

# On Exploring the Power-Law Relationship in the Itemset Support Distribution

Kun-Ta Chuang<sup>1</sup>, Jiun-Long Huang<sup>2</sup>, and Ming-Syan Chen<sup>1</sup>

<sup>1</sup> Graduate Institute of Communication Engineering,  
National Taiwan University, Taipei, Taiwan, ROC  
doug@arbor.ee.ntu.edu.tw, mschen@cc.ee.ntu.edu.tw

<sup>2</sup> Department of Computer Science,  
National Chiao Tung University, Hsinchu, Taiwan, ROC  
jluang@csie.nctu.edu.tw

**Abstract.** We identify and explore in this paper an important phenomenon which points out that the power-law relationship appears in the distribution of itemset supports. Characterizing such a relationship will benefit many applications such as providing the direction of tuning the performance of the frequent-itemset mining. Nevertheless, due to the explosive number of itemsets, it will be prohibitively expensive to retrieve characteristics of the power-law relationship in the distribution of itemset supports. As such, we also propose in this paper a valid and cost-effective algorithm, called algorithm *PPL*, to extract characteristics of the distribution without the need of discovering all itemsets in advance. Experimental results demonstrate that algorithm *PPL* is able to efficiently extract the characteristics of the power-law relationship with high accuracy.

## 1 Introduction

The importance of mining frequent itemsets has been recognized in various applications, including web log mining, DNA sequence mining, frequent episodes mining, periodic patterns, to name a few [8]. Due to the data-driven nature of mining algorithms, it is believed in the literature that the parameter tuning of the designed algorithm is usually requested in order to achieve the better result on the targeted applications. It is beyond dispute that the deeper knowledge about the characteristics of your data will lead to the better execution efficiency and the better interpretation of the mining result. As such, a mechanism to precisely estimate the data characteristics is usually deemed as an important pre-processing means for mining applications.

Recent research advances in frequent-itemset mining algorithms are thus in the direction of discovering characteristics of real datasets. For example, the works in [7] and [11] both seek the relationship between different itemset lengths in the targeted dataset. Such relationships can be further utilized to control the mining process [7], or to generate the realistic synthetic datasets for the system parameter tuning [11].

To provide better understanding on real datasets, we in this paper investigate the more important characteristic in real datasets, named the *itemset support*

*distribution*. The *itemset support distribution* refers to the distribution of the count of itemsets versus the itemset support, where an itemset complies with the definition in [1]. Explicitly, we shall study the relationship between the value of support, say 0.01, and the number of itemsets having the support 0.01 in the dataset. The *itemset support distribution*, which is indeed a kind of the probability density function, will state the degree of the cohesion between different items in the dataset. To the best of our knowledge, this fundamental question has not been formally addressed.

Inspired by the power-law relationship observed in many distributions of single words (users, web pages) [3][17], it is important to examine whether the *itemset support distribution* also follows the power-law relationship. From observations on various retail datasets and as validated by our empirical studies later, it is amazingly found that the power-law relationship indeed also appears in the *itemset support distribution* and we can characterize that by the Zipf distribution [17].

However, to find the parameters characterizing the *itemset support distribution* will be more challenging than to find the parameters in the distribution of single items since all itemsets need to be retrieved. The extremely large time and memory consumption cannot be avoided due to the itemset combinational explosion. Note that the costly process will drastically decrease the practicability of knowing the characteristics of the *itemset support distribution*. To remedy this, we also propose in this paper a valid and cost-effective algorithm, called algorithm *PPL* (standing for **P**redict the **P**ower-**L**aw relationship), to correctly estimate the parameters of the *itemset support distribution* from a sample dataset while avoiding the need of generating all itemsets. As shown in our empirical studies, algorithm *PPL* is able to efficiently and precisely extract the characteristics of the power-law relationship. Hence algorithm *PPL* can be utilized as an excellent pre-processing step for extensive applications of mining frequent patterns.

Our contributions are to solidly study issues related to the power-law relationship in the *itemset support distribution*. More precisely:

- (1) We first formalize the problem of the *itemset support distribution* and explore the important phenomenon that the distribution follows the Zipf distribution.
- (2) We present a valid and cost-effective algorithm, called algorithm *PPL*, to identify characteristics of the *itemset support distribution* without the need of discovering all itemsets in advance.
- (3) We complement our analytical and algorithmic results by a thorough empirical study on real data and demonstrate that the *PPL* algorithm is able to accurately the characterization of the *itemset support distribution*.

We then individually present these issues in the following sections.

## 2 Identify the Power-Law Relationship in the Itemset Support Distribution

### 2.1 Review of the Power-Law Relationship

Since the first observation of the power-law relationship in [17], which discovered the frequency of the  $n^{th}$  most-frequently-used word in the natural

language is approximately inversely proportional to  $n$ , the power-law relationship has been successively discovered in many real world data, including WWW characteristics, Internet topology, to name a few<sup>1</sup>. Specifically, the power-law relationship can be characterized by several mathematical models, including the well-known Zipf distribution and its variations such as the DGX distribution [2]. Among them, the Zipf distribution is the most widely used form due to its simplicity, as shown by  $f_i \propto (1/r_i^\phi)$ , where  $f_i$  denotes the frequency of words (users, events, ...) that are ranked as the  $r_i^{\text{th}}$  most frequent words (users, events, ...) in the dataset, and  $\phi$  is the parameter characterizing the skewness of the distribution. In practice, the Zipf distribution can be further extended to characterize the "count-frequency" relationship, which is stated as  $f_i \propto (1/c_i^\phi)$ , where  $f_i$  is the count of distinct words that appear  $c_i$  times in the dataset [2]. Without loss of generality, we will discuss the "count-frequency" relationship in the sequel because the "count-frequency" relationship can be deemed as a kind of the probability density function, which is more desirable.

