Logical Itemset Mining

Shailesh Kumar
Google Inc.
Hyderabad, India
Email: shkumar@google.com

Chandrashekar V and C V Jawahar
International Institute of Information Technology
Hyderabad, India
Email: {chandrasekhar.v@students, jawahar}@iiit.ac.in

***Abstract*—Frequent Itemset Mining (FISM) attempts to find large and frequent itemsets in bag-of-items data such as retail market baskets. Such data has two properties that are not naturally addressed by FISM: (i) a market basket might contain items from more than one *customer intent* (mixture property) and (ii) only a subset of items related to a customer intent are present in most market baskets (projection property). We propose a simple and robust framework called LOGICAL ITEMSET MINING (LISM) that treats each market basket as a *mixture-of, projections-of, latent customer intents*. LISM attempts to discover *logical itemsets* from such bag-of-items data. Each logical itemset can be interpreted as a latent customer intent in retail or semantic concept in text tagsets. While the mixture and projection properties are easy to appreciate in retail domain, they are present in almost all types of bag-of-items data. Through experiments on two large datasets, we demonstrate the quality, novelty, and actionability of logical itemsets discovered by the simple, scalable, and aggressively noise-robust LISM framework. We conclude that while FISM discovers a large number of noisy, *observed*, and frequent itemsets, LISM discovers a small number of high quality, *latent* logical itemsets.**

***Keywords*-Frequent Itemset Mining, Market basket analysis, Indirect and Rare Itemsets, Semantically Associated Itemsets, Apriori Algorithm.**

I. INTRODUCTION

Bag-of-items data, such as market baskets in retail or tagsets in text, is growing at a tremendous rate in many domains. A retail market basket comprises of products purchased by a customer in a store visit. A tagset comprises of a set of keywords describing an object (e.g. YouTube video, Flickr image, or a movie, etc.). Bag-of-items data mining attempts to discover novel patterns, create actionable insights, engineer predictive features, and drive intelligent decisions from such data.

More than a decade ago, Frequent Itemset Mining (FISM) [1] powered by the *Apriori* algorithm [2] became the standard for finding *large* and *frequent* itemsets in bag-of-items data. As the vocabulary and data size grew, scaling the original Apriori algorithm became the primary focus of research. This lead to a number of innovations in scalable data structures and algorithms, some of which are highlighted in Section II-A. Several other paradigms such as *rare itemset mining* [15], [14], *indirect association mining* [10], etc, emerged to address limitations of, and expand applications of the original FISM framework.

A common observation in traditional (direct) FISM is that it generates a very large number of noisy itemsets of which very few are really useful, novel, or actionable. In case of *indirect* association mining, where (potentially noisy) direct links are used to induce indirect associations, there is always a danger that the noise gets exaggerated and spurious indirect associations get created.

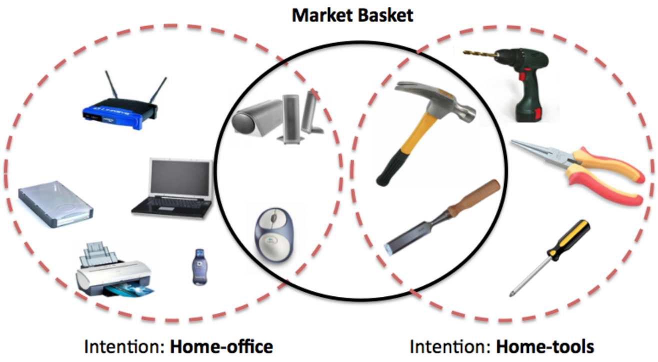


Figure 1. A hypothetical market basket (solid black circle) composed of items from two logical itemsets (red dotted circles), representing *latent* customer intentions.

In this paper, we first explore the underlying nature of the bag-of-items data to explain the inability of FISM to reduce noise and generate more useful itemsets. We then develop an alternate itemset mining framework that addresses the subtle nuances of such data more *naturally* than FISM does. We start with the following definitions, observations and assumptions[1]:

- Define **Logical Itemset** as a set of items that *completes* a *customer intent* in retail domain or *semantic concept* in a text or vision domain.
- These logical itemsets are **latent** in the data and the goal of LISM is to **discover** them in a completely unsupervised fashion.
- The observed bag-of-items data may be best described as a **mixture-of, projections-of, latent logical item-**

[1] This looks similar to, but is not exactly the same as, the topic-model argument that forms the basis of Latent Dirichlet Allocation [24] in text mining.

sets, i.e. it has two fundamental properties: the mixture property and the projection property.

- **Mixture property**: In retail, each market basket might contain *more than one customer intent*. Similarly, in text domain, each tagset might contain *more than one semantic concept*.
- **Projection property**: In retail, each market basket contains *only a subset of products* associated with a customer intent. Similarly, in text domain, each tagset contains *only a subset of keywords* associated with a semantic concept. In other words, a complete logical itemset is rarely present in its entirety in the bag-of-items data.

