Conference, Mumbai(Bombay), India, pp 134–145, 1996.

[9] P. Shenoy, J.R. Haritsa, S. Sudarshan, G. Bhalotia, M. Bawa, and D. Shah, "Turbo-charging vertical mining of large databases". ACM SIGMOD Record, Volume 29, Issue 2 (June 2000), pp 22-23, ISSN: 0163-5808.

[10] M. Zaki, "Scalable algorithms for association mining". IEEE Transactions on Knowledge Data Engineering, Volume 12, Issue 3 (May/Jun 2000). pp372 – 390, 2000. ISSN: 1041-4347.

[11] K. Gade, J. Wang, G. Karypis, "Efficient closed pattern mining in the presence of tough block constraints". In Proceeding of the 2004 international conference on knowledge discovery and data mining (KDD'04), Seattle, WA, pp 138–147, 2004.

[12] D. T. Drewry, L. Gu, A. B Hocking, et al, "Current State of Data Mining", Technical Report: CS-2001-15, University of Virginia, Charlottesville, VA, USA, 2002.

[13] N. Pasquier, Y. Bastide , R. Taouil, L. Lakhal, "Discovering frequent closed itemsets for association rules". Proceedings of the 7th international conference on database theory (ICDT'99), Jerusalem, Israel, pp 398–416, 1999.

[14] RJ Bayardo, "Efficiently mining long patterns from databases". Proceedings of the

1998 ACM-SIGMOD international conference on management of data (SIGMOD'98), Seattle,WA, pp 85–93, 1998.

[15] H. Xiong, P. Tan, V. Kumar, "Hyperclique pattern discovery". Data Mining and Knowledge Discovery, Volume 13, Number 2, pp. 219-242(24), Publisher: Springer, September 2006.

[16] H. Tijms, "Understanding Probability". Cambridge University Press, 2004. ISBN: 0521833299.

[17] P. Billingsley, "Probability and Measure", 3rd Edition. Wiley-Interscience, 1995. ISBN-10: 0471007102.

[18] D. Hand, "Statistics and Data Mining: Intersecting Disciplines". SIGKDD exploration, ACM SIGKDD, volume 1, issue 1, 1999.

[19] Q. Yang, X. Wu, "10 Challenging Problems in Data Mining Research". International Journal of Information Technology & Decision Making, Vol. 5, No. 4 (2006), pp597–604 World Scientific Publishing Company, 2006.

[20] "Principle of Inclusion-exclusion", PlanetMath, Licensed under GFDL. URL: http://planetmath.org/?op=getobj&from=objects&id=2803, accessed in December 2009.

[21] P. Halmos, "Measure theory". 1St Edition edition (January 1, 1950) Van Nostrand and Co. ISBN-10: 0442030630

[22] M. Avriel, W.E. Diewert, S. Schaible and I. Zang, "Generalized Concavity", Plenum Press, New York, New York, 1988. ISBN10: 0306426560.

[23] R. Ng, L.V.S. Lakshmanan, J. Han, A. Pang, "Exploratory mining and pruning optimizations of constrained associations rules". In: Proceeding of the 1998 ACM-SIGMOD international conference on management of data SIGMOD'98), Seattle, WA, pp 13–24, 1998.

[24] F. Geerts, B. Goethals, J. V.Den Bussche, "Tight upper bounds on the number of candidate patterns", ACM Transactions on Database Systems (TODS), Volume 30, Issue 2, June 2005.

[25] B.Goethals and M. J. Zaki, "Advances in Frequent Itemset Mining Implementations: Introduction to fimi03". In Bart Goethals and Mohammed J. Zaki, editors, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'03), volume 90 of CEUR Workshop Proceedings, Melbourne, Florida, USA, 19. November 2003.

[26] J. Pei, J. Han, L.V.S. Lakshmanan, "Mining frequent itemsets with convertible constraints". Proceeding of the 2001 international conference on data engineering (ICDE'01), Heidelberg, Germany, pp 433–332, 2001

[27] B. Morantz, J, "Constrained Data Mining". In Volume I of "Encyclopedia of Data Warehouse", by J. Wang, Second Edition. Publisher, Information Science Reference, 2009, ISBN: 978-1-60566-010-3

[28] T. Calders, B. Goethals, "Mining All Non-derivable Frequent Itemsets". Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, PKDD 2002: pp74-85, 2002.

[29] J. F. Boulicaut, A. Bykowski, C. Rigotti, "Approximation of frequency queries by means of free-sets". Proceedings of PKDD Intentional Conference on Principles of Data Mining and Knowledge Discovery, pages 75–85, 2000.

[30] G. Liu, J. Li and L. Wong, "A new concise representation of frequent itemsets using generators and a positive border". Knowledge and Information Systems, Vol. 17, Issue 1, pp 35-56, 2008. ISSN:0219-1377

[31] M. Kryszkiewicz, "Concise representation of frequent patterns based on disjunction-free generators". Proceedings of IEEE Int. Conf. on Data Mining, pp305–312, 2001.

[32] J. Wang, J. Han, Y. Lu, P. Tzvetkov, "TFP: An efficient algorithm for mining top-k frequent closed itemsets". IEEE Trans Knowl Data Eng (2005) 17: pp652–664, 2005

[33] X. Yan, H. Cheng, J. Han, D. Xin, "Summarizing itemset patterns: a profile-based approach". Proceedings of the 2005 ACM SIGKDD international conference on knowledge discovery in databases (KDD'05), Chicago, IL, pp 314–323, 2005

[34] T. Mielikäinen, "An Automata Approach to Pattern Collections". Knowledge Discovery in Inductive Databases, 3rd International Workshop, KDID 2004,

pp130-149, 2004.

[35] T. Mielikäinenv, "Implicit Enumeration of Patterns". Knowledge Discovery in Inductive Databases, 3rd International Workshop, KDID 2004, pp150-172, 2004

[36] G. Ramesh, W. A. Maniatty, M. J. Zaki, "Feasible itemset distributions in data mining: theory and application". In Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (June 2003), PODS 2003, pp284-295, 2003.

[37] A. Schrijver, "Theory of Linear and Integer Programming". John Wiley & sons, 1998. ISBN: 978-0-471-98232-6.

[38] M. J. Todd. "The many facets of linear programming". Mathematical Programming 91 (3) pp417—436, Feb. 2002.

[39] Frequent Itemset Mining Dataset Repository, http://fimi.cs.helsinki.fi/data/, accessed July 2009.

[40] A. Silberschatz and A. Tuzhilin, "What Makes Patterns Interesting in Knowledge Discovery Systems", IEEE Transactions on Knowledge and Data Engineering, v.8 n.6, pp970-974, December 1996.

[41] C. Y. Chan and Y. E. Ioannidis , "An Efficient Bitmap Encoding Scheme for Selection Queries", Proceedings of the 1999 ACM SIGMOD international conference on management of data, pp.215-226, 1999.

[42] S. Bistarelli, and F. Bonchi, "Interestingness is not a Dichotomy: Introducing Softness in Constrained Pattern Mining", Knowledge Discovery in Databases:

PKDD 2005, Springer Berlin / Heidelberg, 2005. ISSN: 0302-9743.

[43] D. Cheung , J. Han, V. Ng, C. Wong, "Maintenance of discovered association rules in large databases: an incremental updating technique". Proceedings of the 1996 international conference on data engineering (ICDE'96), New Orleans, LA, pp 106–114, 1996.

[44] SBrin, R. Motwani, J. Ullman, S. Tsur, " Dynamic itemset counting and implication rules for

market basket analysis". Proceedings of the 1997 ACM-SIGMOD international conference on

management of data (SIGMOD'97), Tucson, AZ, pp 255–264, 1997

[45] D. Cheung, J. Han, V. Ng, A. Fu, Y. Fu, "A fast distributed algorithm for mining association rules". Proceedings of the 1996 international conference on parallel and distributed information systems, Miami Beach, FL, pp 31–44, 1996.

[46] J. Park, M. Chen, P. Yu, " An effective hash-based algorithm for mining association rules". Proceedings of the 1995 ACM-SIGMOD international conference on management of data (SIGMOD'95), San Jose, CA, pp 175–186, 1995.

[47] A. Savasere, E. Omiecinski, S. Navathe, "An efficient algorithm for mining association rules in large databases". Proceeding of the 1995 international conference on very large data bases (VLDB'95), Zurich, Switzerland, pp 432–443, 1995.

[48] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, D. Yang, "H-Mine: Hyper-Structure

Mining of Frequent Patterns in Large Databases". Proceedings of 2001 First IEEE International Conference on Data Mining (ICDM'01), pp441-448, ICDM, 2001.

[49] G. Liu, H. Lu, Y. Xu, J. Yu, "Ascending Frequency Ordered Prefix-tree: Efficient Mining of Frequent Patterns". Proceedings of the Eighth International Conference on Database Systems for Advanced Applications, pp65, 2003, ISBN:0-7695-1895.

[50] G. Liu, H. Lu, W. Lou, J. Yu, "On computing, storing and querying frequent patterns". Proceedings of the 2003 ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03), Washington, DC, pp 607–612, 2003.

[51] G. Grahne, J. Zhu, "Efficiently using prefix-trees in mining frequent itemsets". Proceedings of the ICDM'03 international workshop on frequent itemset mining implementations (FIMI'03), Melbourne, FL, pp 123–132, 2003.

[52] T. Wang, B. C. Desai, H. Zheng, Y. Qiao, "Knowledge discovery in Chinese medicine". Proceedings of Canadian Conference on Computer Science & Software Engineering, C3S2E 2008, Montreal, Quebec, Canada, 2008. ACM International Conference Proceeding Series, ACM 2009, pp113-116. ISBN 978-1-60558-401-0.

[53] Y. Aumann, Y. Lindell, "A statistical theory for quantitative association rules". Proceeding of the 1999 international conference on knowledge discovery and data mining (KDD'99), San Diego, CA, pp 261–270, 1999.