In essence, the Zipf distribution is often demonstrated by scatterplotting the data with the x axis being  $\log(c_i)$  and the y axis being  $\log(f_i)$ . The distribution will be deemed following the power-law relationship if the points in the log-log plot are close to a single straight line, as shown by

$$\log(f_i) = \theta \log(c_i) + \Omega. \quad (1)$$

In particular, the slope  $\theta$  and the Y-intercept  $\Omega$  in Eq. 1 can be estimated by the linear regression<sup>2</sup>:

$$\theta = \frac{\sum_{i=1}^k \log(c_i) \log(f_i) - \frac{(\sum_{i=1}^k \log(c_i)) \times (\sum_{i=1}^k \log(f_i))}{k}}{\sum_{i=1}^k \log^2(c_i) - \frac{(\sum_{i=1}^k \log(c_i))^2}{k}}, \quad (2)$$

$$\Omega = \frac{\sum_{i=1}^k \log(f_i)}{k} - \theta \times \frac{\sum_{i=1}^k \log(c_i)}{k}, \quad (3)$$

where  $k$  denotes the number of points in the log-log plot. Note that the linear regression technique is a method based on the least-square errors. The correlation coefficient<sup>3</sup> (or said the *goodness of fit* of the regression line) can be utilized to examine whether those points in the log-log plot exactly lie in the line  $\log(f_i) = \theta \log(c_i) + \Omega$  or not [12]. Due to space limitations, we only describe the Zipf

<sup>1</sup> See <http://www.nslj-genetics.org/wli/zipf/> for the power-law references from different domains.

<sup>2</sup> Other measurements to estimate the parameters of the power-law distribution include the non-linear regression and the maximum likelihood estimation. Among them, the linear regression is the most widely utilized approach due to its feasibility and simplicity.

<sup>3</sup> For convenience of discussion, we will postpone the formula of the correlation coefficient to Eq. 5 in Section 3.3.

distribution here. For details of the regression technique, which will be out of scope for this paper, the reader is asked to follow the pointers in some well-known materials such as [12].

Note that previous observations mostly concentrate on the power-law relationship in the distribution consisting of single events, e.g., single words or single items [3][17]. Naturally, it is important to investigate whether the prevalent power-law relationship also appears in the support distribution of units consisting of a set of words or items. Such cases were first investigated in the computational linguistics literature [6], where the power-law relationship of N-grams had been demonstrated (N-grams denote phrases consisting of N consecutive words). Their studies show that the "count-frequency" relationship of N-grams (with a fixed  $N$ ) follows the Zipf distribution.

**2.2 Observations on the Itemset Support Distribution**

In this paper, our first goal is to investigate whether the power-law relationship appears in the distribution of itemset supports in real datasets, where an itemset complies with the definition in [1]. Specifically, let  $\mathcal{I} = \{x_1, x_2, \dots, x_m\}$  be a set of distinct items in the dataset. A set  $X \subseteq \mathcal{I}$  with  $k = |X|$  is called a  $k$ -itemset or simply an itemset. Let the support of an itemset  $X$  in the database  $D$  be the fraction of transactions in  $D$  that contain  $X$ <sup>4</sup>. We would like to investigate whether the support distribution of itemsets follows the Zipf distribution, as the form shown by

$$\log(f_i) = \theta \log(s_i) + \Omega, \tag{4}$$

where  $s_i$  denotes the support of itemsets and  $f_i$  denotes the frequency of itemsets whose supports are  $s_i$ . Note that the "support-frequency" relationship in Eq. 4 is physically equivalent to the "count-frequency" relationship since the "count" presents the absolute support count. For interest of space, we in this paper concentrate on the investigation of retail datasets, which are skewed and sparse, and most association-rule discovery algorithms were designed for such types of data [16] (interested readers can find observations on other types of real datasets in <http://arbor.ee.ntu.edu.tw/~doug/paper/PPL/index.html>).

**Table 1.** Parameters of real datasets

Dataset	$I_s$	$ D $	$T_{max}$	$T_{avg}$
BMS-POS	1,657	515,596	164	6.5
Retail	16,470	88,162	76	10.3
3C_chain	130,108	8,000,000	87	5.4
Book	12,082	100,000	13	2.3

To examine whether the support distribution of itemsets in retail datasets follows the Zipf distribution, four real datasets are investigated in this paper, in-

<sup>4</sup> The support is considered as the *relative* occurrence frequency. Note that it is defined in some literature as the *absolute* one, i.e., the occurrence frequency in the database.

cluding two well-known retail benchmark datasets<sup>5</sup>, and two transaction datasets from a 3C chain store and a large book store in Taiwan. Those datasets are summarized in Table 1, where  $I_s$  denotes the distinct items in the dataset,  $|D|$  denotes the number of transactions,  $T_{\max}$  denotes the maximum itemset length and  $T_{avg}$  denotes the average itemset length. Furthermore, we execute algorithm FP-growth downloaded from Christian Borgelt's website<sup>6</sup> to obtain itemsets with their supports. Since the number of all itemsets is extremely large (there are  $2^{I_s} - 1$  possible itemsets at most), it is very difficult to discover all itemsets in a reasonable execution time. For efficiency reasons, we did not retrieve all itemsets in the BMS-POS and the 3C\_chain datasets, but instead retrieve itemsets whose support counts exceed 30, where 30 is a sufficient number in the statistical sense [12].

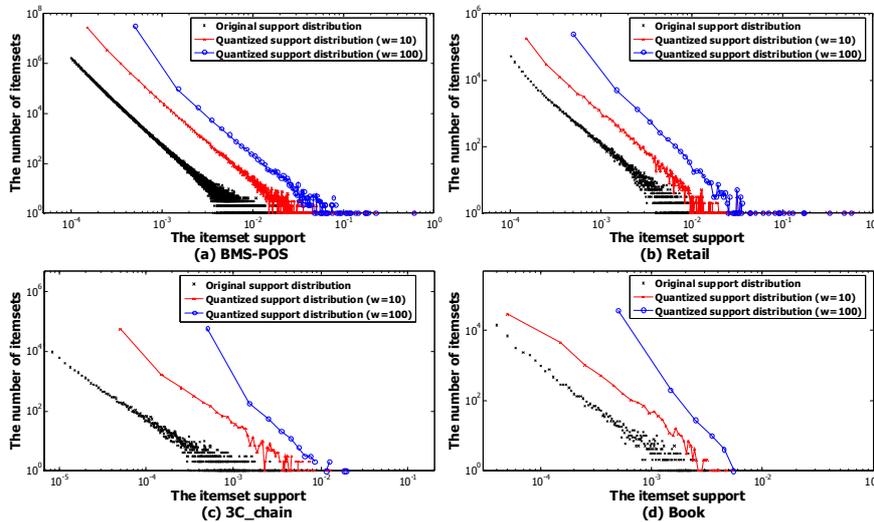


Fig. 1. The support distribution of four real datasets

The observations are shown in Figure 1, where the curve of the *original support distribution* presents the log-log relationship of the itemset support versus the number of itemsets with the corresponding support (the curve of the *quantized support distribution* will be discussed in the next section). As can be seen, the log-log plot is very Zipf-like, meaning that the power-law relationship indeed appears in the distribution of the itemset support. In addition, the "top-concavity"<sup>7</sup> phenomenon, which is prevalent in the distribution of single items [2][17], is

<sup>5</sup> Downloaded from the website, <http://fimi.cs.helsinki.fi/data/>, of the ICDM workshop on Frequent Itemset Mining, 2003.