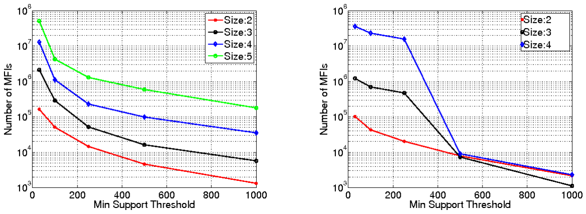
Figure 1 shows a **hypothetical example** of the *mixture-of, projections-of, latent customer intents* in retail. Consider a market basket with four products (shown in solid black circle). These products come from two different customer intents, each represented by a *logical* group of products (shown in red dotted circles). In other words, (a) the market basket is composed of products from more than one customer intent (mixture property) and, (b) the market basket does not contain *all* products in either of the two intents (projection property). It only contains a subset of products associated with each intent. This could happen for various reasons: the customer already has the other items in those intents, she might purchase the remaining items in the intents elsewhere or at some other time, or she might not even be aware that the other items *complete* her intents, etc.

The *noise* due to the mixture property and the *incompleteness* due to the projection property make it challenging to discover the latent logical itemsets from bag-of-items data. A complete logical itemset will have a very low support as it hardly occurs in the data - thanks to the projection property. Also, each frequent itemset discovered by the traditional FISM framework might have sufficient noise in it - thanks to the mixture property. Finally, also note that some logical items might occur more rarely in the data than others. In fact, in some cases, it might be more useful to find rare itemsets [15] rather than frequent itemsets.

It should be fairly obvious from this discussion as to why frequent itemset framework is not a *natural* framework for discovering logical itemsets and why we need a radically different framework for finding logical itemsets in such data. It should also be obvious that unless we effectively deal with the mixture-of-intents noise, in the bag-of-items data, any indirect association mining will suffer from the propagation of this noise to higher order associations.

One approach to find logical itemsets could be the traditional topic models such as LDA [24]. But they may not be directly applicable here for several reasons: (i) LDA is more suitable when a typical bag-of-items is much larger as in bag-of-words in text or bag-of-visual-words in images, (ii) LDA depends on the *weight* (e.g. term frequency of a term in the document) of each item in the bag but bag-of-items inherently do not have such weights - an item is either present or not present in the bag. (iii) The scaling and convergence properties of LDA will make it prohibitively costly to apply on such *thin* data, where the number of bags is typically much larger than the size of each bag, and (iv) finally, LDA requires us to specify apriori the number of concepts (or latent customer intents) to discover - something that is already hard in the text domain and will be even harder in the retail domain.

In this paper, we propose a simple and intuitive framework called the LOGICAL ITEMSET MINING (LISM) to find all the logical itemsets in bag-of-items data in an unsupervised and scalable fashion. This addresses the mixture and projection properties highlighted above in a novel fashion and is able to discover a relatively small number of very precise and high quality logical itemsets even if they have a low (or even zero) support in the data. In contrast, FISM typically discovers a large number of noisy itemsets. We first describe some of the prior work in FISM and indirect associations mining in Section II. Then we describe the LISM framework in detail in Section III. Finally we demonstrate through both subjective and objective empirical evidence the quality of results obtained by the LISM framework on two large public datasets, IMDB and FLICKR, described in Section IV-A.



(a) IMDB

(b) Flickr

Figure 2. Distribution of Maximal Frequent Itemsets w.r.t. Size and Threshold for IMDB and Flickr dataset. As the itemset size increases or the support threshold decreases, the number of frequent itemsets generated grows exponentially.

II. BACKGROUND

A. Frequent Itemset Mining

The FISM framework was originally developed for market basket analysis in retail domain where frequent itemsets can be used to improve store and catalogue layouts, increase cross sell and up sell, and do product bundling. FISM received a lot of attention since the introduction of association rule mining by Agrawal [1] in 1993. It really took off with the introduction of the elegant *Apriori algorithm* [2] that addressed the core problem of combinatorial explosion in itemsets by using the *anti-monotonicity property* of itemsets. According to this property: *If an itemset is not frequent, any of its supersets cannot be frequent*. Figure 2 demonstrates the effect of support threshold and itemset size on the number of itemsets discovered by FISM. FISM typically generates a very large number of maximal frequent itemsets, most of which tend to be noisy or meaningless.

Over the recent years, with the increase in itemset data, many efficient algorithms based on hash-based techniques, partitioning, sampling and using vertical data formats have emerged. Some of the notable ones are: FP-Growth [3], Eclat algorithm [4], Apriori by Borgelt [6], kDCI algorithm [7], DCI algorithm [7], and lcm [8]. Primary focus of most of these algorithms is to make FISM framework more scalable, practical, and efficient.

