Diverse Dimension Decomposition of an Itemset Space

Mikalai Tsytsarau
University of Trento
tsytsarau@disi.unitn.eu

Francesco Bonchi
Yahoo! Research
bonchi@yahoo-inc.com

Aristides Gionis
Yahoo! Research
gionis@yahoo-inc.com

Themis Palpanas
University of Trento
themis@disi.unitn.eu

***Abstract*—We introduce the problem of diverse dimension decomposition in transactional databases. A dimension is a set of mutually-exclusive itemsets, and our problem is to find a decomposition of the itemset space into dimensions, which are orthogonal to each other, and that provide high coverage of the input database. The mining framework we propose effectively represents a dimensionality-reducing transformation from the space of all items to the space of orthogonal dimensions. Our approach relies on information-theoretic concepts, and we are able to formulate the dimension-finding problem with a single objective function that simultaneously captures constraints on coverage, exclusivity and orthogonality. We describe an efficient greedy method for finding diverse dimensions from transactional databases. The experimental evaluation of the proposed approach using two real datasets, `flickr` and `del.icio.us`, demonstrates the effectiveness of our solution. Although we are motivated by the applications in the collaborative tagging domain, we believe that the mining task we introduce in this paper is general enough to be useful in other application domains.**

I. Introduction

Collaborative content creation and annotation is one of the main activities and distinguishing features of the Web 2.0. The common efforts of many users create huge repositories of all sort of media, usually annotated by the users themselves; for instance photos (`flickr`), urls (`del.icio.us`), blogs (`technorati`), videos (`youtube`), songs (`last.fm`), scientific papers (`bibsonomy` and `citeulike`), and others. All these platforms provide their users with a repository of resources, and the capability of assigning tags, i.e., freely chosen keywords, to these resources.

A repository of tagged resources can be seen as a transactional database, typical of the frequent-itemset-mining paradigm: transactions correspond to resources, and items correspond to tags. In this setting we are interested in studying the problem of discovering an item-space decomposition, which we define to be a set of orthogonal dimensions with high coverage. A dimension in turn is defined to be a set of itemsets that rarely co-occur in the database.

Example 1: Consider for instance a query on `flickr` for photos about art (i.e., annotated with the tag `art`): the dataset $\mathcal{D}$ of such photos can look like the one in Figure 1. In this setting dimensions might be, for example, such sets of itemsets as {{`portrait`},{`landscape`}} or {{`canon`},{`nikon`},{`sony`}}. Indeed, almost all photos in the dataset contain at most one of the itemsets[1] from each of the two dimensions.

[1]While in this example each dimension is formed by singleton items, in general a dimension is formed by itemsets of any size.

In this paper, we are interested in discovering dimensions that represent diverse concepts, such as "type of photo" or "camera brand", and whose different values almost partition the dataset. For instance, each dimension in Figure 1 can be seen as a different way of partitioning the transactions in $\mathcal{D}$, and the three dimensions together can be considered as a diverse decomposition of the space of photos.

In order to achieve our goal, we adopt an information-theoretic perspective. While there exist several studies applying *joint entropy* to the problem of identifying interesting or informative itemsets [1]–[6], this body of work can not be applied to the problem of diverse dimension decomposition, as explained next.

Example 2: Consider the transposed view of the database in Figure 1, given in Table I. Following the approaches that use *joint entropy*, we will get sets (templates) such as {`color`, `nikon`}, having the highest entropy (dark grey lines), or {`landscape`,`sony`} as low-entropy sets (light grey lines).

We notice that high-entropy sets are characterized by more uniform appearance of their instantiations in the database (e.g., instances 01, 10 and 11 appear with roughly the same frequency), while low-entropy sets accumulate support around the few most-frequent instances (in our example: 00), not necessarily representing mutually exclusive items forming the dimension (with instances 001, 010, 100). Thus, using the existing interestingness measures does not solve our problem.