<sup>6</sup> The URL is <http://fuzzy.cs.uni-magdeburg.de/~borgelt/fpgrowth.html>

<sup>7</sup> The "top concavity" phenomenon refers to that the top part of the log-log curve tilts vertically (with relatively concave shapes).

insignificant in the distribution of itemset supports. As such, the Zipf distribution is enough to correctly characterize the power-law relationship in the itemset support distribution. We accordingly demonstrate the fact that the power-law relationship appears in the itemset support distribution.

### 3 Design of Algorithm PPL

As mentioned above, recognizing characteristics of the support distribution will benefit the proper mining system design. However, although we have demonstrated in Section 2 that the power-law relationship appears in the support distribution, it is prohibitively expensive to find all itemsets and further estimate the characteristics of the Zipf distribution, i.e., the slope  $\theta$  and the  $Y$ -intercept  $\Omega$  in Eqs. 2 and 3. The extremely large time consumption results from the expensive process to retrieve all itemsets without the support pruning. An efficient approach is still demanded to correctly estimate those parameters.

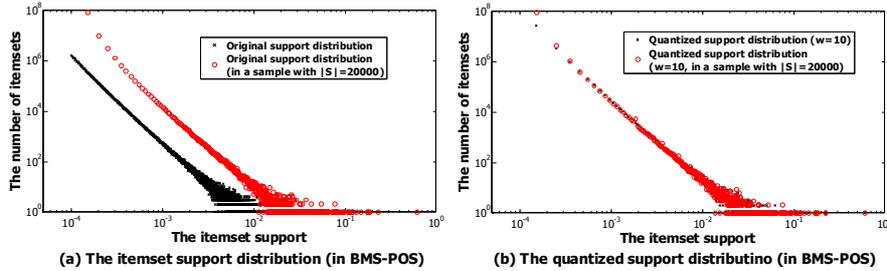


Fig. 2. The support distribution after sampling (the BMS-POS dataset)

As a consequence, we propose in this paper a valid and cost-effective solution, named *PPL* (standing for **P**redict the **P**ower-**L**aw relationship), to estimate the parameters of the power-law relationship in the *itemset support distribution*. Since the time consumption is dominated by the process of retrieving all itemsets in the large database, algorithm *PPL* utilizes two approaches to improve the efficiency. The first one is to utilize *sampling* techniques to retrieve itemsets [13]. The other approach is to retrieve *only the set of high-support itemsets* with the help of the support pruning techniques [10] so as to efficiently discover the parameters of the power-law relationship from the partial set of itemsets. Specifically, to fully utilize the capability of these two approaches, algorithm *PPL* is devised as a three-phase approach: (1) sampling; (2) obtaining high-support itemsets; (3) estimating the parameters of the power-law relationship by the linear regression from the high-support itemsets discovered in the sample.

However, while pursuing the efficiency, algorithm *PPL* will face three challenges:

(1) *The support distribution obtained in a sample will deviate from the support distribution in the original database.* Note that after sampling, the supports

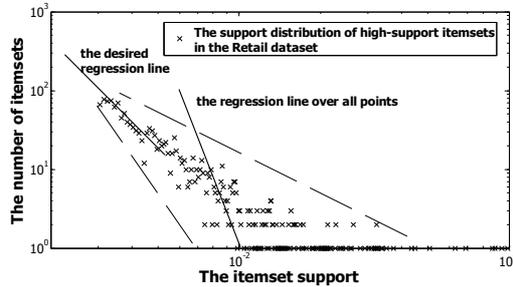


Fig. 3. Estimating the power-law relationship from high-support itemsets

of many low-support itemsets will become higher, and vice versa [15]. Unfortunately, as pointed out in [15], the number of itemsets with a specified support is likely to *increase* after sampling due to the large amount of transfers from low-support itemsets (the number of low-support itemsets is larger than the number of high-support itemsets). As shown in our empirical studies in Figure 2(a), where the support distributions obtained in the original dataset and in a random sample with 20,000 tuples are included, it can be apparently observed that the support distribution in the sample deviates from that in the original dataset. Indeed, due to randomness, we cannot estimate the deviation between the support distribution in a sample and that in the original dataset. Thus after sampling, it will be difficult to correctly predict the characteristics of the support distribution in the original dataset.

(2) *It is very difficult to determine the appropriate minimum support without prior knowledge.* Note that PPL will only discover high-support itemsets. However, we will not know how to determine the subtle minimum support. A large minimum support will result in too few itemsets to provide the sufficient information to correctly estimate the parameters of the Zipf distribution. Oppositely, the small minimum support will generate a lot of itemsets, thus resulting in inefficiency.

(3) *It is difficult to obtain the desired regression line due to the support fluctuation on high-support itemsets.* Note that the Zipf distribution can be characterized by the regression line. However, consider the observation in Figure 3, where a solid straight line represents the regression line over all points with respect to high supports, and the dotted lines show the envelope of the support fluctuation. As can be seen, points with respect to high supports do not exactly follow the Zipf distribution, and the support distribution of these points has the large support fluctuation. It will incur the large least-square errors, and the regression line over points with respect to high supports may deviate from the desired regression line<sup>8</sup>.

<sup>8</sup> In [3], the slope of the log-log plot is obtained by using the linear regression, excluding the rightmost 100 points to avoid the serious effect of the fluctuation. However, such an approach will fail in our cases since we may only have the rightmost 100 points which are summarized from high support itemsets.

To overcome those challenges, several novel mechanisms will be devised in algorithm *PPL*. In the following, we perform step-by-step analysis to discuss the details.

### 3.1 Phase I: Sampling

The goal of Phase I is to select a sample from the original dataset. Note that as mentioned in the first challenge described above, the support distribution in a sample will deviate from that in the original dataset, and the deviation is unpredictable. In fact, this phenomenon can be significantly reduced in the *quantized support distribution*, which will be obtained by the *histogram* technique [9]. Explicitly, all itemsets can be aggregated by means of the traditional equi-width *histogram* and then obtain the *quantized support distribution*. We give the formal definition of the *quantized support distribution* below.