B. *Indirect and Rare Association Rule Mining*

FISM can only discover *direct* relationships observed in the data. However, deeper insights can come from indirect associations [9], [10], [11], [12], [13]. Recently, Liu *et al.* [18] suggested a hypergraph-based method for discovering semantically associated itemsets. In principle, our logical itemset approach is similar to their work at a high level but is substantially different in details. For example, we also restrict our model to pair-wise relationships only, but we use a different class of much simpler and noise-robust measures of associations. Our framework is much simpler and intuitive compared to [18]. In spite of the strong theory, the results presented in [18] have reasonable amount of noise that we believe is due to the mixture-of-intentions noise in the data. Our LISM framework, on the other hand, generates very high quality results both for high and low frequency itemsets.

While most of the focus in traditional data mining is on frequent itemsets, there is a large body of work on rare itemset mining as well. Finding rare itemsets are especially useful in biology and medical domains, where *rare events* are more important than common ones or in applications such as outlier detection, belief contradiction, and exception finding, etc. Szathmary [15], [14] present the first algorithm designed specifically for *rare itemset mining*. Haglin [16] designed an algorithm for finding minimal infrequent itemsets based on $SUDA2$ algorithm for finding minimal unique itemsets [17].

Logical itemset mining is *frequency agnostic*. It discovers rare itemsets as well as common itemsets as long as they are logical. In Section IV-B, we show examples of some of the logical itemsets that have been discovered by LISM on both the FLICKR and IMDB datasets.

III. LOGICAL ITEMSET MINING

As mentioned in Section I, discovering logical itemsets in bag-of-items data has two core problems. First, the noise due to the mixture-of-intentions property and second, the incompleteness due to the projection property. The LOGICAL ITEMSET MINING framework, described in detail in this section, addresses both these problems and attempts to discover many logical itemsets in the data[2]. LISM framework has four stages:

1) **Counting stage** where co-occurrence counts between all pairs of items is computed in one pass through the data.[3]
2) **Consistency stage** where these co-occurrence counts are converted to *consistency* values, quantifying the *statistical significance* or *information content* of seeing each pair of items together vs. random chance.
3) **Denoising stage** where the co-occurrence consistencies are cleaned further to address the mixture-of-intents property.
4) **Discovery stage** where logical itemsets in the form of *cliques* are discovered in the co-occurrence consistency graph by addressing the projection property.

Before we describe these four stages in detail, some notation: Let $\mathbf{V} = \{v_m\}_{m=1}^{M}$ denote the **Vocabulary** of all unique items in the data (e.g. all products sold by a retailer, all keywords in the tagset corpus, etc.). Let $\mathbf{X}$ denote the bag-of-items data with N data points:

$$\mathbf{X} = \left\{ \mathbf{x}^{(n)} = \left\{ x_\ell^{(n)} \right\}_{\ell=1}^{L_n} \subset \mathbf{V} \right\}_{n=1}^{N}, \qquad (1)$$

were L_n is the size of the n^{th} bag, $\mathbf{x}^{(n)}$.

A. *Stage-1: LISM-Counting*

Three types of statistics are counted in a single data pass:

1) **Co-occurrence counts**, $\psi(\alpha, \beta) = \psi(\beta, \alpha)$, for every pair of items $(\alpha, \beta) \in \mathbf{V} \times \mathbf{V}$ is defined as the *number of bags in which both items "co-occur"*:

$$\psi(\alpha, \beta) = \sum_{n=1}^{N} \delta\left(\alpha \in \mathbf{x}^{(n)}\right) \delta\left(\beta \in \mathbf{x}^{(n)}\right), \qquad (2)$$

(where $\delta(bool)$ is a Dirac delta function which is 1 if *bool* is true and 0 otherwise). Co-occurrence counts below a threshold θ_{cooc} are set to zero. The resulting $M \times M$ ($M = |\mathbf{V}|$ = vocabulary size) **Co-occurrence counts matrix**, $\Psi = [\psi(\alpha, \beta)]$ is sparse and symmetric. The time complexity of computing this matrix in one pass through the data is $\mathcal{O}\left(\sum_{n=1}^{N} \binom{L_n}{2}\right)$. The space complexity of storing this matrix is $\mathcal{O}(M^2 \lambda_{cooc})$ where λ_{cooc} is the sparsity factor of the co-occurrence counts matrix. Note that the threshold θ_{cooc} might be used to control the degree of noise in the counting.

2) **Marginal counts**, $\psi(\alpha)$ is defined as the *number of pairs in which the item $\alpha \in \mathbf{V}$ occurred with some other item in the data*. This is obtained by adding each row of the full co-occurrence counts matrix:

$$\psi(\alpha) = \sum_{\beta \in \mathbf{V}, \alpha \neq \beta} \psi(\alpha, \beta), \qquad (3)$$

[2] Discovering all logical itemsets is an NP-hard problem.