In this paper, we propose entropy measure expressing both the orthogonality among dimensions and the interestingness of dimensions. Moreover, we show that it also captures constraints both on exclusivity and coverage. Based on this measure, we formulate diverse dimension decomposition as the problem of finding an optimal set of k dimensions, minimizing an objective function that closely resembles the mutual information measure, except for a parameter α, which allows the analyst to trade-off between information loss and orthogonality of the dimensions.

Our contributions are summarized as follows.

- We introduce the novel problem of diverse dimension decomposition in transactional databases, as an optimization problem. We characterize our objective function and show that the selected dimensions explain well the underlying database.
- We prove a property that allows assessing the level of informativeness for newly-added dimensions, thus allowing to define criteria for terminating the decomposition.

t_1	{fish, art, film, portrait, tattoo, xpro, crossprocessed, nikon, skin, n80}
t_2	{sanfrancisco, black&white, building, art, stairway, fireescape, nikon}
t_3	{portrait, color, art, me, illustration, blood, adobephotoshop, canon}
t_4	{travel, brazil, plant, art, nature, color, strong, nikon, nikond70}
t_5	{sunset, art, museum, landscape, minneapolis, canon, powershotg3}
t_6	{sculpture, art, 2004, festival, japan, culture, clay, a70, canon}
t_7	{portrait, art, painting, color, europe, sony, sonyericssonk750}
t_8	{black&white, art, film, photograph, street-photo, contax645}
t_9	{art, black&white, skin, hand, bodypainting, nikon, d70}
t_{10}	{red, woman, art, face, color, tear, canon, eos300d}
t_{11}	{art, 3d, unfound, photositook, sony, cybershot}
t_{12}	{beautiful, woman, black&white, portrait, art}
t_{13}	{landscape, nature, sunrise, wallpaper, art}

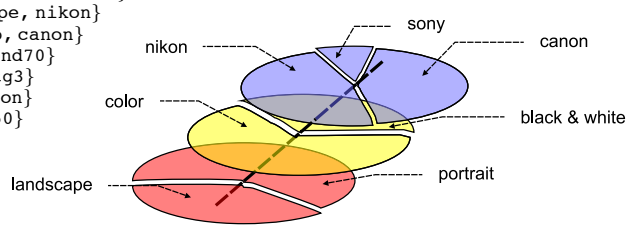

Fig. 1: An example of transactional dataset, having three diverse dimensions (shown on the right). In this specific example from Flickr, each transaction corresponds to a picture, and its associated tags. All pictures have in common the tag `art`.

item	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}
canon	0	0	1	0	1	1	0	0	0	1	0	0	0
nikon	1	1	0	1	0	0	0	0	1	0	0	0	0
sony	0	0	0	0	0	0	1	0	0	0	1	0	0
color	0	0	1	1	0	0	1	0	0	1	0	0	0
black&white	0	1	0	0	0	0	0	1	1	0	0	1	0
landscape	0	0	0	0	1	0	0	0	0	0	0	0	1
portrait	1	0	1	0	0	0	1	0	0	0	0	1	0

TABLE I: A transposed view of the dataset in Figure 1, showing most frequent items taken from several dimensions.

- We show that our problem is trivially NP-hard, and thus turn our interest to approximation algorithms. We propose a greedy algorithm exploiting the well known FP-tree data structure [7], and clever pruning of the search space, based on properties we prove in the characterization of the problem.
- We experiment the proposed approach using two real-world large datasets in the collaborative tagging domain, `flickr` and `del.icio.us`, demonstrating the effectiveness and scalability of our solution.

The rest of the paper is structured as follows. In the next section we discuss related work and in Section III we formally define the problem of mining diverse dimensions from a transactional dataset. In Section IV we present our methods, while in Section V we report experimental assessment. Finally, we discuss future work and conclude in Section VI.

II. Related Work

We next survey the literature related to our work, dividing it into three independent groups: (a) methods that aim at extracting diverse content from web data, (b) space-like representations of itemset databases, and (c) entropy-based measures for itemset interestingness.