[54] H. Zhang, B. Padmanabhan, Tuzhilin, "On the discovery of significant statistical quantitative rules". Proceedings of the 2004 intl. conference on knowledge

discovery and data mining (KDD'04) Seattle, WA, pp 374–383, 2004.

[55] C. Ordonez, E. Omiecinski, L. de Braal, C. A. Santana, et al, "Mining Constrained Association Married Rules to Predict Heart Disease". IEEE International Conf. on Data Mining, ICDM, pp 432--440, 2001.

[56] Y. Feng, Z. Wu, Z. Zhou, "Enhancing Reliability throughout Knowledge Discovery Process". Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06), pp754-758, ICDMW, 2006.

[57] B. R´acz, F.Bodon, L.Schmidt-Thieme, "On Benchmarking Frequent Itemset Mining Algorithms". Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations (OSDM'05), pp36-45, ACM 2005.

[58] S. Stigler, "Fisher and the 5% level". Chance, Springer New York, Vol. 21. No. 4, pp 12, 2008, ISSN: 0933-2480 (Print) 1867-2280 (Online).

[59] Ganter, Bernhard; Wille, Rudolf , " Formal Concept Analysis: Mathematical Foundations", Springer-Verlag, Berlin, 1998, ISBN 3-63311-62767-5. Translated by C. Franzke.

APPENDIX A: THE OUTPUT OF EMPIRICAL DATASETS

The databases used as follows are original from the source [39] without any change, except the format of the files has been changed from html to txt to read into computer easily. The computation is programmed with C#.

Output for mushroom_dat.txt

Total number of elements n = 120

Total number of tuples u = 8124

Longest pattern length Alpha = 23

The Gk distributions:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8124

The Hk Series:

(1) 186852 (2) 2055372 (3) 14387604 (4) 71938020 (5) 273364476 (6) 820093428 (7) 1991655468 (8) 3983310936 (9) 6638851560 (10) 9294392184 (11) 10984281672 (12) 10984281672 (13) 9294392184 (14) 6638851560 (15) 3983310936 (16) 1991655468 (17) 820093428 (18) 273364476 (19) 71938020 (20) 14387604 (21) 2055372 (22) 186852 (23) 8124

The Hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12= 13- 14- 15- 16- 17- 18- 19- 20- 21- 22- 23-

The Hk Genuine Concavity: Hk:(Hk - (Hk-1 + Hk+1)/2 >= 0?); theoretic concavity domain = [8, 15]; exact = [8, 15].

Detailed as below:

1<> 2:(-5231856) 3:(-22609092) 4:(-71938020) 5:(-172651248) 6:(-312416544) 7:(-410046714) 8:(-331942578) 9:(0) 10:(482825568) 11:(844944744) 12:(844944744) 13:(482825568) 14:(0) 15:(-331942578) 16:(-410046714) 17:(-312416544) 18:(-172651248) 19:(-71938020) 20:(-22609092) 21:(-5231856) 22:(-844896)

The Rk Series:

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

The Rk Monotonic: + Increase, - decrease, = equality.

= 1+ 2= 3= 4= 5= 6= 7= 8= 9= 10= 11= 12= 13= 14= 15= 16= 17= 18= 19= 20= 21= 22=

The accumulative frequency w = 68149043268

The sum of odd length pattern frequencies H_odd = 34074525696

The sum of even length pattern frequencies H_even = 34074517572

The hk Series:

(1) 8124 (2) 178728 (3) 1876644 (4) 12510960 (5) 59427060 (6) 213937416 (7) 606156012 (8) 1385499456 (9) 2597811480 (10) 4041040080 (11) 5253352104 (12) 5730929568 (13) 5253352104 (14) 4041040080 (15) 2597811480 (16) 1385499456 (17) 606156012 (18) 213937416 (19) 59427060 (20) 12510960 (21) 1876644 (22) 178728 (23) 8124

The hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13- 14- 15- 16- 17- 18- 19- 20- 21- 22- 23-

The hk Genuine Concavity: hk:(hk -(hk-1 + hk+1)/2 >= 0?); Concavity domain = [9, 15].

Detailed as below:

1<> 2:(-763656) 3:(-4468200) 4:(-18140892) 5:(-53797128) 6:(-118854120) 7:(-193562424) 8:(-216484290) 9:(-115458288) 10:(115458288) 11:(367367280) 12:(477577464) 13:(367367280) 14:(115458288) 15:(-115458288) 16:(-216484290) 17:(-193562424) 18:(-118854120) 19:(-53797128) 20:(-18140892) 21:(-4468200) 22:(-763656)

Output for pumsb_dat.txt

Total number of elements n = 7117

Total number of tuples u = 49046

Longest pattern length Alpha = 74

The Gk distributions:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 49046

The Hk Series:

(1) 3629404 (2) 132473246 (3) 3179357904 (4) 56433602796 (5) 790070439144 (6) 9085810050156 (7) 88262154772944 (8) 739195546223406 (9) 5.42076733897164E+15 (10) 3.52349877033157E+16 (11) 2.05003564819291E+17 (12) 1.07626871530128E+18 (13) 5.13297387297533E+18 (14) 2.23651004465354E+19 (15) 8.94604017861415E+19 (16) 3.29885231586397E+20 (17) 1.12549079011829E+21 (18) 3.56405416870793E+21 (19) 1.05045807077708E+22 (20) 2.88875969463696E+22 (21) 7.42823921478075E+22 (22) 1.78953035628809E+23 (23) 4.04589471856437E+23 (24) 8.59752627694929E+23 (25) 1.71950525538986E+24 (26) 3.24060605823473E+24 (27) 5.76107743686175E+24 (28) 9.67037998330365E+24 (29) 1.5339223421792E+25 (30) 2.3008835132688E+25 (31) 3.26577014786539E+25 (32) 4.38837863619412E+25 (33) 5.58520917333797E+25 (34) 6.73510517961344E+25 (35) 7.69726306241536E+25 (36) 8.33870165094997E+25 (37) 8.56407196584051E+25 (38) 8.33870165094997E+25 (39) 7.69726306241536E+25 (40) 6.73510517961344E+25 (41) 5.58520917333797E+25 (42) 4.38837863619412E+25 (43) 3.26577014786539E+25 (44) 2.3008835132688E+25 (45) 1.5339223421792E+25 (46) 9.67037998330365E+24 (47) 5.76107743686175E+24 (48) 3.24060605823473E+24 (49) 1.71950525538986E+24 (50) 8.59752627694929E+23 (51) 4.04589471856437E+23 (52) 1.78953035628809E+23 (53) 7.42823921478074E+22 (54) 2.88875969463696E+22 (55) 1.05045807077707E+22 (56) 3.56405416870793E+21 (57) 1.12549079011829E+21 (58) 3.29885231586397E+20 (59) 8.94604017861415E+19 (60) 2.23651004465354E+19 (61) 5.13297387297533E+18 (62) 1.07626871530128E+18 (63) 2.05003564819291E+17 (64) 3.52349877033157E+16 (65) 5.42076733897164E+15 (66) 739195546223406 (67) 88262154772944 (68) 9085810050156 (69) 790070439144 (70) 56433602796 (71) 3179357904 (72) 132473246 (73) 3629404 (74) 49046

The Hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+ 29+ 30+ 31+ 32+ 33+ 34+ 35+ 36+ 37+ 38- 39- 40- 41- 42- 43- 44- 45- 46- 47- 48- 49- 50- 51- 52- 53- 54- 55- 56- 57- 58- 59- 60- 61- 62- 63- 64- 65- 66- 67- 68- 69- 70- 71- 72- 73- 74-

The Hk Genuine Concavity: Hk:(Hk - (Hk-1 + Hk+1)/2 >= 0?); theoretic concavity domain = [32, 42]; exact = [32, 42].

Detailed as below:

1<> 2:(-1459020408) 3:(-25103680117) 4:(-340191295728) 5:(-3781051387332) 6:(-35440302555888) 7:(-285878523363837) 8:(-2.01531920064889E+15) 9:(-1.25663242857979E+16) 10:(-6.99771783758158E+16) 11:(-3.50748286683006E+17) 12:(-1.59272000359603E+18) 13:(-6.58771070794299E+18) 14:(-2.4931587383023E+19) 15:(-8.66647642303246E+19) 16:(-2.77590364365821E+20) 17:(-8.2147891002887E+20) 18:(-2.25098158023659E+21) 19:(-

5.721244849768E+21) 20:(-1.35058894814195E+22) 21:(-2.96379241397818E+22) 22:(-6.04828963733135E+22) 23:(-1.14763359805432E+23) 24:(-2.02294735928219E+23) 25:(-3.30674087574973E+23) 26:(-4.9968528789107E+23) 27:(-6.94415583907443E+23) 28:(-8.79770446023222E+23) 29:(-1.00038413620383E+24) 30:(-9.89627317534964E+23) 31:(-7.88609268660678E+23) 32:(-3.71110244075615E+23) 33:(2.34672654341932E+23) 34:(9.38690617367726E+23) 35:(1.60359647133653E+24) 36:(2.08034136822037E+24) 37:(2.2537031489054E+24) 38:(2.08034136822037E+24) 39:(1.60359647133653E+24) 40:(9.38690617367726E+23) 41:(2.34672654341932E+23) 42:(-3.71110244075615E+23) 43:(-7.88609268660673E+23) 44:(-9.89627317534968E+23) 45:(-1.00038413620383E+24) 46:(-8.79770446023222E+23) 47:(-6.94415583907443E+23) 48:(-4.9968528789107E+23) 49:(-3.30674087574973E+23) 50:(-2.02294735928219E+23) 51:(-1.14763359805432E+23) 52:(-6.04828963733135E+22) 53:(-2.96379241397817E+22) 54:(-1.35058894814195E+22) 55:(-5.721244849768E+21) 56:(-2.25098158023659E+21) 57:(-8.2147891002887E+20) 58:(-2.77590364365821E+20) 59:(-8.66647642303245E+19) 60:(-2.4931587383023E+19) 61:(-6.58771070794299E+18) 62:(-1.59272000359603E+18) 63:(-3.50748286683006E+17) 64:(-6.99771783758158E+16) 65:(-1.25663242857979E+16) 66:(-2.01531920064889E+15) 67:(-285878523363837) 68:(-35440302555888) 69:(-3781051387332) 70:(-340191295728) 71:(-25103680117) 72:(-1459020408) 73:(-62631742)

The Rk Series:

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

The Rk Monotonic: + Increase, - decrease, = equality.