[3] This stage is akin to finding all frequent itemsets of size 2.

3) **Total Counts**, ψ_0, defined as the *total number of pairs in which some item co-occurred with some other item in the transaction data*. This is obtained by adding all the elements in the co-occurrence count matrix[4].

$$\psi_0 = \frac{1}{2}\sum_{\alpha \in \mathbf{V}} \psi(\alpha) = \frac{1}{2}\sum_{\alpha \in \mathbf{V}}\sum_{\beta \in \mathbf{V}} \psi(\alpha, \beta) \quad (4)$$

These three counts are then converted into co-occurrence and marginal probabilities[5]:

$$P(\alpha, \beta) = \frac{\psi(\alpha, \beta)}{\psi_0}, P(\alpha) = \frac{\psi(\alpha)}{\psi_0} \quad (5)$$

B. Stage 2: LISM-Consistency

FISM depends on the *support* i.e. *frequency* as a key statistic on itemsets. In fact, the pair-wise co-occurrence counting in Equation (2) in LISM is the same as finding all frequent itemsets of size 2 with a support threshold of θ_{cooc}. Consider the two examples where using co-occurrence counts does not make sense:

- **High Co-occurrence Noise** Consider a pair of common products such as `DVD` and `Shoes` sold by a retailer. Since both are high volume by themselves, they might co-occur in a large number of market baskets. This is an artifact of *mixture-of common intents*. We need a mechanism to ignore this high co-occurrence count. Unfortunately, if we raise the support thresholds too high, we might loose valid co-occurrences with lower counts.
- **Low Co-occurrence Signal** Consider a pair of rare products such as `home-theatre-system` and `high-definition-TV`. While the joint co-occurrence count for this pair of products might be low, the “confidence” (using the FISM terminology) - measured by the conditional probability of seeing one product given the other - might still be high. To keep such low frequency co-occurrences, the support threshold will have to be reduced substantially, which in turn will result in addition of lot of spurious product pairs.

Thus, we need a systematic mechanism to remove deceptive high co-occurrences that are an artifact of mixture-of-intents noise, while at the same time preserve the important low co-occurrence counts that contain important logical connections between pairs of rare products.

The first fundamental difference between FISM and LISM is that instead of using the joint probability as it is, LISM normalizes these co-occurrence counts by the priors of the two items. This not only addresses the noise due to mixture-of-intents (the deceptive high co-occurrence counts), but will also preserve the rare but logical co-occurrence between products (important low co-occurrence counts). In LISM, we call this *statistical significance measure* as the **Co-occurrence Consistency** defined as the *degree with which the actual co-occurrence of a pair of items compares with random chance*. In other words, if the actual joint probability, $P(\alpha, \beta)$, is more compared to the random chance, (e.g. $P(\alpha)P(\beta)$) then the two items are said to have co-occurred with high consistency. There are a number of measures that can be used here. We list a few candidates here. See [25] for a more exhaustive list of measures that can be used here.

- **Cosine**

$$\phi_{csn}(\alpha, \beta) = \frac{P(\alpha, \beta)}{\sqrt{P(\alpha)P(\beta)}} \in [0, 1] \quad (6)$$

- **Jaccard Coefficient**

$$\phi_{jcd}(\alpha, \beta) = \frac{P(\alpha, \beta)}{P(\alpha) + P(\beta) - P(\alpha, \beta)} \in [0, 1] \quad (7)$$

- **Point-wise Mutual Information**

$$\phi_{pmi}(\alpha, \beta) = max\left\{0, \log\left(\frac{P(\alpha, \beta)}{P(\alpha)P(\beta)}\right)\right\} \in [0, \infty] \quad (8)$$

- **Normalized Point-wise Mutual Information**

$$\phi_{nmi}(\alpha, \beta) = \frac{\phi_{pmi}(\alpha, \beta)}{-\log P(\alpha, \beta)} \in [0, 1] \quad (9)$$

We use normalized point-wise mutual information in this paper, as it is bounded and addresses a well known problem with point-wise mutual information, that PMI exaggerates rare items/pairs more. A threshold θ_{consy} is used to remove all product pairs whose consistency is below this threshold. The resulting *co-occurrence consistency matrix* is used to find logical itemsets, but before that there is scope to reduce even more noise in this matrix.