**Definition 1 (The Quantized Support Distribution).** *Given all points  $(s_i, f_i)$  in the original support distribution, where  $s_i$  denotes the support of itemsets and  $f_i$  denotes the count of itemsets whose supports are  $s_i$ . After aggregating those points by means of the equi-width histogram, a set of new points  $(\hat{s}_j, \hat{f}_j)$  will be obtained, where  $\hat{s}_j$  denotes the representative value (the default is the median value) of the support range corresponding to the  $j^{\text{th}}$  bucket of the histogram, and  $\hat{f}_j$  denotes the count of itemsets with supports falling in the  $j^{\text{th}}$  bucket. The quantized support distribution is the distribution consisting of all points  $(\hat{s}_j, \hat{f}_j)$ .*

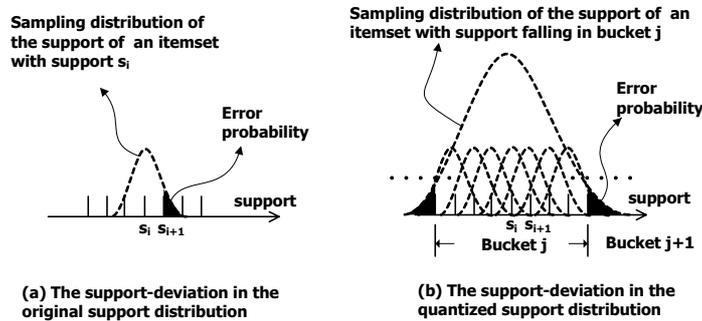


Fig. 4. Influence of the support-deviation

The argument that the *quantized support distribution* is able to reduce the influence of support-deviation follows the observation below:

**Observation.** Suppose that we repeatedly generate a lot of samples of the same sample size. The distribution of the *support* of  $X$  among these samples, i.e., the *sampling distribution* of the support of  $X$ , will approximately follow a *normal* distribution with *mean* equal to the support of  $X$  in the entire dataset [5]. In addition, the variance of the *sampling distribution* depends on the sample size [13]. As shown in Figure 4(a), the *sampling distribution* of an itemset with support equal to  $s_i$  in the entire database indicates that the support will be likely

larger than  $s_i$  with the probability equal to the shadow region. As a result, a percentage of itemsets in the sample will have supports inconsistent with the corresponding support in the entire dataset. Accordingly, the *itemset support distribution* after sampling will deviate from the *itemset support distribution* in the entire dataset. This argument is demonstrated in Figure 2(a).

On the other hand, consider the case of the *quantized support distribution*. As shown in Figure 4(b), the error probability, i.e., the probability of itemsets with supports in bucket  $j$  changing to bucket  $j + 1$  after sampling, will be relatively small as compared to the error probability illustrated in Figure 4(a). The reason lies in that the supports of most itemsets are likely to remain in the same support bucket after sampling. In other words, only itemsets with supports in the margin of a bucket are likely to have the support not falling in the same bucket after sampling. This argument is demonstrated in Figure 2(b), where the *quantized support distributions* obtained in the original dataset and in the sample with 20,000 tuples are shown and the parameter  $w$  denotes the number of aggregated points. It is clear to see that the *quantized support distribution* in a sample will be close to the *quantized support distribution* in the original dataset. ■

Following the observation, we comment that the *quantized support distribution* will be insensitive to the support-deviation, meaning that *the quantized support distribution in the sample will be close to the quantized support distribution in the entire dataset*. As a result, we will aim to obtain the *quantized support distribution* in the sample.

Another problem, as shown in Figure 1, is that the *quantized support distribution* still deviates from the original support distribution. Importantly, assuming that the original support distribution approximately follows the Zipf distribution, Proposition 1 below indicates that the *quantized support distribution* also has the same slope as the slope in the original support distribution and has a "predictable" drift of the Y-intercept.

**Proposition 1.** *Suppose that the itemset support distribution follows the Zipf distribution so that we have  $\log(f_i) \approx \theta \log(s_i) + \Omega$ . Assuming that there are  $w$  distinct points in the original support distribution being aggregated as a point in the quantized support distribution, we will have an approximate Zipf distribution as the form*

$$\log(\widehat{f}_k) \approx \theta \log(\widehat{s}_k) + \Omega + \log(w),$$

*in the quantized support distribution, where  $\widehat{s}_k$  denotes the representative of the quantized support in the  $k^{\text{th}}$  bucket and  $\widehat{f}_k$  denotes the count of itemsets whose supports fall in the  $k^{\text{th}}$  bucket. As such, the log-log plot in the quantized support distribution has the slope  $\theta$  and the Y-intercept  $\Omega + \log(w)$ .*

**Proof.** Note that we have  $e^\Omega \times s_{i,j}^\theta \approx f_{i,j}$  for the point  $(s_{i,j}, f_{i,j})$  in the original support distribution since it follows the Zipf distribution. Suppose that  $|D|$  is the database size. Let points  $(s_{k,1}, f_{k,1}), (s_{k,2}, f_{k,2}), \dots, (s_{k,w}, f_{k,w})$  be summarized as the  $k^{\text{th}}$  point  $(\widehat{s}_k, \widehat{f}_k)$  in the *quantized support distribution*. We have  $\widehat{f}_k = \sum_{j=1}^w f_{k,j}$ , and

$$\widehat{s}_k = \frac{s_{k,1} + s_{k,w}}{2} = \frac{s_{k,1} + \left(s_{k,1} + \frac{w}{|D|}\right)}{2} = s_{k,1} + \frac{w}{2 \times |D|}.$$

Since  $\frac{w}{|D|}$  is in general much weak as compared to  $s_{k,1}$ , we have

$$\frac{s_{k,j}^\theta}{\widehat{s}_k^\theta} = \left(\frac{s_{k,1} + \frac{j}{|D|}}{s_{k,1} + \frac{w}{2 \times |D|}}\right)^\theta \approx 1.$$

Therefore  $s_{k,j}^\theta$ , for  $1 \leq j \leq w$ , will be approximately equal to  $\widehat{s}_k^\theta$ , which yields that

$$\widehat{f}_k = \sum_{j=1}^w f_{k,j} \approx e^\Omega \times \sum_{j=1}^w \widehat{s}_k^\theta = e^\Omega \times w \times \widehat{s}_k^\theta,$$

$$\log(\widehat{f}_k) \approx \theta \log(\widehat{s}_k) + \Omega + \log(w). \quad \blacksquare$$

Proposition 1 indicates that the slope  $\theta$  remains in the *quantized support distribution*, and the Y-intercept will be changed to  $\Omega + \log(w)$ . Figure 1 demonstrates Proposition 1, where we can see that, for high-support points, the slope of the *quantized support distribution* ( $w = 10$  or  $100$ ) is equal to that of the *quantized support distribution* without sampling. As a result, the side-effect of sampling is overcome.