= 1+ 2= 3= 4= 5= 6= 7= 8= 9= 10- 11+ 12= 13+ 14- 15= 16= 17= 18+ 19- 20+ 21- 22- 23+ 24= 25+ 26- 27= 28- 29+ 30- 31+ 32= 33= 34= 35= 36= 37= 38= 39= 40= 41= 42= 43- 44- 45+ 46= 47= 48- 49+ 50+ 51= 52- 53= 54= 55= 56= 57= 58= 59= 60+ 61- 62+ 63- 64= 65= 66= 67= 68= 69= 70= 71- 72+ 73=

The accumulative frequency w = 9.26452746075298E+26

The sum of odd length pattern frequencies H_odd = 4.63226373037649E+26

The sum of even length pattern frequencies H_even = 4.63226373037649E+26

The hk Series:

(1) 49046 (2) 3580358 (3) 128892888 (4) 3050465016 (5) 53383137780 (6) 736687301364 (7) 8349122748792 (8) 79913032024152 (9) 659282514199254 (10) 4.76148482477239E+15 (11) 3.04735028785433E+16 (12) 1.74530061940748E+17 (13) 9.01738653360531E+17 (14) 4.2312352196148E+18 (15) 1.81338652269206E+19 (16) 7.13265365592209E+19 (17) 2.58558695027176E+20 (18) 8.66932095091119E+20 (19) 2.69712207361681E+21 (20) 7.80745863415393E+21 (21) 2.10801383122156E+22 (22) 5.32022538355918E+22 (23) 1.25750781793217E+23 (24) 2.7883869006322E+23 (25) 5.80913937631709E+23 (26) 1.13859131775815E+24 (27) 2.10201474047658E+24 (28) 3.65906269638516E+24 (29) 6.01131728691848E+24 (30) 9.32790613487351E+24 (31) 1.36809289978145E+25 (32) 1.89767724808394E+25 (33) 2.49070138811018E+25 (34) 3.09450778522779E+25 (35)

3.64059739438564E+25 (36) 4.05666566802971E+25 (37) 4.28203598292025E+25 (38)
4.28203598292025E+25 (39) 4.05666566802971E+25 (40) 3.64059739438564E+25 (41)
3.09450778522779E+25 (42) 2.49070138811018E+25 (43) 1.89767724808394E+25 (44)
1.36809289978145E+25 (45) 9.32790613487351E+24 (46) 6.01131728691848E+24 (47)
3.65906269638516E+24 (48) 2.10201474047658E+24 (49) 1.13859131775815E+24 (50)
5.80913937631709E+23 (51) 2.7883869006322E+23 (52) 1.25750781793217E+23 (53)
5.32022538355918E+22 (54) 2.10801383122156E+22 (55) 7.80745863415394E+21 (56)
2.69712207361681E+21 (57) 8.66932095091119E+20 (58) 2.58558695027176E+20 (59)
7.13265365592209E+19 (60) 1.81338652269206E+19 (61) 4.2312352196148E+18 (62)
9.01738653360531E+17 (63) 1.74530061940748E+17 (64) 3.04735028785433E+16 (65)
4.76148482477239E+15 (66) 659282514199254 (67) 79913032024152 (68) 8349122748792 (69)
736687301364 (70) 53383137780 (71) 3050465016 (72) 128892888 (73) 3580358 (74) 49046

The hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+ 29+ 30+ 31+ 32+ 33+ 34+ 35+ 36+ 37+ 38= 39- 40- 41- 42- 43- 44- 45- 46- 47- 48- 49- 50- 51- 52- 53- 54- 55- 56- 57- 58- 59- 60- 61- 62- 63- 64- 65- 66- 67- 68- 69- 70- 71- 72- 73- 74-

The hk Genuine Concavity: hk:(hk -(hk-1 + hk+1)/2 >= 0?); Concavity domain = [33, 42].

Detailed as below:

1<> 2:(-60890609) 3:(-1398129799) 4:(-23705550318) 5:(-316485745410) 6:(-3464565641922) 7:(-31975736913966) 8:(-253902786449871) 9:(-1.76141641419902E+15) 10:(-1.08049078715989E+16) 11:(-5.91722705042169E+16) 12:(-2.91576016178789E+17) 13:(-1.30114398741724E+18) 14:(-5.28656672052575E+18) 15:(-1.96450206624973E+19) 16:(-6.70197435678273E+19) 17:(-2.10570620797994E+20) 18:(-6.10908289230876E+20) 19:(-1.64007329100571E+21) 20:(-4.08117155876228E+21) 21:(-9.42471792265725E+21) 22:(-2.02132062171245E+22) 23:(-4.02696901561891E+22) 24:(-7.44936696492427E+22) 25:(-1.27801066278976E+23) 26:(-2.02873021295997E+23) 27:(-2.96812266595073E+23) 28:(-3.97603317312369E+23) 29:(-4.82167128710853E+23) 30:(-5.18217007492973E+23) 31:(-4.71410310041995E+23) 32:(-3.17198958618682E+23) 33:(-5.39112854569321E+22) 34:(2.88583939798859E+23) 35:(6.50106677568863E+23) 36:(9.5348979376767E+23) 37:(1.1268515744527E+24) 38:(1.1268515744527E+24) 39:(9.5348979376767E+23) 40:(6.50106677568863E+23) 41:(2.88583939798859E+23) 42:(-5.39112854569321E+22) 43:(-3.17198958618682E+23) 44:(-4.71410310041995E+23) 45:(-5.18217007492973E+23) 46:(-4.82167128710853E+23) 47:(-3.97603317312369E+23) 48:(-2.96812266595073E+23) 49:(-2.02873021295997E+23) 50:(-1.27801066278976E+23) 51:(-7.44936696492427E+22) 52:(-4.0269690156189E+22) 53:(-2.02132062171245E+22) 54:(-9.42471792265725E+21) 55:(-4.08117155876229E+21) 56:(-1.64007329100571E+21) 57:(-6.10908289230876E+20) 58:(-2.10570620797994E+20) 59:(-6.70197435678273E+19) 60:(-1.96450206624973E+19) 61:(-5.28656672052575E+18) 62:(-1.30114398741724E+18) 63:(-2.91576016178789E+17) 64:(-5.91722705042169E+16) 65:(-1.08049078715989E+16) 66:(-1.76141641419902E+15) 67:(-253902786449871) 68:(-31975736913966) 69:(-3464565641922) 70:(-316485745410) 71:(-23705550318) 72:(-1398129799) 73:(-60890609)

Output for retail_dat.txt

Total number of elements n = 16470

Total number of tuples u = 88162

Longest pattern length Alpha = 76

The Gk distributions:

3016 5516 6919 7210 6814 6163 5746 5143 4660 4086 3751 3285 2866 2620 2310 2115 1874 1645 1469 1290 1205 981 887 819 684 586 582 472 480 355 310 303 272 234 194 136 153 123 115 112 76 66 71 60 50 44 37 37 33 22 24 21 21 10 11 10 9 11 4 9 7 4 5 2 2 5 3 3 0 0 1 0 1 1 0 1

The Hk Series:

(1) 908576 (2) 7164335 (3) 52502539 (4) 366817927 (5) 2447321444 (6) 15534598332 (7) 93307736462 (8) 527550301625 (9) 2796416534241 (10) 13863139450195 (11) 64204046715896 (12) 277757200264229 (13) 1.12312584494064E+15 (14) 4.24904654295735E+15 (15) 1.5058885990449E+16 (16) 5.00625023811958E+16 (17) 1.56327472119759E+17 (18) 4.59121121980175E+17 (19) 1.26976301242088E+18 (20) 3.31067777753649E+18 (21) 8.14636975894685E+18 (22) 1.8935525717633E+19 (23) 4.16131440789675E+19 (24) 8.65288386954046E+19 (25) 1.70361679656958E+20 (26) 3.17787016348576E+20 (27) 5.61949830792223E+20 (28) 9.4248316680112E+20 (29) 1.49988185541216E+21 (30) 2.26577690430042E+21 (31) 3.2501290738274E+21 (32) 4.42825143223911E+21 (33) 5.73212226482143E+21 (34) 7.05069716806149E+21 (35) 8.24219004470137E+21 (36) 9.15769085268638E+21 (37) 9.67116815427317E+21 (38) 9.70768639718059E+21 (39) 9.26120299243453E+21 (40) 8.39616377302037E+21 (41) 7.23232941850291E+21 (42) 5.91772992838759E+21 (43) 4.59811839985694E+21 (44) 3.39150713519183E+21 (45) 2.37356853011541E+21 (46) 1.57537354482505E+21 (47) 9.91013257283473E+20 (48) 5.9046404820075E+20 (49) 3.32957641309953E+20 (50) 1.77535631861495E+20 (51) 8.94240979386786E+19 (52) 4.25025368313273E+19 (53) 1.90382440924376E+19 (54) 8.0257650807726E+18 (55) 3.17919252128806E+18 (56) 1.18129875755773E+18 (57) 4.10926627354641E+17 (58) 1.33528383786941E+17 (59) 4.04304582589712E+16 (60) 1.13749193542012E+16 (61) 2.96417842579272E+15 (62) 712836643924391 (63) 157536096738717 (64) 31838940145963 (65) 5851270021055 (66) 971240578752 (67) 144437457258 (68) 19056211853 (69) 2203389079 (70) 219831833 (71) 18542293 (72) 1285749 (73) 70375 (74) 2851 (75) 76 (76) 1

The Hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+ 29+ 30+ 31+ 32+ 33+ 34+ 35+ 36+ 37+ 38+ 39- 40- 41- 42- 43- 44- 45- 46- 47- 48- 49- 50- 51- 52- 53- 54- 55- 56- 57- 58- 59- 60- 61- 62- 63- 64- 65- 66- 67- 68- 69- 70- 71- 72- 73- 74- 75- 76-

The Hk Genuine Concavity: Hk:(Hk - (Hk-1 + Hk+1)/2 >= 0?); theoretic concavity domain = [33, 43]; exact = [33, 43].