C. Stage-3: LISM-Denoise

Some mixture-of-intents noise was removed by converting co-occurrence counts to co-occurrence consistencies. This, however, does not completely eliminate the entire noise from the consistency matrix. We need to do further denoising using the following intuition. In the first pass through the data, all pairs of items in a market basket are counted as there is insufficient knowledge to know whether a pair of items is **noise** (e.g. `mouse, hammer`, in example shown in Figure 1) due to mixture-of-intents property or *signal* (e.g. `mouse, speakers`) i.e. they really belong to the same customer intent. After computing the co-occurrence consistencies after the first pass, however, some knowledge is created, as to whether a particular product pair in a bag is signal or noise. The assumption, we are making is that in each iteration, in spite of the mixture-of-intents noise, product pairs that are likely to belong to an eventual logical itemset will remain *connected*. In fact, we observe,

[4] Note that we divide this sum by 2 due to double counting in the symmetrical matrix.

[5] Laplacian smoothing might be used to compute these probabilities

Tag	Before Denoising	After Denoising
bride	0.3257	0.5750
reception	0.3720	0.5728
marriage	0.3195	0.5658
cake	0.1699	0.3629
love	0.0148	0.2449
honeymoon	0.0183	0.2262
jason	0.2081	0
chris	0.1461	0

Table I
Effect of denoising on the tag “**wedding**” in Flickr dataset.

as expected, that the consistency strength between within-logical-itemset pairs grows and consistency strength between across-logical-itemset-pairs shrinks as seen in Table I.

The iterative denoising algorithm uses the co-occurrence *consistencies* obtained in the previous iteration to remove noisy co-occurrence *counts* in the next iteration, recompute the margin and total from these cleaned up counts and then, compute the consistencies in the next iteration. Let $\psi^{(t)}(\alpha, \beta)$ and $\phi^{(t)}(\alpha, \beta)$ denote co-occurrence counts and co-occurrence consistencies in the t^{th} iteration of the denoising procedure. Then, denoising, using the following update $\forall(\alpha, \beta) \in \mathbf{V} \times \mathbf{V}$: $\psi^{(0)}(\alpha, \beta) > \theta_{cooc}$,

$$\psi^{(t+1)}(\alpha, \beta) \leftarrow \psi^{(0)}(\alpha, \beta)\delta(\phi^{(t)}(\alpha, \beta) > \theta_{consy}) \tag{10}$$

The margin and total counts are updated in each iteration as well, using Equations (3) and (4). Note that, we don't need to make another pass through the data, but just reuse the co-occurrence counts computed in the first iteration. As denoising happens, the overall *quality* of the resulting consistency matrix improves. The following quality is measured in each denoising iteration.

$$Q(\Phi^{(t)}) = \sum_{(\alpha,\beta)\in\mathbf{V}\times\mathbf{V}} P^{(t)}(\alpha, \beta)\phi^{(t)}(\alpha, \beta). \tag{11}$$

Empirically, we observed that denoising converges very quickly in two to three iterations, where convergence is measured by the fraction of co-occurrence counts that become zero in any iteration. A significant improvement was seen in the quality of the final logical itemsets obtained after the denoising procedure. Table I shows how the iterative procedure affects the consistency of tag `wedding` with some other tags of FLICKR dataset. Here the consistency of tag `wedding` with relevant tags `dress` & `reception` increases significantly after the first iteration itself and decreases to zero for irrelevant tags such as `chris` & `jason`.

To give an idea about the final consistency matrix after denoising, the top consistent tags associated with some random tags are shown in Table II.

D. Stage 4: LISM-Discovery

The first three stages provide several knobs to robustly reduce the mixture-of-intents noise in the data: (i) ignoring very low frequency co-occurrence counts via the threshold (θ_{cooc}), (ii) converting these counts into consistencies via prior normalization (iii) ignoring low co-occurrence consistencies via the threshold θ_{cons}, and (iv) iterative denoising of co-occurrence consistency. By this time, it is expected that:

- **Intra (within) Logical Itemset Consistencies are high**: While the projection property suggests that the entire logical itemset has a very low support, subsets of logical itemsets will still have high consistency because every time at least two products (tags) from the same intent (concept) are in the same market basket (tagset), co-occurrence consistency between them goes up.
- **Inter (across) Logical Itemset Consistencies are low**: This is related to the mixture-of-intents noise removal in the first three stages. Any noise related to cross intention products or cross concept tags will be removed in the first three stages.

The sparse and symmetric co-occurrence consistency graph is thresholded and binarized such that an edge between two items is present if their co-occurrence consistency is above the threshold θ_{consy}.