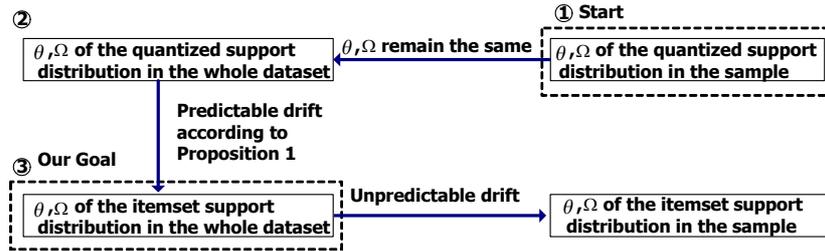


Fig. 5. The flow to overcome problems incurred by sampling

Based on the foregoing, the process to overcome problems incurred by sampling, as shown in Figure 5, will be summarized as:

- (1) Obtain the characteristics of the *quantized support distribution* in the sample.
- (2) The characteristics of the *quantized support distribution* in the whole dataset are expected equal to that in the sample.
- (3) In light of Proposition 1, obtain the characteristics of the original *itemset support distribution*.

Note that while step 1 is completed, steps 2 and 3 can be straightforwardly executed with the mathematical manipulation mentioned above. How to precisely achieve step 1 will be discussed in Section 3.2 and Section 3.3.

The remaining issue in this phase is, what is the appropriate sample size to obtain the *quantized support distribution* in the sample which is consistent with that in the entire database. Formally, the level of consistency depends on the

variance of the *sampling distribution* of the support, and the variance relies on the sample size [13]. A small sample size will lead to a large variance as compared to the variance in a large sample size. As pointed out in previous works of sampling for mining association rules, a sample size equal to 20,000 [13] or a sample rate equal to 10% [15], will be sufficient to generate the accurate set of frequent itemsets. We argue that the sample size 20,000 or 10% is also sufficient to generate the accurate *quantized support distribution* by following several points: (1) the complexity to generate the accurate *quantized support distribution* is analogous to the complexity to generate accurate frequent itemsets; (2) in Phase II only high-support itemsets will be generated, whose supports, as indicated in [13], can be easily preserved in samples as compared to supports of low-support itemsets; (3) the discrepancy between counts of itemsets within a bucket in the sample and in the entire dataset will be unapparent in the log-log scale (the characteristics of the power-law relationship is estimated in the log-log scale); (4) the technique in Phase III is specifically designed to be robust to the inconsistency between *quantized support distributions* in the sample and in the entire dataset.

As simultaneously considering execution efficiency and above points, we therefore set the sample size as 20,000 in default since the sample can be easily executed and maintained in main memory. The discreet users can set the size as 10% of the entire size, as the suggestion in [15]. We will also investigate the issue of the sample size in our empirical studies later.

### 3.2 Phase II: Discover High-Support Itemsets in the Sample

In this phase, the high-support itemsets in the sample will be discovered. Without prior knowledge to determine the appropriate minimum support, we resort to the technique of "discover top- $k$  itemsets" [4][14] instead of "discover itemsets with the specified minimum support," where top- $k$  itemsets refer to the  $k$  most frequent itemsets in the dataset. In practice, the size of  $k$  can be easily specified a priori. As will be shown in our experimental results,  $k$  equal to 5,000 will suffice to correctly estimate the parameters of the power-law relationship in most cases. As such we set  $k$  as 5,000 in default, where top 5,000 itemsets can be efficiently retrieved by the state-of-the-art algorithm for mining top- $k$  frequent itemsets.

### 3.3 Phase III: Characterize the Power-Law Relationship

The parameters of the Zipf distribution will be estimated in this phase. Suppose that  $\{X_1, \dots, X_k\}$  is the set of top- $k$  itemsets which are obtained in Phase II. At the beginning of this phase, we will aggregate these itemsets by means of *histogram* with the support bucket width equal to  $\frac{w}{|S|}$ , where  $|S|$  is the size of the sample dataset and  $w$  is the number of distinct and consecutive support counts which will be aggregated into the same bucket. Note that the default of  $w$  is 10 in this paper since empirically  $w = 10$  is able to preserve the slope of the *itemset support distribution*, as shown in Figure 1. As such, top- $k$  itemsets will be aggregated into a set of points  $H_k = \{(\hat{s}_1, \hat{f}_1), (\hat{s}_2, \hat{f}_2), \dots, (\hat{s}_z, \hat{f}_z)\}$

sorted by  $\widehat{s}_i$ , where  $\widehat{s}_i < \widehat{s}_j$  iff  $i < j$ . We therefore can characterize the power-law relationship by performing the regression analysis over the partial *quantized support distribution* which is summarized from *top-k* itemsets discovered in the sample.

However, as pointed out as the third challenge described in the beginning of Section 3, directly executing the regression analysis over all points in  $H_k$  will result in the incorrect estimation due to the support fluctuation on high support itemsets. Therefore the problem arises: "how to select an appropriate subset of points from  $H_k$  to correctly estimate the parameters of the Zipf distribution?" Recall the observation in Figure 3. Points with respect to very high-supports usually do not accurately follow the Zipf distribution. On the other hand, without loss of generality, points with respect to low supports usually follow the Zipf distribution. As such, one may intuitively claim a naive approach as follows.

**Naive Approach.** It is intuitive to suggest the regression line over first several points in  $H_k$  since they are sufficient to correctly fit the power-law relationship. For example, we may estimate the power-law relationship by performing the regression analysis over the first five points in  $H_k$ , i.e.,  $\{(\widehat{s}_1, \widehat{f}_1), \dots, (\widehat{s}_5, \widehat{f}_5)\}$ . Nevertheless, we indeed did not know how many points are sufficient to obtain the desired regression line. Thus we have to examine all possible regression lines, and then select the one with the best correlation coefficient since it will have the best power to explain the log-log relationship in the Zipf distribution. ■

However, such an approach suffers from the problem that the best correlation coefficient does not imply the best fit of the Zipf distribution. In particular, sometimes few points will result in the best correlation coefficient, but the regression line could be bias to outlier points [12]. In addition, sampling in Phase I may incur noise, which will also affect the result of the linear regression. As a result, we devise a novel solution, which is inspired from the training and testing scenario in supervised learning [8], to correctly estimate the parameters of the Zipf distribution from  $H_k$ .