A. Diversity in Information Retrieval

Extracting a set of diverse dimensions, that covers the various aspects of the underlying dataset, can be seen as a problem of automatic facet discovery. Such a facet-discovery process has many applications in improving user experience, for instance, tag recommendation [8], search and exploration [9], tag clustering [10]–[12], and more. Although in this paper we deal with the fundamental problem of diverse dimension decomposition in general transactional database, we believe that our proposals can be applied in these problems, and we are indeed motivated by the collaborative tagging domain, as witnessed by our experiments in Section V.

Web search is another domain in which finding an answer set with diversity is important. Several studies have focused on the problem of search engines query-result diversification [13]–[16], where the goal is to produce an answer set that includes results relevant to different aspects (facets) of the query. In this area, the work mostly related to ours is the paper by Bonchi et al. [13] where the problem of topical query decomposition is introduced. Given a query and a document retrieval system, the goal is to select a small set of queries representing coherent, conceptually well-separated topics, and whose union of resulting documents corresponds approximately to that of the original query. The authors propose two methods, one based on a special instance of the weighted set covering problem, and one based on constrained clustering.

B. Space-like Representation of Itemset Databases

Traditionally, in association rule mining, itemsets are represented as binary vectors in the space of items: each axis corresponds to an item, and binary coordinate values indicate whether each particular item is contained in the itemset. This representation works well, if we are interested in finding association rules of the form $\{\texttt{bread,milk}\} \Rightarrow \{\texttt{butter}\}$, which capture itemset-level correlations in data. However, binary coordinates do not facilitate geometric decompositions of the item space (which can be interpreted by a human).

As a possible solution, Korn et al. [17] used real-valued coordinates, where coordinates could be interpreted as quantities of each item employed in the construction of rules. This framework allowed to perform spectral decomposition of the item space (similar to SVD [18]), and discovery of Ratio Rules, i.e., quantitative correlations between itemsets in data. An example of such rule is $\{\texttt{1:bread,2:milk,5:butter}\}$, which says that a typical ratio of bread, milk and butter within the itemsets is $\{\texttt{1:2:5}\}$, so we can predict missing values of different items given these rules.

Alternatively, one can represent a database in the transposed space of transactions rather than items (like the one shown in Table I). This is the main idea behind the "geometrically inspired itemset mining" framework proposed by Verhein and Chawla [19]. Their proposal is a framework for frequent itemset mining, which can accept space transformations, such as SVD, subject to the constraint that a measurement function should be able to be computed in the new space. For instance, in the case of SVD, each new axis represents a linear combi-

nation of transactions, featuring the largest variance in data. However, such a transformation is not very easily interpretable.

Our work is different in that we propose a principled method for decomposing the space of items in a set of orthogonal dimensions that are readily interpretable. Moreover, our problem formulation is based on information theory, and is capable of identifying dimensions in transactional databases in general, regardless whether transactions have real values associated with items or not.

C. Entropy-Based Measures of Itemset Interestingness

Knobbe and Ho [1], employing Information Theory, define a measure for itemset interestingness, *joint entropy*, which is optimizing for the uniform co-occurrence among items. In their terminology, a set is a template (or a collection of attributes taking binary values), whose instances are itemsets. Entropy is calculated as a negative sum of logarithm-multiplied occurrence probabilities for observed instances. This measure indicates how likely a randomly-chosen set instance is to appear in data. The same authors also introduced a notion of "pattern teams" [2], that can be seen as feature sets. They theoretically evaluate the effectiveness of different filtering criteria for feature sets used in machine learning classifiers, noticing that *joint entropy* does not satisfy the intuitions we use for dimensions (i.e. mutual exclusivity, high coverage). Instead, the authors find that *exclusive coverage* (i.e. the sum of coverages minus co-occurrences) is much more suitable as a measure optimizing for these intuitions.