A logical itemset, given this graph, is defined as a set of items $\mathbf{L} = (\ell_1, \ell_2, ..., \ell_k)$ such that each item in this set has a high co-occurrence consistency with all other items in this set. In order to find the largest logical itemsets, we just have to find all *maximal cliques* in the binarized co-occurrence consistency graph. The problem of finding all maximal cliques in a graph is NP-hard with the worst case

Tag	Most consistent tags in **IMDB dataset**
food	lifestyle, money, restaurant, drinking, cooking
road	truck, motorcycle, car, road-trip, bus
singer	singing, song, dancing, dancer, musician
suicide	suicide-attempt, hanging, depression, mental-illness, drowning
hospital	doctor, nurse, wheelchair, ambulance, car-accident

Tag	Most consistent tags in **Flickr dataset**
art	painting, gallery, paintings, sculpture, artist
france	paris, french, eiffeltower, tower, europe
island	tropical, islands, newzealand, thailand, sand
animals	zoo, pets, wild, cats, animal
airplane	flying, airshow, fly, military, aviation

Table II
Top 5 most consistent tags from the IMDB and Flickr datasets

Property	Flickr	IMDB
Original data size	3,546,729	449,524
Original vocab size	656,291	120,550
Final data size	2,710,578	395,802
Original Keywords/bag	5.42	9.13
Cleaned keywords/bag	2.94	5.13

Table III
Characteristics of Flickr and IMDB datasets

time complexity of $\mathcal{O}(3^{n/3})$ for a graph with n vertices [20]. A large amount of improvements have been made over classical algorithm by Bron and Kerbonsch [19] (e.g. [21]) for finding all maximal cliques.

Dharwadker [22] proposed a approximate polynomial-time algorithm (polynomial to the number of vertices) for finding a maximal clique in all known examples of graphs. The algorithm stops as soon it finds a maximal clique of fixed size k and during this process finds maximal cliques of size $< k$. We used this algorithm for finding maximal cliques of different sizes in our binarized co-occurrence consistency graph by setting a very high value of k.

IV. Experiments

A. Datasets Used

Typical benchmark datasets used in FISM are located at the FIMI repository [5][6]. In these datasets, the vocabulary is obfuscated as the focus is mostly on scale improvements and not quality improvements. Since our focus is primarily on quality improvements and to demonstrate the logical itemsets discovered, we worked with two datasets, Flickr [23] [7] tagsets and IMDB keywords[8], where the item dictionary is available and the number of data points is large. Due to the large data size and memory and computational limitations, the following preprocessing was applied to both datasets:

- Compute frequency of all items (keywords in this case),
- Keep only top 1000 most frequent keywords,
- Remove low freq. keywords from each tagset
- Remove all bags with less than two keywords[9]

Table III shows the Original and Cleaned data statistics.

B. Logical Itemsets Discovered

Our main result is the quality of the Logical itemsets discovered. Table VI and IV contain examples of logical itemsets found in the Flickr and IMDB datasets respectively. These tables are sorted first in descending order of itemset size and for each size in descending order of frequency in the data. There are three key properties to notice about these logical itemsets discovered:

- **Large sizes**: As shown in figure 2, the number of frequent itemsets generated grows exponentially with itemset size. In contrast, large (> size 5) logical itemsets are easily found in the data. Since we are using a clique finding algorithm, the main complexity comes from the NP-hard problem of finding all maximal cliques in the graph. Various parameters such as the count and consistency thresholds might be used to control the sparsity of the co-occurrence consistency graph and therefore the complexity, noise, and quality of the logical itemsets obtained. On these datasets, FISM took prohibitively large number of resources (ram, cpu) that for sizes above 5, we were not even able to generate maximal frequent itemsets on these two datasets.

Size	Freq.	Logical Itemset
7	18	family-relationships father-daughter-relationship mother-daughter-relationship brother-sister-relationship teenage-girl sister-sister-relationship girl
7	11	husband-wife-relationship marriage infidelity adultery extramarital-affair unfaithfulness affair
6	4331	conundrum vocabulary number-game math-whiz word-smith oxford-english-dictionary
6	83	photographer judge competition model photography fashion
6	37	family-relationships father-son-relationship mother-son-relationship brother-brother-relationship brother-sister-relationship dysfunctional-family
6	26	lawyer judge trial courtroom witness court
5	88	murder police detective police-detective murder-investigation
5	25	robbery thief theft bank-robbery bank
5	18	police policeman police-officer police-station police-car
4	685	murder killer killing murderer
4	67	blood gore zombie splatter
4	43	seduction obsession voyeurism voyeur
4	24	spy espionage secret-agent british
4	18	brother mother father sister
3	902	deception duplicity deceit
3	309	blood knife stabbing
3	123	gun shooting criminal
3	79	superhero mask based-on-comic-book
3	36	rifle shooting revolution
2	1116	martial-arts kung-fu

Table IV
Examples of Logical Itemsets discovered in IMDB dataset

Size	Freq.	Logical Itemset
10	18	airplane airport plane flying flight aircraft air jet ..
7	35	music rock concert show band live guitar
7	0	animals africa wildlife lion rhino safari elephant
6	39	baby kids children boy child kid
6	35	beach sea ocean sand surf waves
6	24	cat cats cute pet kitten kitty
6	13	nyc newyork newyorkcity manhattan ny brooklyn
5	15	bike race motorcycle racing motorbike
5	1	fire police ambulance rescue emergency
4	7	light reflection window glass
4	6	mountain hiking hike trail

Table V
Rare Logical Itemsets discovered in Flickr dataset

- **Meaningful Logical Itemsets**: LISM is developed with the promise of noise-robustness and high quality results. Note that almost all the logical itemsets discovered in both datasets are quite meaningful and have very little noise. Compared to the results obtained in [18], the results obtained here, on even much larger datasets, are significantly better.