Detailed as below:

1<> 2:(-19541222.5) 3:(-134488592) 4:(-883094064.5) 5:(-5503386685.5) 6:(-32342930621) 7:(-178234713516.5) 8:(-917311833726.5) 9:(-4398928341669) 10:(-19637092174873.5) 11:(-81606123141316) 12:(-315907745564039) 13:(-1.14027602667015E+15) 14:(-3.84195937473746E+15) 15:(-1.20968884716276E+16) 16:(-3.56306766739083E+16) 17:(-9.8264340060926E+16) 18:(-2.53924120290146E+17) 19:(-6.15136437337449E+17) 20:(-1.39738860814738E+18) 21:(-2.97673198863789E+18) 22:(-5.94423120132421E+18) 23:(-1.11190381275513E+19) 24:(-1.94585731725583E+19) 25:(-3.1796247865032E+19) 26:(-4.83687388760149E+19) 27:(-6.81852607826247E+19) 28:(-8.84326763010704E+19) 29:(-1.04248180138614E+20) 30:(-1.09228560319354E+20) 31:(-9.68850944423698E+19) 32:(-6.28742370853074E+19) 33:(-7.35203532886717E+18) 34:(6.35410133000913E+19) 35:(1.37996034327435E+20) 36:(2.01011753199109E+20) 37:(2.38479529339683E+20) 38:(2.41500823826744E+20) 39:(2.09277907334049E+20) 40:(1.49397567551648E+20) 41:(7.53825677989307E+19) 42:(2.50601920766935E+18) 43:(-5.65001319327722E+19) 44:(-9.43363297943486E+19) 45:(-1.09871809893028E+20) 46:(-1.06917348874388E+20) 47:(-9.19055392294298E+19) 48:(-7.1521401095963E+19) 49:(-5.10421987211691E+19) 50:(-3.36552377628212E+19) 51:(-2.05949864077324E+19) 52:(-1.17286341842308E+19) 53:(-6.22590686361234E+18) 54:(-3.08295322609023E+18) 55:(-1.4243393978771E+18) 56:(-6.13760816763625E+17) 57:(-2.46486943317692E+17) 58:(-9.21501590198654E+16) 59:(-3.20211933115998E+16) 60:(-1.03223989881807E+16) 61:(-3.07969957327008E+15) 62:(-848020617341325) 63:(-214801695296460) 64:(-49854743233923) 65:(-10553820341302.5) 66:(-2026613160404.5) 67:(-350710938044.5) 68:(-54264211315.5) 69:(-7434632764) 70:(-891133853) 71:(-92016498) 72:(-8020585) 73:(-573925) 74:(-32374.5) 75:(-1350)

The Rk Series:

0.210272925251529 0.297094051410328 0.382831246282321 0.463316718039126 0.536416236147605 0.600644667263741 0.65552176787858 0.701570924935985 0.739920275015639 0.771879594163558 0.798676329287781 0.82134661868907 0.840718401553246 0.857434446016042 0.871986691039956 0.884749696089944 0.896009184273812 0.905984998255989 0.914848923844976 0.922738141516824 0.929765099200748 0.936024549137197 0.941598411268332 0.94655903067817 0.950971295213339 0.954893979352975 0.958380588604126 0.961479900011198 0.964236331011297 0.96669022077285 0.968878073679284 0.970832791317424 0.972583904574083 0.974157808823637 0.975578000710482 0.976865313167724 0.978038144975912 0.979112681618661 0.980103104973037 0.981021790211479 0.981879489047814 0.98268549907259 0.983447819379726 0.984173293000019 0.984867736850611 0.985536060009938 0.98618237115967 0.986810076020608 0.987421965565713 0.988020295733908 0.988606859302998 0.989183050515657 0.989749922993555 0.990308241423762 0.990858527459561 0.991401100244955 0.991936111947395 0.992463578665651 0.992983407067619 0.993495417104725 0.993999361143785 0.994494939852474 0.994981815169571 0.995459620685058 0.995927969747206 0.996386461603664 0.996834685870954 0.997272225611996 0.997698659284314 0.998113561803096 0.99851650494359 0.998907057287231 0.999284783895796 0.999649245878639 1

The Rk Monotonic: + Increase, - decrease, = equality.

= 1+ 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+ 29+ 30+ 31+ 32+ 33+ 34+ 35+ 36+ 37+ 38+ 39+ 40+ 41+ 42+ 43+ 44+ 45+ 46+ 47+ 48+ 49+ 50+ 51+ 52+ 53+ 54+ 55+ 56+ 57+ 58+ 59+ 60+ 61+ 62+ 63+ 64+ 65+ 66+ 67+ 68+ 69+ 70+ 71+ 72+ 73+ 74+ 75+

The accumulative frequency w = 1.08160582031538E+23

The sum of odd length pattern frequencies H_odd = 5.4080291015769E+22

The sum of even length pattern frequencies H_even = 5.4080291015769E+22

The hk Series:

(1) 88162 (2) 820414 (3) 6343921 (4) 46158618 (5) 320659309 (6) 2126662135 (7) 13407936197 (8) 79899800265 (9) 447650501360 (10) 2348766032881 (11) 11514373417314 (12) 52689673298582 (13) 225067526965647 (14) 898058317974992 (15) 3.35098822498236E+15 (16) 1.17078977654666E+16 (17) 3.83546046157292E+16 (18) 1.1797286750403E+17 (19) 3.41148254476145E+17 (20) 9.28614757944738E+17 (21) 2.38206301959175E+18 (22) 5.7643067393551E+18 (23) 1.31712189782779E+19 (24) 2.84419251006897E+19 (25) 5.8086913594715E+19 (26) 1.12274766062243E+20 (27) 2.05512250286333E+20 (28) 3.56437580505891E+20 (29) 5.86045586295229E+20 (30) 9.13836269116928E+20 (31) 1.35194063518349E+21 (32) 1.8981884386439E+21 (33) 2.53006299359521E+21 (34) 3.20205927122623E+21 (35) 3.84863789683527E+21 (36) 4.3935521478661E+21 (37) 4.76413870482027E+21 (38) 4.90702944945289E+21 (39) 4.80065694772769E+21 (40) 4.46054604470683E+21 (41) 3.93561772831354E+21 (42) 3.29671169018937E+21 (43) 2.62101823819822E+21 (44) 1.97710016165872E+21 (45) 1.41440697353311E+21 (46) 9.59161556582305E+20 (47) 6.1621198824275E+20 (48) 3.74801269040722E+20 (49) 2.15662779160028E+20 (50) 1.17294862149926E+20 (51) 6.02407697115692E+19 (52) 2.91833282271095E+19 (53) 1.33192086042178E+19 (54) 5.71903548821977E+18 (55) 2.30672959255284E+18 (56) 8.72462928735225E+17 (57) 3.08835828822501E+17 (58) 1.02090798532141E+17 (59) 3.14375852548001E+16 (60) 8.99287300417102E+15 (61) 2.38204635003018E+15 (62) 582132075762540 (63) 130704568161851 (64) 26831528576866 (65) 5007411569097 (66) 843858451958 (67) 127382126794 (68) 17055330464 (69) 2000881389 (70) 202507690 (71) 17324143 (72) 1218150 (73) 67599 (74) 2776 (75) 75 (76) 1

The hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+ 29+ 30+ 31+ 32+ 33+ 34+ 35+ 36+ 37+ 38+ 39- 40- 41- 42- 43- 44- 45- 46- 47- 48- 49- 50- 51- 52- 53- 54- 55- 56- 57- 58- 59- 60- 61- 62- 63- 64- 65- 66- 67- 68- 69- 70- 71- 72- 73- 74- 75- 76-

The hk Genuine Concavity: hk:(hk -(hk-1 + hk+1)/2 >= 0?); Concavity domain = [33, 43].