**Minimizing Testing Error Approach.** Suppose that  $H_k$  is divided into two distinct and consecutive subsets of points, i.e., the training set  $T_r$  and the testing set  $T_e$ , where  $T_e = \{(\widehat{s}_1, \widehat{f}_1), \dots, (\widehat{s}_m, \widehat{f}_m)\}$  and  $T_r = \{(\widehat{s}_{m+1}, \widehat{f}_{m+1}), \dots, (\widehat{s}_z, \widehat{f}_z)\}$ . Here  $m$  is the parameter to adjust the size of the testing set and  $m < z$ . Consider the illustration in Figure 6, where each point in  $T_e$  is called a testing point. Our goal is to find the *best fit regression line* from  $T_r$  so that all testing points in  $T_e$  can well lie in the line. Formally, we give the definition of the *best fit regression line* in the following.

**Definition 2 (Best Fit Regression Line).** Given the training set  $T_r$  and the testing set  $T_e$ . The best fit regression line, denoted by  $\mathbb{R}_g(\widehat{s}_i) = \widehat{\theta}_g \log(\widehat{s}_i) + \widehat{\Omega}_g$ , will satisfy:

- (1)  $\mathbb{R}_g(\widehat{s}_i) = \widehat{\theta}_g \log(\widehat{s}_i) + \widehat{\Omega}_g$  is the regression line over the first  $g$  points in  $T_r$ , i.e.,  $\{(\widehat{s}_{m+1}, \widehat{f}_{m+1}), \dots, (\widehat{s}_g, \widehat{f}_g)\}$ , where  $m + 1 \leq g \leq z$ .

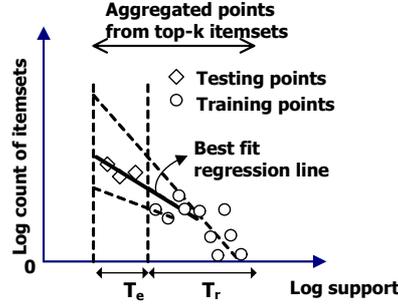


Fig. 6. The illustration of the best fit regression line

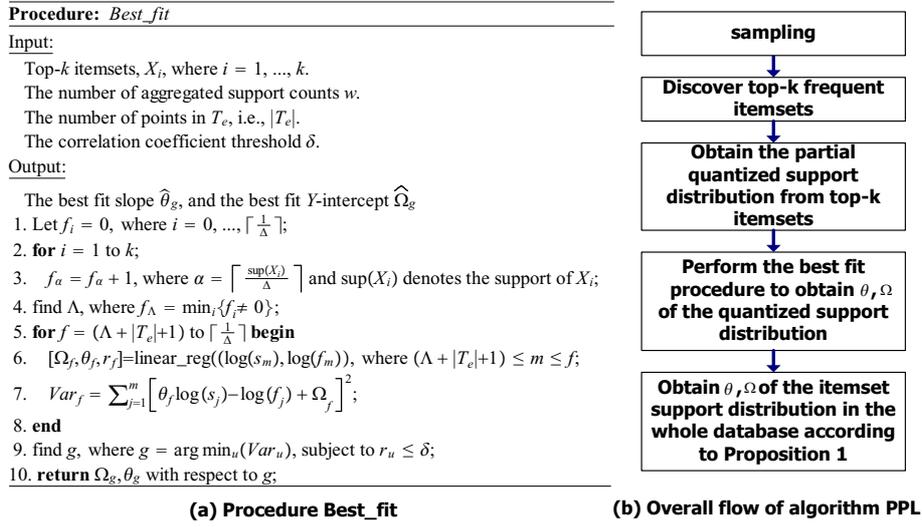


Fig. 7. The implementation of algorithm PPL

(2) The correlation coefficient,  $r_g$ , over the data points  $\left\{ \left( \hat{s}_{m+1}, \hat{f}_{m+1} \right), \dots, \left( \hat{s}_g, \hat{f}_g \right) \right\}$  is smaller than a pre-defined threshold  $\delta$ . Note that,

$$r_g = \frac{\sum_{i=m+1}^g \sum_{j=m+1}^g (\log(\hat{s}_i) - u_s) (\log(\hat{f}_j) - u_f)}{\sqrt{\sum_{i=m+1}^g (\log(\hat{s}_i) - u_s)^2} \sqrt{\sum_{j=m+1}^g (\log(\hat{f}_j) - u_f)^2}}, \quad (5)$$

where  $u_s$  and  $u_f$  are the mean of  $\log(\hat{s}_i)$  and  $\log(\hat{f}_j)$ , respectively.

(3)  $g = \arg \min_u \left\{ \sum_{j=1}^m \left( \mathbb{R}_u(\hat{s}_j) - \log(\hat{f}_j) \right)^2 \right\}$ , subject to the correlation coefficient  $r_f \leq \delta$  and  $m + 1 \leq u \leq z$ .

The whole procedure to find the best fit regression line is outlined in Procedure *Best\_fit* in Figure 7(a), where the function `linear_reg()` will return three parameters, the intercept  $\Omega$  (see Eq. 3), the slope  $\theta$  (see Eq. 2) and the *correlation coefficient*  $r$ . Specifically, the correlation coefficient  $r_g$  (a value between -1 and 1) can represent the level how those points are explained by the regression line. The regression line will fit points better when  $r_g \rightarrow -1$  since without loss of generality,  $f_i$  and  $\hat{s}_i$  are negatively correlated. Statistically, it is believed that  $r_g \leq -0.8$  is sufficient to claim the regression line can explain these points [12]. Thus  $\delta$  is set as  $-0.8$  in default. Note that criterion 3 in Definition 2 states that we desire the regression line with the minimum testing error. It is worth mentioning that, algorithm *PPL* will degenerate to the naive approach if there is no testing point in  $T_e$  and simply choose the regression line with the best correlation coefficient. For comparison purposes, we will also show the result of the naive approach in our experimental results. Note that the best fit regression line will be discovered in the *quantized support distribution* generated from top-k itemsets in the sample. In light of Proposition 1, the slope and the Y-intercept in the original *itemset support distribution* will be equal to  $\hat{\theta}_g$  and  $\hat{\Omega}_g - \log(w)$ , respectively.