[6] http://fimi.ua.ac.be/

[7] http://www.flickr.com

[8] http://www.imdb.com

[9] since single items don't contribute to pairwise co-occurrences

- **Low frequencies**: Unlike FISM that looks for high frequency itemsets, LISM is *frequency agnostic*. Once the co-occurrence consistency graph is created, the logical itemsets are discovered directly from the graph. Statistics on these logical itemsets discovered is computed in the second pass through the data. Tables VII and V show examples of rare itemsets discovered by LISM in IMDB and Flickr datasets respectively.

Thus overall, we conclude that LISM generates a small number of high quality, latent itemsets while FISM produces a very large number of noisy, observed itemsets from the data. This is possible because **fundamentally, LISM decouples the observed noisy bag-of-items data from the latent logical itemsets via a highly noise free and scalable co-occurrence consistency graph**. This unique and novel property of LISM makes it highly effective in dealing with the bag-of-items data compared to the FISM and other frequency based frameworks.

Size	Freq.	Logical Itemsets
7	194	girl woman model beautiful sexy beauty pretty
6	877	racing sport performance team 911 motorsport
5	1821	rescue shelter found hurricanekatrina nola
5	164	art street graffiti streetart stencil
5	157	blue red green yellow purple
5	148	tree fall autumn leaves leaf
5	85	sky sunset clouds sun sunrise
5	78	city architecture building downtown buildings
4	838	ice sports youth hockey
4	504	london england uk unitedkingdom
4	475	travel vacation islands holidays
4	469	bridge francisco golden gate
4	446	paris france tower eiffeltower
4	414	lake mountains hiking climbing
4	369	california usa sanfrancisco roadtrip
4	333	germany football soccer worldcup
4	327	nikon digital camera set
4	327	canada vancouver bc britishcolumbia
4	324	fish diving underwater scuba
4	313	snow winter ice cold
4	288	europe spain madrid espa
4	244	travel vacation trip adventure
4	229	me portrait selfportrait self
3	1609	china beijing greatwall
3	1212	germany berlin deutschland
3	563	wedding family reception
3	387	washington july fireworks
3	312	art museum history
3	213	family mom dad
3	187	art sculpture statue
3	86	street road sign

Table VI
Examples of Logical Itemsets discovered in Flickr dataset

Size	Freq	Logical Itemset
13	0	independent-film student-film experimental women educational human-rights asian alternative underground-film hispanic docudrama asian-american ..
10	7	satire parody television celebrity sketch-comedy joke actor-playing-multiple-roles comedian humour entertainment
10	1	school teacher teenage-girl student teenage-boy high-school bully classroom basketball teacher-student-relationship
9	5	religion church christian prayer god catholic bible christianity faith
9	0	dog cat bird anthropomorphism anthropomorphic-animal rabbit mouse pig cow
8	9	singer song singing musician piano guitar band concert
8	0	beach boat island swimming fishing sea fish ocean
8	0	boat island ship fishing sea fish ocean storm
7	4	castle king princess fairy-tale witch queen prince
7	4	interview actor behind-the-scenes film-making making-of filmmaking director
7	3	funeral cemetery death-of-mother coffin graveyard grief grave
7	0	nightmare guilt hallucination insanity psychiatrist paranoia mental-illness
7	0	forest nature river mountain lake woods tree
7	0	boat nature water river fishing lake fish
6	5	alien robot future outer-space spaceship space
6	2	ghost fear nightmare hallucination supernatural-power curse
6	0	politics corruption politician mayor election speech
6	0	children christmas girl orphan little-girl doll
6	0	computer scientist science time-travel future outer-space
6	0	maid inheritance mansion wealth butler servant
6	0	politics corruption politician mayor election speech
5	9	doctor hospital nurse ambulance heart-attack
5	5	scientist science professor laboratory experiment
5	2	reporter newspaper politician journalist scandal
5	0	artist painting obsession photography writing
5	0	ghost supernatural demon witch curse
4	7	kidnapping bound-and-gagged abduction tied-to-chair
4	2	anime student japan based-on-manga
4	1	betrayal corruption conspiracy greed

Table VII
Examples of **rare** Logical Itemsets discovered in IMDB dataset

V. Conclusion

Efficient frequent itemset mining is used ubiquitously for finding *interesting* and actionable insights in bag-of-items data in a variety of domains such as retail, text, vision, biology, etc. In this paper, we propose an alternate framework to FISM, called the Logical Itemset Mining that is simple, scalable, and highly effective in dealing with the *mixture-of-intents-noise* and *projection-of-intents-incompleteness* of bag-of-items data. Results on two large bag-of-items datasets demonstrate the high quality of logical itemsets discovered by LISM. The LISM framework is highly *noise-robust*, uses only two passes through the transaction data and stores only sparse pair-wise statistics and is therefore, highly scalable. It is able to discover logical itemsets, that are not obvious in the data and is also able to generalize to novel logical itemsets with zero support.