Detailed as below:

1<> 2:(-2395627.5) 3:(-17145595) 4:(-117342997) 5:(-765751067.5) 6:(-4737635618) 7:(-27605295003) 8:(-150629418513.5) 9:(-766682415213) 10:(-3632245926456) 11:(-16004846248417.5) 12:(-65601276892898.5) 13:(-250306468671140) 14:(-889969557999009) 15:(-2.95198981673845E+15) 16:(-9.14489865488914E+15) 17:(-2.64857780190192E+16) 18:(-7.17785620419068E+16) 19:(-1.8214555824824E+17) 20:(-4.3299087908921E+17) 21:(-9.64397729058168E+17) 22:(-2.01233425957972E+18) 23:(-3.93189694174449E+18) 24:(-7.18714118580676E+18) 25:(-1.22714319867515E+19) 26:(-1.95248158782805E+19) 27:(-2.88439229977343E+19) 28:(-3.93413377848903E+19) 29:(-4.90913385161801E+19) 30:(-5.51568416224334E+19) 31:(-5.407171869692E+19) 32:(-4.2813375745450 1E+19) 33:(-2.00608613398568E+19) 34:(1.27088260109896E+19) 35:(5.08321872891022E+19) 36:(8.71638470383339E+19) 37:(1.13847906160774E+20) 38:(1.2463162317891E+20) 39:(1.16869200647833E+20) 40:(9.24087066862158E+19) 41:(5.69888608654323E+19) 42:(1.83937069334968E+19) 43:(-

1.5887687725827E+19) 44:(-4.06124442069455E+19) 45:(-5.37238855874031E+19) 46:(-5.6147924305625E+19) 47:(-5.07694245687632E+19) 48:(-4.11361146606667E+19) 49:(-3.03852864352963E+19) 50:(-2.06569122858727E+19) 51:(-1.29983254769484E+19) 52:(-7.59666093078404E+18) 53:(-4.13197325344678E+18) 54:(-2.09393361016557E+18) 55:(-9.8901961592466E+17) 56:(-4.35319781952443E+17) 57:(-1.78441034811182E+17) 58:(-6.80459085065098E+16) 59:(-2.41042505133557E+16) 60:(-7.91694279824414E+15) 61:(-2.40545618993661E+15) 62:(-6742433833333473) 63:(-173777234007852) 64:(-41024461288608) 65:(-8830281945315) 66:(-1723538395987.5) 67:(-303074764417) 68:(-47636173627.5) 69:(-6628037688) 70:(-806595076) 71:(-84538777) 72:(-7477721) 73:(-542864) 74:(-31061) 75:(-1313.5)

Output for accident_dat.txt

Total number of elements n = 469

Total number of tuples u = 340183

Longest pattern length Alpha = 51

The Gk distributions:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 8 8 24 72 173 338 847 2126 5424 13454 25037 29434 35640 42540 45977 42977 35957 25896 16093 9187 4689 2322 1069 486 225 102 40 19 8 4 2 1

The Hk Series:

(1) 11500870 (2) 190126271 (3) 2048535348 (4) 16178712662 (5) 99868593350 (6) 501747368618 (7) 2109641396806 (8) 7575707412975 (9) 23596557346745 (10) 64532436426248 (11) 156494619826759 (12) 339292670918357 (13) 662267155920527 (14) 1.17088896993161E+15 (15) 1.8852993348772E+15 (16) 2.77836082215053E+15 (17) 3.76507207472581E+15 (18) 4.71292018449639E+15 (19) 5.47335759367226E+15 (20) 5.9231617931486 7E+15 (21) 5.9984917558589E+15 (22) 5.70812727877794E+15 (23) 5.12307216430698E+15 (24) 4.35034034688795E+15 (25) 3.50335749197144E+15 (26) 2.67904254984003E+15 (27) 1.94572268328735E+15 (28) 1.34084120169436E+15 (29) 875055956580332 (30) 539379672531933 (31) 313012776894420 (32) 170412248731500 (33) 86712999547974 (34) 41079409043847 (35) 18045666283976 (36) 7319881053213 (37) 2729465391540 (38) 931089778543 (39) 289014188456 (40) 81138980458 (41) 20459031625 (42) 4595198520 (43) 910273047 (44) 157094976 (45) 23255333 (46) 2893313 (47) 294207 (48) 23479 (49) 1379 (50) 53 (51) 1

The Hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22- 23- 24- 25- 26- 27- 28- 29- 30- 31- 32- 33- 34- 35- 36- 37- 38- 39- 40- 41- 42- 43- 44- 45- 46- 47- 48- 49- 50- 51-

The Hk Genuine Concavity: Hk:(Hk - (Hk-1 + Hk+1)/2 >= 0?); theoretic concavity domain = [21, 30]; exact = [16, 25].

Detailed as below:

1<> 2:(-839891838) 3:(-6135884118.5) 4:(-34779851687) 5:(-159094447290) 6:(-603007626460) 7:(-1929085993990.5) 8:(-5277391958800.5) 9:(-12457514572866.5) 10:(-25513152160504) 11:(-45417933845543.5) 12:(-70088216955286) 13:(-92823664504456.5) 14:(-102894275467252) 15:(-89325561163873) 16:(-46824882650971) 17:(19431571402343.5) 18:(93705350297363) 19:(155316604849722) 20:(187237118383095) 21:(182847219895598) 22:(147345318694995) 23:(93838351474037) 24:(37125518748737) 25:(-11333956392545.5) 26:(-45497537789368) 27:(-64219192479843.5) 28:(-69548118239482) 29:(-65054480532814) 30:(-54654694205443) 31:(-41883183737296.5) 32:(-29450639489697) 33:(-19032829339699.5) 34:(-11299923872128) 35:(-6153978764554) 36:(-3067684784545) 37:(-1396020024338) 38:(-578150011455) 39:(-217100191044.5) 40:(-73597629582.5) 41:(-22408057864) 42:(-6089453816) 43:(-1465873701) 44:(-309669214) 45:(-56738811.5) 46:(-8881457) 47:(-1164189) 48:(-124314) 49:(-10387) 50:(-637)

The Rk Series:

0.661258743034223 0.659669654543449 0.658141465063295 0.656685031958284 0.655314215959906 0.654047156055942 0.652907811677863 0.651927808078982 0.651148622656721 0.650624147205726 0.650423646437093 0.650635093421442 0.651368776120728 0.65276090286432 0.654976644410335 0.658211589674931 0.662689941194264 0.668657005852476 0.676362919352789 0.686034681672803 0.69783541843641 0.711814225724103 0.727856847392233 0.745654195061704 0.764707157627939 0.784377425463368 0.803976203159195 0.822865095523693 0.840537944733476 0.856662762258845 0.871081368229159 0.88377912861112 0.894842061853888 0.90441520981245 0.912669674289271 0.919780150823181 0.925911606151859 0.931212634215335 0.935813115744347 0.939824362739502 0.943340532325904 0.946440576340251 0.949190323549149 0.951644496165901 0.953848578875793 0.955840519155722 0.95765226524182 0.959311157488252 0.960841189267585 0.962264150943396

The Rk Monotonic: + Increase, - decrease, = equality.

= 1+ 2- 3- 4- 5- 6- 7- 8- 9- 10- 11- 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+ 29+ 30+ 31+ 32+ 33+ 34+ 35+ 36+ 37+ 38+ 39+ 40+ 41+ 42+ 43+ 44+ 45+ 46+ 47+ 48+ 49+ 50+

The accumulative frequency w = 5.96696205467851E+16

The sum of odd length pattern frequencies H_odd = 2.98348102735626E+16

The sum of even length pattern frequencies H_even = 2.98348102732224E+16

The hk Series:

(1) 340183 (2) 11160687 (3) 178965584 (4) 1869569764 (5) 14309142898 (6) 85559450452 (7) 416187918166 (8) 1693453478640 (9) 5882253934335 (10) 17714303412410 (11) 46818133013838 (12) 109676486812921 (13) 229616184105436 (14) 432650971815091 (15) 738237998116519 (16) 1.14706133676068E+15 (17) 1.63129948538985E+15 (18) 2.13377258933595E+15 (19)

2.57914759516044E+15 (20) 2.89420999851182E+15 (21) 3.02895179463686E+15 (22) 2.96953996122204E+15 (23) 2.73858731755589E+15 (24) 2.38448484675109E+15 (25) 1.96585550013686E+15 (26) 1.53750199183458E+15 (27) 1.14154055800545E+15 (28) 804182125281904 (29) 536659076412455 (30) 338396880167877 (31) 200982792364056 (32) 112029984530364 (33) 58382264201136 (34) 28330735346838 (35) 12748673697009 (36) 5296992586967 (37) 2022888466246 (38) 706576925294 (39) 224512853249 (40) 64501335207 (41) 16637645251 (42) 3821386374 (43) 773812146 (44) 136460901 (45) 20634075 (46) 2621258 (47) 272055 (48) 22152 (49) 1327 (50) 52 (51) 1

The hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22- 23- 24- 25- 26- 27- 28- 29- 30- 31- 32- 33- 34- 35- 36- 37- 38- 39- 40- 41- 42- 43- 44- 45- 46- 47- 48- 49- 50- 51-

The hk Genuine Concavity: hk:(hk -(hk-1 + hk+1)/2 >= 0?); Concavity domain = [17, 26].

Detailed as below:

1<> 2:(-78492196.5) 3:(-761399641.5) 4:(-5374484477) 5:(-29405367210) 6:(-129689080080) 7:(-473318546380) 8:(-1455767447610.5) 9:(-3821624511190) 10:(-8635890061676.5) 11:(-16877262098827.5) 12:(-28540671746716) 13:(-41547545208570) 14:(-51276119295886.5) 15:(-51618156171365.5) 16:(-37707404992507.5) 17:(-9117477658463.5) 18:(28549049060807) 19:(65156301236556) 20:(90160303613166) 21:(97076814769928.5) 22:(85770405125669.5) 23:(61574913569325.5) 24:(32263437904711.5) 25:(4862080844025.5) 26:(-16196037236571) 27:(-29301500552797) 28:(-34917691927046.5) 29:(-34630426312435.5) 30:(-30424054220378.5) 31:(-24230639985064.5) 32:(-17652543752232) 33:(-11798095737465) 34:(-7234733602234.5) 35:(-4065190269893.5) 36:(-2088788494660.5) 37:(-978896289884.5) 38:(-417123734453.5) 39:(-161026277001.5) 40:(-56073914043) 41:(-17523715539.5) 42:(-4884342324.5) 43:(-1205111491.5) 44:(-260762209.5) 45:(-48907004.5) 46:(-7831807) 47:(-1049650) 48:(-114539) 49:(-9775) 50:(-612)

Output for pumsb_star_dat.txt

Total number of elements n = 7117

Total number of tuples u = 49046

Longest pattern length Alpha = 63

The Gk distributions:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 22935 9873 3818 4765 2609 2069 1598 840 337 122 50 18 9 2 1

The Hk Series:

(1) 2475947 (2) 61354029 (3) 994833553 (4) 11871905109 (5) 111195816937 (6) 851319404422 (7) 5478723572673 (8) 30249595410859 (9) 145536526875785 (10) 617668265835985 (11) 2.33540366924541E+15 (12) 7.93100064889181E+15 (13) 2.43565338674182E+16 (14) 6.80363124277246E+16 (15) 1.73726669682348E+17 (16) 4.07257800899788E+17 (17) 8.79814160690642E+17 (18) 1.7574197680144E+18 (19) 3.25536545673323E+18 (20) 5.60654354176194E+18 (21) 8.99841235384127E+18 (22) 1.34865978950395E+19 (23) 1.89100778445261E+19 (24) 2.48447602743104E+19 (25) 3.06294523332462E+19 (26) 3.5476672516081E+19 (27) 3.86467885423445E+19 (28) 3.96329973373586E+19 (29) 3.82934583427739E+19 (30) 3.48833924491648E+19 (31) 2.99776520013333E+19 (32) 2.43152661257564E+19 (33) 1.86226459072274E+19 (34) 1.3471702716558E+19 (35) 9.20693786834402E+18 (36) 5.94511230350647E+18 (37) 3.62689357357827E+18 (38) 2.08997787870144E+18 (39) 1.13708536219218E+18 (40) 5.83709528649578E+17 (41) 2.82450446058859E+17 (42) 1.28673318925082E+17 (43) 5.51003443981372E+16 (44) 2.21365926396406E+16 (45) 8.32482138536006E+15 (46) 2.9228275554106E+15 (47) 955173953763785 (48) 289545255456520 (49) 81096958191925 (50) 20893465857031 (51) 4926251959547 (52) 1056719242649 (53) 204804664373 (54) 35570797824 (55) 5481629787 (56) 740386627 (57) 86293143 (58) 8503379 (59) 688905 (60) 44060 (61) 2086 (62) 65 (63) 1

The Hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+ 29- 30- 31- 32- 33- 34- 35- 36- 37- 38- 39- 40- 41- 42- 43- 44- 45- 46- 47- 48- 49- 50- 51- 52- 53- 54- 55- 56- 57- 58- 59- 60- 61- 62- 63-

The Hk Genuine Concavity: Hk:(Hk - (Hk-1 + Hk+1)/2 >= 0?); theoretic concavity domain = [27, 36]; exact = [23, 33].

Detailed as below:

1<> 2:(-437300721) 3:(-4971796016) 4:(-44223420136) 5:(-320399837828.5) 6:(-1943640290383) 7:(-10071733834967.5) 8:(-45258029813370) 9:(-178422403747637) 10:(-622801832224613) 11:(-1.93893078811849E+15) 12:(-5.41496811944001E+15) 13:(-1.362712267089E+16) 14:(-3.10052893471588E+16) 15:(-6.39203869814079E+16) 16:(-1.19512614286707E+17) 17:(-2.02524623766451E+17) 18:(-3.10170040697539E+17) 19:(-4.26616198154934E+17) 20:(-5.20345363525314E+17) 21:(-5.4815836455947E+17) 22:(-4.67647204144157E+17) 23:(-2.55601240148853E+17) 24:(7.49951854242324E+16) 25:(4.68735938050552E+17) 26:(8.38552078285578E+17) 27:(1.09195361562472E+18) 28:(1.16287389479943E+18) 29:(1.03526344951221E+18) 30:(7.47837277111153E+17) 31:(3.78322713872712E+17) 32:(1.51171714760417E+16) 33:(-2.7083851392974E+17) 34:(-4.43089171227771E+17) 35:(-5.01469641688197E+17) 36:(-4.71803417454677E+17) 37:(-3.90651517525681E+17) 38:(-2.92011589183787E+17) 39:(-1.99758341483328E+17) 40:(-1.26058375475942E+17) 41:(-7.37409777284711E+16) 42:(-4.01020763034159E+16) 43:(-2.03046113842242E+16) 44:(-9.57599025210805E+15) 45:(-4.20488871216552E+15) 46:(-1.71717011415133E+15) 47:(-651012451669773) 48:(-228590200521335) 49:(-74122402464850.5) 50:(-22118139218705) 51:(-6048840590293) 52:(-1508809069311) 53:(-341340355863.5) 54:(-69572349256) 55:(-12673962438.5) 56:(-2043574838) 57:(-288151860) 58:(-34987645) 59:(-3584814.5) 60:(-301435.5) 61:(-19976.5) 62:(-978.5)

The Rk Series:

0.799355650911127 0.797441359743064 0.795570614012051 0.793754180999234 0.792003935231445
0.790332862627539 0.788755029462367 0.787285506161521 0.785940235398599 0.784735834312018
0.78368932235445 0.782817769858833 0.782137868362556 0.781665432451595 0.781414854427055
0.781398546973771 0.781626423970857 0.78210548344993 0.782839566398919 0.78382936695266
0.785072759884749 0.786565487599363 0.788302210448135 0.790277873459113 0.792489284825161
0.794936744564861 0.797625514522334 0.800566891621507 0.803778641516764 0.807284573837009
0.811113096004816 0.815294672530897 0.819858242566212 0.824826809129727 0.830212599248142
0.836012380588156 0.842203663140048 0.84874255517094 0.855563924013948 0.862584207594161
0.869706763894576 0.876829133362802 0.88385116897483 0.890682796103732 0.897250265368588
0.903500120543403 0.909400599698905 0.914940670244909 0.920127247453258 0.92498130812786
0.929533599935222 0.933820517630707 0.937880535278096 0.941751401362476 0.94546815279118
0.949061895815607 0.952559235596893 0.955982204250804 0.959348531364992 0.962672113784234
0.965963566634708 0.969230769230769

The Rk Monotonic: + Increase, - decrease, = equality.

= 1+ 2- 3- 4- 5- 6- 7- 8- 9- 10- 11- 12- 13- 14- 15- 16- 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+ 29+ 30+ 31+ 32+ 33+ 34+ 35+ 36+ 37+ 38+ 39+ 40+ 41+ 42+ 43+ 44+ 45+ 46+ 47+ 48+ 49+ 50+ 51+ 52+ 53+ 54+ 55+ 56+ 57+ 58+ 59+ 60+ 61+ 62+

The accumulative frequency w = 4.05464141001747E+20

The sum of odd length pattern frequencies H_odd = 2.02732070500873E+20

The sum of even length pattern frequencies H_even = 2.02732070500873E+20

The hk Series:

(1) 49046 (2) 2426901 (3) 58927128 (4) 935906425 (5) 10935998684 (6) 100259818253 (7) 751059586169 (8) 4727663986504 (9) 25521931424355 (10) 120014595451430 (11) 497653670384555 (12) 1.83774999886086E+15 (13) 6.09325065003095E+15 (14) 1.82632832173873E+16 (15) 4.97730292103373E+16 (16) 1.23953640472011E+17 (17) 2.83304160427777E+17 (18) 5.96510000262865E+17 (19) 1.16090976775153E+18 (20) 2.0944556889817E+18 (21) 3.51208785278024E+18 (22) 5.48632450106103E+18 (23) 8.00027339397851E+18 (24) 1.09098044505476E+19 (25) 1.39349558237628E+19 (26) 1.66944965094834E+19 (27) 1.87821760065975E+19 (28) 1.9864612535747E+19 (29) 1.97683848016116E+19 (30) 1.85250735411623E+19 (31) 1.63583189080025E+19 (32) 1.36193330933308E+19 (33) 1.06959330324256E+19 (34) 7.92671287480185E+18 (35) 5.54498984175611E+18 (36) 3.6619480265879E+18 (37) 2.28316427691856E+18 (38) 1.34372929665971E+18 (39) 7.46248582041731E+17 (40) 3.9083678015045E+17 (41) 1.92872748499128E+17 (42) 8.95776975597307E+16 (43) 3.90956213653515E+16 (44) 1.60047230327857E+16 (45) 6.13186960685486E+15 (46) 2.1929517785052E+15 (47) 729875776905397 (48) 225298176858388 (49) 64247078598132 (50) 16849879593793 (51) 4043586263238 (52) 882665696309 (53) 174053546340 (54) 30751118033 (55) 4819679791 (56) 661949996 (57) 78436631 (58) 7856512 (59) 646867 (60) 42038 (61) 2022 (62) 64 (63) 1

The hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+ 29- 30- 31- 32- 33- 34- 35- 36- 37- 38- 39- 40- 41- 42- 43- 44- 45- 46- 47- 48- 49- 50- 51- 52- 53- 54- 55- 56- 57- 58- 59- 60- 61- 62- 63-

The hk Genuine Concavity: hk:(hk -(hk-1 + hk+1)/2 >= 0?); Concavity domain = [24, 33].