We finally summarize the overall flow of algorithm *PPL*, as shown in Figure 7(b): (1) sampling; (2) discover *top-k* frequent itemsets in the sample; (3) aggregate the support of *top-k* itemsets by means of the equi-width histogram so as to obtain the partial *quantized support distribution*; (4) perform Procedure *Best\_fit* to obtain the characteristics of the *quantized support distribution* in the sample; (5) identify the characteristics of the power-law relationship in the *itemset support distribution* in the entire database according to Proposition 1.

## 4 Experimental Studies

The four real skewed datasets described in Table 1 are utilized in our experimental studies. Since the goal to show the support distribution follows the Zipf distribution has been demonstrated in Section 2, we in this section investigate whether algorithm *PPL* can efficiently and correctly estimate the parameters of the power-law relationship in the itemset support distribution. The simulation is coded by C++ and performed on Windows XP in a 1.7GHz IBM compatible PC with 512MB of memory. The default parameters in the experiments are: (1)  $k = 5,000$  (*top-k* itemsets); (2) the number of aggregated support counts  $w = 10$ ; (3) the number of points in the training set  $|T_e| = 5$ ; (4) the correlation coefficient threshold  $\delta = -0.8$ ; (5) the sample size  $|S| = 20,000$ .

We investigate whether algorithm *PPL* with the default parameters is able to correctly characterize the power-law relationship in four real datasets. The results are presented in Figure 8(a)~Figure 8(d), where the original support distributions and the *best fit regression lines* obtained by algorithm *PPL* (with their slopes  $\theta$  and Y-intercepts  $\Omega$ ) are shown. Note that the *best fit regression line* is discovered in the quantized support distribution in the sample. As can be seen, the *best fit regression line* can perfectly characterize the Zipf distribution in the four real datasets, showing the effectiveness of *PPL*.

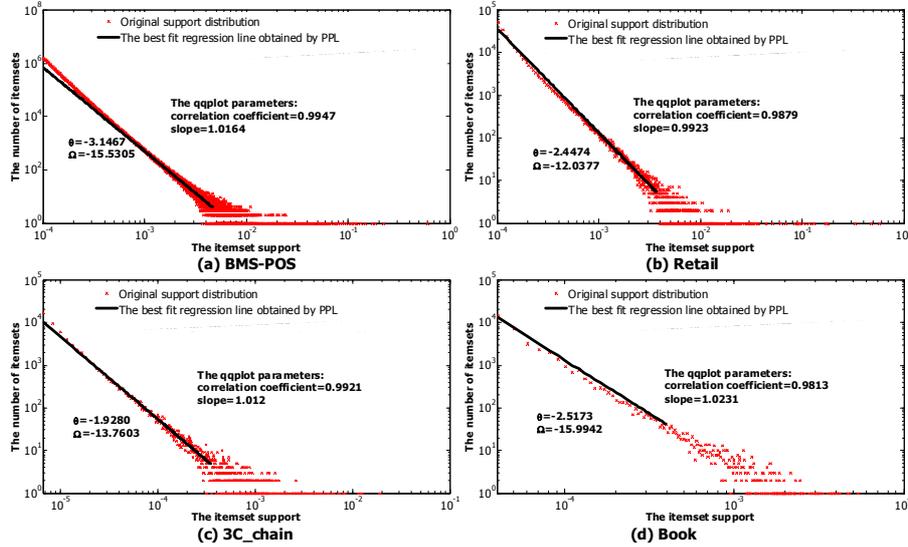


Fig. 8. The results of algorithm *PPL*

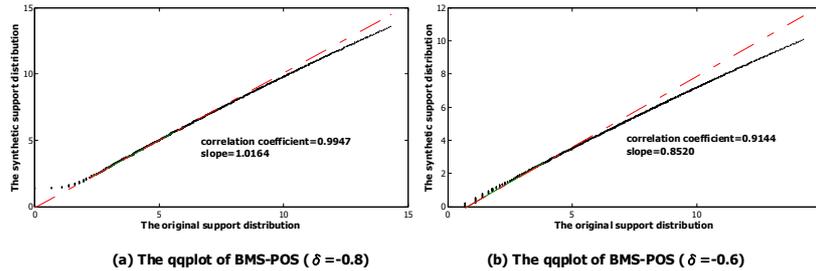


Fig. 9. The qqplot results in BMS-POS with various  $\delta$

Furthermore, the execution time is shown in Figure 10, where the execution time of "Brute force approach" indicates the time to retrieve the original support distribution in Figure 1 by algorithm FP-growth. Indeed, the brute force approach can correctly determine the parameters of the Zipf distribution by finding most of itemsets, but it will pay for the extremely large time consumption. On the other hand, *PPL* can efficiently estimate the parameters of the power-law relationship by avoiding the expensive process to obtain all itemsets. It is worth mentioning that the efficiency gain in Figure 10, which is calculated as the execution time of the brute force approach divided by the execution time of algorithm *PPL*, shows that algorithm *PPL* is in orders of magnitude faster than the brute force approach.

Same as the experiments in [2], the quantitative analysis of algorithm *PPL* will be evaluated by the traditional method of quantile-quantile plot (qqplot), as

Dataset	Brute force approach	Algorithm PPL	Efficiency Gain
BMS-POS	632 sec	8 sec	79
Retail	1248 sec	5 sec	249.6
3C_chain	2547 sec	10 sec	254.7
Book	492 sec	3 sec	164

Fig. 10. The execution time of different approaches

Variant Parameters	BMS-POS		Retail		3C_chain		Book	
	Corr. Coef.	Slope						
Default	0.99	1.02	0.99	0.99	0.99	1.01	0.98	1.02
$T_e=0$ (naive)	0.89	0.73	0.83	0.82	0.91	0.83	0.87	0.93
$T_e=10$	0.99	0.98	0.98	0.96	0.98	1.01	0.99	1.02
$\delta=-0.5$	0.84	0.86	0.91	1.08	0.92	0.94	0.87	1.11
$\delta=-0.9$	0.99	1.01	0.99	1.02	0.97	1.06	0.98	1.03
$T_e=0; \delta=-0.5$	0.81	1.21	0.77	1.11	0.73	1.18	0.84	1.13
$k=10,000$	0.98	1.02	0.99	0.97	0.97	1.08	0.98	1.07
$k=50,000$	0.99	0.99	0.99	1.02	0.98	0.97	0.98	0.94
$ S =10,000$	0.93	1.09	0.98	0.97	0.95	1.09	0.98	0.97
$ S =50,000$	0.99	0.99	0.99	1.03	0.98	0.99	0.99	1.03
$ S =0.1 D $	0.99	1.03	0.93	1.13	0.99	0.98	0.91	0.94
$ S =0.2 D $	0.99	1.01	0.94	1.04	0.98	0.96	0.94	1.03
$w=50$	0.98	1.03	0.91	0.94	0.94	1.02	0.96	1.08
$w=100$	0.93	1.14	0.88	1.13	0.98	1.01	0.92	0.94