LISM can be improved in several ways: (i) Instead of using binarized graph to find logical itemsets, the original weighted co-occurrence consistency graph can be used to find *soft* logical itemsets as opposed to the *hard* logical itemsets, as in the current version. Further, as frequent itemsets has been extended to *indirect* frequent itemsets, similarly, it is straightforward to find *higher order co-occurrence consistencies* that span across items, that don't have direct co-occurrences in the data. Finally, scaling and parallelizing the maximal clique finding algorithms and extending them to the notion of soft maximal cliques will make LISM even more practical for larger datasets and for a variety of application.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami, "Mining association rules between sets of items in large databases," *SIGMOD*, pp. 207–216, 1993.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," *VLDB*, pp. 487–499, 1994.

[3] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *SIGMOD*, pp. 1–12, 2000.

[4] M. J. Zaki, "Scalable algorithms for association mining," *IEEE TKDE*, vol. 12, pp. 372–390, 2000.

[5] R. J. B. Jr., B. Goethals, and M. J. Zaki, Eds., *Workshop on FIMI, ICDM*, ser. CEUR Workshop Proceedings, vol. 126, 2004.

[6] C. Borgelt, "Recursion pruning for the apriori algorithm," *Workshop on FIMI, ICDM*, 2004.

[7] S. Orlando, C. Lucchese, P. Palmerini, R. Perego, and F. Silvestri, "kdci: a multi-strategy algorithm for mining frequent sets," *Workshop on FIMI, ICDM*, 2003.

[8] T. Uno, M. Kiyomi, and H. Arimura, "Lcm ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets," *Workshop on FIMI, ICDM*, 2004.

[9] P. Tan, V. Kumar, and J. Srivastava, "Indirect association:mining higher order dependencies in data," *PKDD*, pp. 632–637, 2000.

[10] Q. Wan and A. An, "Efficient mining of indirect associations using hi-mine," *CAIAC*, 2003.

[11] A. Sheth, B. Aleman-Meza, F. S. Arpinar *et al.*, "Semantic association identification and knowledge discovery for national security applications," *Journal of Database Management*, vol. 16, pp. 33–53, 2005.

[12] W.-Y. Lin and Y.-C. Chen, "Emia: A new efficient algorithm for indirect associations mining," *GrC*, pp. 404–409, 2011.

[13] W.-Y. Lin, Y.-E. Wei, and C.-H. Chen, "A generic approach for mining indirect association rules in data streams," *IEA/AIE (1)*, pp. 95–104, 2011.

[14] L. Szathmary, P. Valtchev, and A. Napoli, "Finding minimal rare itemsets and rare association rules," *KSEM*, pp. 16–27, 2010.

[15] L. Szathmary, A. Napoli, and P. Valtchev, "Towards rare itemset mining," *ICTAI (1)*, pp. 305–312, 2007.

[16] D. J. Haglin and A. M. Manning, "On minimal infrequent itemset mining," *DMIN*, pp. 141–147, 2007.

[17] A. M. Manning and D. J. Haglin, "A new algorithm for finding minimal sample uniques for use in statistical disclosure assessment," *ICDM*, pp. 290–297, 2005.

[18] H. Liu, P. LePendu, R. Jin, and D. Dou, "A hypergraph-based method for discovering semantically associated itemsets," *ICDM*, pp. 398–406, 2011.

[19] C. Bron and J. Kerbosch, "Finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[20] E. Tomita, A. Tanaka, H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments" *Theoretical Computer Science*, vol. 363(1), pp. 28-42 (2006).

[21] S. Tsukiyama, M. Ide, I. Ariyoshi, and I. Shirakawa, "A new algorithm for generating all the maximal independent sets," *SIAM Journal of Computing*, vol. 6, no. 3, pp. 505–517, 1977.

[22] A. Dharwadker, "The clique algorithm," 2006, available at http://www.geocities.com/dharwadker/clique/.

[23] X. Li, C. G. M. Snoek, and M. Worring, "Learning social tag relevance by neighbor voting," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1310–1322, November 2009.

[24] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *NIPS*, pp. 601–608, 2001.

[25] P. Tan, V. Kumar, and J. Srivastava, "Selecting the Right Interestingness Measure for Association Patterns", *SIGKDD* 2002.