Detailed as below:

1<> 2:(-27061186) 3:(-410239535) 4:(-4561556481) 5:(-39661863655) 6:(-280737974173.5) 7:(-1662902316209.5) 8:(-8408831518758) 9:(-36849198294612) 10:(-141573205453025) 11:(-481228626771588) 12:(-1.4577021613469E+15) 13:(-3.95726595809311E+15) 14:(-9.66985671279687E+15) 15:(-2.13354326343619E+16) 16:(-4.2584954347046E+16) 17:(-7.69276599396608E+16) 18:(-1.25596963826791E+17) 19:(-1.84573076870748E+17) 20:(-2.42043121284186E+17) 21:(-2.78302242241128E+17) 22:(-2.69856122318343E+17) 23:(-1.97791081825811E+17) 24:(-5.78101583230403E+16) 25:(1.32805343747271E+17) 26:(3.35930594303277E+17) 27:(5.02621483982299E+17) 28:(5.89332131642434E+17) 29:(5.73541763156984E+17) 30:(4.61721686355218E+17) 31:(2.8611559075594E+17) 32:(9.22071231167693E+16) 33:(-7.70899516407296E+16) 34:(-1.93748562289009E+17) 35:(-2.49340608938763E+17) 36:(-2.52129032749433E+17) 37:(-2.19674384705243E+17) 38:(-1.70977132820439E+17) 39:(-1.21034456363348E+17) 40:(-7.872388511998E+16) 41:(-4.7334490355962E+16) 42:(-2.64064873725092E+16) 43:(-1.36955889309067E+16) 44:(-6.60902245331746E+15) 45:(-2.96696779879059E+15) 46:(-1.23792091337493E+15) 47:(-479249200776396) 48:(-171763250893376) 49:(-56826949627958.5) 50:(-17295452836892) 51:(-4822686381813) 52:(-1226154208480) 53:(-282654860831) 54:(-58685495032.5) 55:(-10886854223.5) 56:(-1787108215) 57:(-256466623) 58:(-31685237) 59:(-3302408) 60:(-282406.5) 61:(-19029) 62:(-947.5)

Output for T40I10D100K_dat.txt

Total number of elements n = 1000

Total number of tuples u = 100000

Longest pattern length Alpha = 77

The Gk distributions:

0 0 0 1 0 0 2 1 5 5 16 24 17 31 41 71 124 164 258 306 439 593 723 923 1074 1366 1588 1946 2217 2707 2816 3164 3602 3753 3876 4232 4495 4555 4445 4725 4555 4414 4300 4062 3829 3441 3135 2822 2615 2225 1936 1657 1429 1173 909 737 639 472 336 275 208 152 127 79 47 38 32 15 8 8 9 6 1 2 0 0 2

The Hk Series:

(1) 3960507 (2) 80096632 (3) 1099569571 (4) 11498654533 (5) 97500349307 (6) 697040556063 (7) 4315006899213 (8) 23581577918301 (9) 115450483283408 (10) 512200640492561 (11) 2.07838480923294E+15 (12) 7.77254238471906E+15 (13) 2.69602689096333E+16 (14)

8.72093105225744E+16 (15) 2.64298549451808E+17 (16) 7.53449913781887E+17 (17)
2.02737315009449E+18 (18) 5.16427718468851E+18 (19) 1.24842740049338E+19 (20)
2.87009431427177E+19 (21) 6.28554616251805E+19 (22) 1.3130647241166E+20 (23)
2.61920911124247E+20 (24) 4.99250474551047E+20 (25) 9.09809445277863E+20 (26)
1.58561966617132E+21 (27) 2.64318653170815E+21 (28) 4.21450413899866E+21 (29)
6.42723570378944E+21 (30) 9.37347701999184E+21 (31) 1.30705726297083E+22 (32)
1.74224680567533E+22 (33) 2.21941675425744E+22 (34) 2.70126196187154E+22 (35)
3.14029467104375E+22 (36) 3.48593998907014E+22 (37) 3.69383350456697E+22 (38)
3.73504477947147E+22 (39) 3.6026185483506E+22 (40) 3.31341422800792E+22 (41)
2.90459517099384E+22 (42) 2.42577206876913E+22 (43) 1.9290953957398E+22 (44)
1.4600319669151E+22 (45) 1.05104249632529E+22 (46) 7.19197535857808E+21 (47)
4.6745614248956E+21 (48) 2.88379514993618E+21 (49) 1.68714820765631E+21 (50)
9.35204918695762E+20 (51) 4.9066566544455E+20 (52) 2.43392596483649E+20 (53)
1.14009537719929E+20 (54) 5.03619463496462E+19 (55) 2.09481653579947E+19 (56)
8.19139092845594E+18 (57) 3.00569848327089E+18 (58) 1.03282769896598E+18 (59)
3.31604783678242E+17 (60) 9.92248748007835E+16 (61) 2.75920564468882E+16 (62)
7.10728187982204E+15 (63) 1.68956517387226E+15 (64) 369111622206975 (65) 73743280712269 (66)
13396208900647 (67) 2197849349159 (68) 323037347878 (69) 42117832681 (70) 4811996111 (71)
474320277 (72) 39521111 (73) 2706699 (74) 146302 (75) 5852 (76) 154 (77) 2

The Hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+
27+ 28+ 29+ 30+ 31+ 32+ 33+ 34+ 35+ 36+ 37+ 38+ 39- 40- 41- 42- 43- 44- 45- 46- 47- 48- 49- 50- 51-
52- 53- 54- 55- 56- 57- 58- 59- 60- 61- 62- 63- 64- 65- 66- 67- 68- 69- 70- 71- 72- 73- 74- 75- 76- 77-

The Hk Genuine Concavity: Hk:(Hk - (Hk-1 + Hk+1)/2 >= 0?); theoretic concavity domain = [34, 43]; exact = [33, 43].

Detailed as below:

1<> 2:(-471668407) 3:(-4689806011.5) 4:(-37801304906) 5:(-256769255991) 6:(-1509213068197) 7:(-
7824302337969) 8:(-36301167173009.5) 9:(-152440625922023) 10:(-584717005765613) 11:(-
2.06398670337287E+15) 12:(-6.74678447471404E+15) 13:(-2.05306575440135E+16) 14:(-
5.8420098658146E+16) 15:(-1.56031062700423E+17) 16:(-3.92385935991259E+17) 17:(-
9.31490399140712E+17) 18:(-2.09154639282562E+18) 19:(-4.44833615876934E+18) 20:(-
8.96892467233942E+18) 21:(-1.71482461520083E+19) 22:(-3.10817139630536E+19) 23:(-
5.33575623571069E+19) 24:(-8.66147036500077E+19) 25:(-1.3262562508332E+20) 26:(-
1.90878322321689E+20) 27:(-2.56875370876839E+20) 28:(-3.20706978750133E+20) 29:(-
3.66754875705814E+20) 30:(-3.75427146757043E+20) 31:(-3.27399908664255E+20) 32:(-
2.09902029388026E+20) 33:(-2.33762951599835E+19) 34:(2.14062492209486E+20)
35:(4.66936955729053E+20) 36:(6.88759012647823E+20) 37:(8.33411202961639E+20)
38:(8.68187530126885E+20) 39:(7.83890446109018E+20) 40:(5.98073683357018E+20)
41:(3.5002022605313E+20) 42:(8.92678540231117E+19) 43:(-1.38066221023168E+20) 44:(-
3.00369791174419E+20) 45:(-3.85722550611654E+20) 46:(-4.00517835496172E+20) 47:(-
3.63323829361528E+20) 48:(-2.97059666339774E+20) 49:(-2.22351826659664E+20) 50:(-
1.53702017854666E+20) 51:(-9.86330921451557E+19) 52:(-5.89450050985906E+19) 53:(-
3.28677336967184E+19) 54:(-1.71169051893157E+19) 55:(-8.32850328105641E+18) 56:(-
3.78554099217683E+18) 57:(-1.60641083044007E+18) 58:(-6.35823934508589E+17) 59:(-
2.34421503205138E+17) 60:(-8.03735452617814E+16) 61:(-2.55740218934146E+16) 62:(-
7.53352893055817E+15) 63:(-2.04863157714224E+15) 64:(-512542605085291) 65:(-117510634841542)
66:(-24574356130067) 67:(-4661773775103.5) 68:(-796946243042) 69:(-121806839313.5) 70:(-

16484080368) 71:(-1951438334) 72:(-198992377) 73:(-17127007.5) 74:(-1209973.5) 75:(-67376) 76:(-2773)

The Rk Series:

0.532206130493686 0.549121501638171 0.565265636725842 0.580772828486351 0.595759024640537 0.610327780377984 0.62457315580858 0.638581469879227 0.652432293565568 0.666198613909566 0.679946041857347 0.693731023265632 0.707598160120602 0.721576924331548 0.735678223852825 0.749891428293542 0.764182536073514 0.778494120578493 0.792747495116936 0.806847190367279 0.820687403393911 0.834159663435217 0.847160690026802 0.859599380655357 0.871402069082201 0.882515571705354 0.892907969046751 0.902567435357276 0.911499656693908 0.919724449254064 0.927272134279971 0.934180095009313 0.940489786208399 0.946244325093674 0.951486683594253 0.956258430888995 0.960598937691932 0.964544941509479 0.968130376204133 0.971386382222729 0.97434143037415 0.977021508618985 0.979450336062951 0.9816495804161 0.983639064462367 0.985436953827253 0.987059922953293 0.988523299152587 0.989841186332512 0.991026570850394 0.992091412237422 0.993046721462159 0.993902629138512 0.994668445730243 0.99535271543366 0.995963265072766 0.99650724903779 0.9969911910463 0.997421023305281 0.997802123497653 0.998139349900338 0.99843707485557 0.998699216755331 0.998929270654587 0.999130337596984 0.999305152713364 0.999456112135685 0.999585298755041 0.999694506841141 0.99978526553124 0.99985886118885 0.999916358626659 0.999958621183959 0.999986329646895 1 1

The Rk Monotonic: + Increase, - decrease, = equality.