Fig. 11. The qqplot results of four real datasets

the one shown in Figure 9. The qqplot is used to compare the quantiles of two datasets. If the distributions of these two datasets are similar, the qqplot will be linear and the slope will be close to one. As such, we generate a synthetic support distribution according to the parameters estimated by algorithm *PPL*, and then make a qqplot between the original support distribution and the synthetic support distribution. Afterward, two important factors can be calculated: (1) the slope of the qqplot; (2) the correlation coefficient of points in the qqplot. If both are close to one, we can claim that the real distribution and the synthetic distribution are from the same distribution [2], meaning that the regression line can perfectly represent the data distribution.

The qqplots on various correlation coefficient thresholds  $\delta$  are shown in Figure 9, where Figure 9(a) is the qqplot corresponding to the result of Figure 8(a). We can find that the qqplot in Figure 9(a) is close to linear, except points with respect to very low supports and very high supports. Note that points with respect to high supports have been observed not exactly following the power-law relationship and points with respect to low supports in the BMS-POS dataset upwardly vary from the Zipf distribution, thus causing the deviation of a few points. However, the slope and the correlation coefficient are very close to unity, indicating that the synthetic distribution can mostly correctly fit the real distribution. Furthermore, when we increase the threshold  $\delta$ , as shown in Figure 9(b), the estimated quality degrades, showing the importance of the criterion 2 of the *best fit regression line*. Indeed, a regression line with the low correlation coeffi-

cient loses its effectiveness to estimate the power-law relationship, even though it satisfies criterion 3, i.e., having the minimum testing error.

Due to space limitations, other qqplot results of four real datasets are summarized in Figure 11. At first, we observe results with various  $|T_e|$ . Note that the case  $|T_e| = 0$  can be deemed as the naive approach discussed in Section 3.3. As can be seen, the naive approach cannot correctly model the distribution since the correlation coefficient and the slope deviate a lot from unity. On the other hand,  $|T_e| = 5$  (default cases) and  $|T_e| = 10$  both lead to the desirable result. Moreover, the studies of various  $\delta$  are also shown, and we can find that  $\delta = -0.8$  (default cases) or  $-0.9$  will result in the correlation coefficient and the slope close to one. Note that without loss of generality, the results of  $|T_e| = 0$  and  $\delta = -0.5$  can be deemed as the case to obtain the regression line over all points from *top-k* itemsets. It can be seen that the regression line over all points loses of its power to explain the real data distribution. The above observations all demonstrate the effectiveness of algorithm *PPL*.

In addition, with the result of various  $k$ , we can conclude that the default  $k = 5,000$  is sufficient to obtain high quality results. Note that top-5000 itemsets can be efficiently retrieved in the sample, indicating the efficiency and effectiveness of algorithm *PPL*. We also investigate the influence of the sample size. Clearly, the result obtained in the sample with the default size 20,000 is close to the result obtained in the large sample with size equal to  $0.2 \times |D|$ , showing that the resulting quality is insensitive to the sample size if the sample size is not arbitrarily small. Finally, we observe the result of various  $w$ . Note that Proposition 1 will not hold when  $w$  is large. Thus it can be seen that  $w = 100$  slightly degrades the estimated quality of algorithm *PPL*. Since the goal of *histogram* in this paper is to diminish the side-effect of sampling, we conclude that  $w = 10$  is sufficient to achieve this, and will give the excellent fit of the itemset support distribution.

## 5 Conclusions

In this paper, we demonstrated that the power-law relationship appears in the distribution of itemset supports in the real datasets. Discovering such a relationship is useful for many applications. To avoid the costly process of retrieving all itemsets, we proposed algorithm *PPL* to efficiently extract characteristics of the power-law relationship. As shown in the experimental results, algorithm *PPL* is able to efficiently extract the characteristics of the power-law relationship with high accuracy.

## References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of VLDB*, 1994.
2. Z. Bi, C. Faloutsos, and F. Korn. The "DGX" Distribution for Mining Massive, Skewed Data. In *Proc. of SIGKDD*, 2000.
3. L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. In *Proc. of IEEE INFOCOM*, 1999.

4. Y.L. Cheung and A.W. Fu. Mining Association Rules without Support Threshold: with and without Item Constraints. In *TKDE*, 2004.
5. W. G. Cochran. *Sampling Techniques*. John Wiley and Sons, 1977.
6. L. Egghe. The distribution of n-grams. *Scientometrics*, 2000.
7. F. Geerts, B. Goethals, and J. V. d. Bussche. A tight upper bound on the number of candidate patterns. In *Proc. of IEEE ICDM*, 2001.
8. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.
9. Y. Ioannidis. The history of histograms. In *Proc. of VLDB*, 2003.
10. J.-S. Park, M.-S. Chen, and P. S. Yu. An effective hash based algorithm for mining association rules. In *Proc. of SIGMOD*, 1995.
11. G. Ramesh, W. A. Maniatty, and M. J. Zaki. Feasible itemset distributions in data mining: Theory and application. In *Proc. of ACM PODS*, 2003.
12. J. A. Rice. *Mathematical statistics and data analysis*. Duxbury Press, 1995.
13. H. Toivonen. Sampling large databases for association rules. In *Proc. of VLDB*, 1996.
14. J. Wang, J. Han, Y. Lu, and P. Tzvetkov. TFP: An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets. In *TKDE*, 2005.
15. M.J. Zaki, S. Parthasarathy, Wei Li, and M. Ogihara. Evaluation of sampling for data mining of association rules. In *Int. Workshop on Research Issues in Data Engineering*, 1997.
16. Z. Zheng, R. Kohavi, and L. Mason. Real world performance of association rule algorithms. In *Proc. of SIGKDD*, 2001.
17. G.K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley Press, 1949.