= 1+ 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+ 29+ 30+ 31+ 32+ 33+ 34+ 35+ 36+ 37+ 38+ 39+ 40+ 41+ 42+ 43+ 44+ 45+ 46+ 47+ 48+ 49+ 50+ 51+ 52+ 53+ 54+ 55+ 56+ 57+ 58+ 59+ 60+ 61+ 62+ 63+ 64+ 65+ 66+ 67+ 68+ 69+ 70+ 71+ 72+ 73+ 74+ 75+ 76-

The accumulative frequency w = 4.31580101724358E+23

The sum of odd length pattern frequencies H_odd = 2.15790050862179E+23

The sum of even length pattern frequencies H_even = 2.15790050862179E+23

The hk Series:

(1) 100000 (2) 3860507 (3) 76236125 (4) 1023333446 (5) 10475321087 (6) 87025028220 (7) 610015527843 (8) 3704991371370 (9) 19876586546931 (10) 95573896736477 (11) 416626743756084 (12) 1.66175806547686E+15 (13) 6.1107843192422 1E+15 (14) 2.08494845903911E+16 (15) 6.63598259321834E+16 (16) 1.97938723519624E+17 (17) 5.55511190262263E+17 (18) 1.47186195983222E+18 (19) 3.69241522485629E+18 (20) 8.79185878007748E+18 (21) 1.99090843626402E+19 (22) 4.29463772625403E+19 (23) 8.83600951491197E+19 (24) 1.73560815975127E+20 (25) 3.2568965857592E+20 (26) 5.84119786701943E+20 (27) 1.00149987946938E+21 (28) 1.64168665223878E+21 (29) 2.57281748675989E+21 (30) 3.85441821702955E+21 (31) 5.51905880296229E+21 (32) 7.55151382674604E+21 (33) 9.87095423000729E+21 (34) 1.23232133125671E+22 (35) 1.46894063061483E+22 (36) 1.67135404042891E+22 (37) 1.81458594864123E+22 (38) 1.87924755592574E+22 (39) 1.85579722354573E+22 (40) 1.74682132480487E+22 (41) 1.56659290320305E+22 (42)

1.33800226779079E+22 (43) 1.08776980097834E+22 (44) 8.41325594761459E+21 (45)
6.18706372153644E+21 (46) 4.32336124171646E+21 (47) 2.86861411686162E+21 (48)
1.80594730803398E+21 (49) 1.0778478419022E+21 (50) 6.09300365754111E+20 (51)
3.25904552941651E+20 (52) 1.64761112502899E+20 (53) 7.86314839807498E+19 (54)
3.53780537391793E+19 (55) 1.49838926104668E+19 (56) 5.96427274752782E+18 (57)
2.22711818092812E+18 (58) 7.78580302342771E+17 (59) 2.54247396623204E+17 (60)
7.73573870550372E+16 (61) 2.18674877457461E+16 (62) 5.72456870114201E+15 (63)
1.38271317868002E+15 (64) 306851995192239 (65) 62259627014736 (66) 11483653697533 (67)
1912555203114 (68) 285294146045 (69) 37743201833 (70) 4374630848 (71) 437365263 (72) 36955014
(73) 2566097 (74) 140602 (75) 5700 (76) 152 (77) 2

The hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+
27+ 28+ 29+ 30+ 31+ 32+ 33+ 34+ 35+ 36+ 37+ 38+ 39- 40- 41- 42- 43- 44- 45- 46- 47- 48- 49- 50- 51-
52- 53- 54- 55- 56- 57- 58- 59- 60- 61- 62- 63- 64- 65- 66- 67- 68- 69- 70- 71- 72- 73- 74- 75- 76- 77-

The hk Genuine Concavity: hk:(hk -(hk-1 + hk+1)/2 >= 0?); Concavity domain = [33, 43].

Detailed as below:

1<> 2:(-34307555.5) 3:(-437360851.5) 4:(-4252445160) 5:(-33548859746) 6:(-223220396245) 7:(-
1285992671952) 8:(-6538309666017) 9:(-29762857506992.5) 10:(-122677768415031) 11:(-
462039237350582) 12:(-1.60194746602229E+15) 13:(-5.14483700869175E+15) 14:(-
1.53858205353217E+16) 15:(-4.30342781228242E+16) 16:(-1.12996784577599E+17) 17:(-
2.7938915141366E+17) 18:(-6.52101247727052E+17) 19:(-1.43944514509857E+18) 20:(-
3.00889101367078E+18) 21:(-5.96003365866864E+18) 22:(-1.11882124933397E+19) 23:(-
1.98935014697139E+19) 24:(-3.3464060887393E+19) 25:(-5.31506427626147E+19) 26:(-
7.9474982320705E+19) 27:(-1.11403340000984E+20) 28:(-1.45472030875855E+20) 29:(-
1.75234947874278E+20) 30:(-1.91519927831536E+20) 31:(-1.83907218925509E+20) 32:(-
1.43492689738746E+20) 33:(-6.64093396492825E+19) 34:(4.30330444893032E+19)
35:(1.71029447720183E+20) 36:(2.95907508008868E+20) 37:(3.92851504638957E+20)
38:(4.4055969832268E+20) 39:(4.27627831804205E+20) 40:(3.56262614304817E+20)
41:(2.41811069052199E+20) 42:(1.08209157000934E+20) 43:(-1.89413029778261E+19) 44:(-
1.19124918045337E+20) 45:(-1.81244873129084E+20) 46:(-2.04477677482568E+20) 47:(-
1.960401580136u5E+20) 48:(-1.67283671347923E+20) 49:(-1.29775994991852E+20) 50:(-
9.25758316678117E+19) 51:(-6.11261861868545E+19) 52:(-3.75069059583011E+19) 53:(-
2.14380991402894E+19) 54:(-1.1429634556429E+19) 55:(-5.68727063288676E+18) 56:(-
2.64123264816965E+18) 57:(-1.14430834400718E+18) 58:(-4.62102486432889E+17) 59:(-
1.737214480757E+17) 60:(-6.07000551294379E+16) 61:(-1.96734901323435E+16) 62:(-
5.90053176107107E+15) 63:(-1.6329971694871E+15) 64:(-415634407655141) 65:(-96908197430150)
66:(-20602437411392) 67:(-3971918718675) 68:(-689855056428.5) 69:(-107091186613.5) 70:(-
14715652700) 71:(-1768427668) 72:(-183010666) 73:(-15981711) 74:(-1145296.5) 75:(-64677) 76:(-
2699)

Output for T1014D100k_dat.txt

Total number of elements n = 1000

Total number of tuples u = 100000

Longest pattern length Alpha = 29

The Gk distributions:

128 545 1607 3287 4849 6525 7990 9759 10471 10892 10242 8719 7364 5685 4185 2932 1963 1255 756 388 215 117 67 27 17 11 2 0 2

The Hk Series:

(1) 1010228 (2) 5270095 (3) 18790249 (4) 51202603 (5) 113283212 (6) 211446703 (7) 341991592 (8) 488936418 (9) 627432107 (10) 731187069 (11) 780208311 (12) 765930113 (13) 692835561 (14) 576788640 (15) 440503927 (16) 307192014 (17) 194493524 (18) 111044874 (19) 56721409 (20) 25679299 (21) 10189328 (22) 3495680 (23) 1019555 (24) 247387 (25) 48507 (26) 7373 (27) 814 (28) 58 (29) 2

The Hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12- 13- 14- 15- 16- 17- 18- 19- 20- 21- 22- 23- 24- 25- 26- 27- 28- 29-

The Hk Genuine Concavity: Hk:(Hk - (Hk-1 + Hk+1)/2 >= 0?); theoretic concavity domain = [11, 18]; exact = [7, 15].

Detailed as below:

1<> 2:(-4630143.5) 3:(-9446100) 4:(-14834127.5) 5:(-18041441) 6:(-16190699) 7:(-8199968.5) 8:(4224568.5) 9:(17370363.5) 10:(27366860) 11:(31649720) 12:(29408177) 13:(21476184.5) 14:(10118896) 15:(-1486400) 16:(-10306711.5) 17:(-14624920) 18:(-14562592.5) 19:(-11640677.5) 20:(-7776069.5) 21:(-4398161.5) 22:(-2108761.5) 23:(-851978.5) 24:(-286644) 25:(-78873) 26:(-17287.5) 27:(-2901.5) 28:(-350)

The Rk Series:

0.372624157262378 0.396160874603673 0.41922401020133 0.442490050749959 0.466632917770728 0.492248855567246 0.519881380858361 0.549968185083263 0.582682222381074 0.617761948514005 0.654466336918235 0.691728161626673 0.728441333570636 0.763718104780982 0.796988553131721 0.827943675745325 0.856415717985551 0.882286046073426 0.905453494640798 0.925846872481475 0.943449852630124 0.958316101179578 0.970568532349898 0.980387004976009 0.987991423918197 0.993625389936254 0.997542997542998 1

The Rk Monotonic: + Increase, - decrease, = equality.

= 1+ 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+ 18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+

The accumulative frequency w = 6556956652

The sum of odd length pattern frequencies H_odd = 3278528326

The sum of even length pattern frequencies H_even = 3278428326

The hk Series:

(1) 100000 (2) 910228 (3) 4359867 (4) 14430382 (5) 36772221 (6) 76510991 (7) 134935712 (8) 207055880 (9) 281880538 (10) 345551569 (11) 385635500 (12) 394572811 (13) 371357302 (14) 321478259 (15) 255310381 (16) 185193546 (17) 121998468 (18) 72495056 (19) 38549818 (20) 18171591 (21) 7507708 (22) 2681620 (23) 814060 (24) 205495 (25) 41892 (26) 6615 (27) 758 (28) 56 (29) 2

The hk Quasi Concavity: + Increase, - decrease, = equality.

1<> 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13- 14- 15- 16- 17- 18- 19- 20- 21- 22- 23- 24- 25- 26- 27- 28- 29-

The hk Genuine Concavity: hk:(hk -(hk-1 + hk+1)/2 >= 0?); Concavity domain = [8, 16].

Detailed as below:

1<> 2:(-1319705.5) 3:(-3310438) 4:(-6135662) 5:(-8698465.5) 6:(-9342975.5) 7:(-6847723.5) 8:(-1352245) 9:(5576813.5) 10:(11793550) 11:(15573310) 12:(16076410) 13:(13331767) 14:(8144417.5) 15:(1974478.5) 16:(-3460878.5) 17:(-6845833) 18:(-7779087) 19:(-6783505.5) 20:(-4857172) 21:(-2918897.5) 22:(-1479264) 23:(-629497.5) 24:(-222481) 25:(-64163) 26:(-14710) 27:(-2577.5) 28:(-324)

Produced on 9/11/2009 1:08:53 AM

By Tongyuan Wang, Dept of CS, Concordia University