Continuing the above line of research, Heikinheimo et al. define two related problems, namely, mining high- and low-entropy sets [5]. Zhang and Masseglia [6] extended their method to work on streaming data and proposed to reduce its output by removing similar sets according to criteria based on mutual information [20].

Finally, Tatti [3] and Mampaey et al. [4] proposed to use joint entropy in an MDL optimization framework, aiming at compressing the database. Maximizing the entropy ensures that all the pattern subsets are uniformly distributed, while the limit on pattern frequency (according to the exponential frequency decrease assumption) facilitates the selection of frequent patterns.

Although these papers deal with itemset mining using joint entropy, their goal is different from ours: they aim at extracting sets of items, which co-occur in the database uniformly (when optimized for high entropy: same frequency for all subset combinations) or sparsely (when optimized for low entropy: only certain subsets are frequent). We discussed the difference between these approaches and our proposal earlier, in Example 2.

In contrast to the above methods, we formulate the entropy of a dimension as the uncertainty of the dimension's itemsets for each document, and use it as an indicator of quality for dimensions. Moreover, our goal is to find sets of itemsets (not items), which are not only mutually exclusive (within each dimension), but also independent (across dimensions).

III. Problem Statement

We are given a transactional dataset $\mathcal{D}$, i.e., a multiset of transactions $t \subseteq \mathcal{I}$, where $\mathcal{I}$ is a ground set of items. An example of a transactional dataset is given in Figure 1. As usual we call itemset any set of items $X \subseteq \mathcal{I}$, and we denote by $\mathcal{D}(X)$ its supporting set of transactions, i.e., $\mathcal{D}(X) = \{t \in \mathcal{D} \mid X \subseteq t\}$. Moreover we denote by $\mathbb{I}$ the space of all possible itemsets on $\mathcal{I}$.

In this paper we are studying the following problem. We are given an integer k and the goal is to discover a collection of k dimensions, that decompose the itemset space $\mathbb{I}$. Moreover, we want each dimension to almost partition the dataset $\mathcal{D}$; that is to say, we want (almost) all transactions $t \in \mathcal{D}$ to contain one and only one of itemsets from the dimension.

Definition 1 (Dimension): Given an itemset space $\mathbb{I}$, a dimension $\delta^i \subset \mathbb{I}$ is a collection of pairwise disjoint itemsets, i.e., $\delta^i = \{X^i_0, ..., X^i_m\}$, such that for all pairs of itemsets $X^i_k, X^i_l \in \delta^i$ with $l \neq k$ it holds $X^i_k \cap X^i_l = \emptyset$.

As in decomposition methods in linear algebra, we want to decompose the itemset space in dimensions that can be though as "orthogonal." While orthogonality in linear algebra is a well-understood concept, when talking about the itemset space the concept of orthogonality is much less clear. Motivated by our example, we would like to argue that the dimension *camera-brand* = {{`canon`},{`nikon`},{`sony`}, ...} is orthogonal to the dimension *type-of-photograph* = {{`portrait`}, {`landscape`},{`street-photo`}, ...}. The concept of orthogonality can thus be formulated as independence among the dimensions: the fact that a photograph is tagged by `nikon` should not reveal any information about the type of the photograph. That is, the likelihood of that photograph being `portrait` or `landscape` should remain the same as it is non conditional on *camera-brand*.

To formalize the above intuition, we use the concept of *mutual information*. Given two random variables, X and Y, mutual information measures the information shared between them. For example, if X and Y are independent, then knowing X does not give any information about Y and vice versa, so their mutual information is zero. In order to employ the definition of mutual information, we need to define precisely how our dimensions define a probability space, and what is the entropy of this probability space. We provide those definitions in the next section.

In addition to finding orthogonal dimensions we also want to find "useful" dimensions, in the sense of being able to explain the dataset succinctly. We express this intuition by the concept of *coverage*. In the previous example, the dimension *camera-brand* has high coverage because most of the photos have one tag from its collection of itemsets {{`canon`},{`nikon`}, ...}. We are able to show that the concept of coverage can also be formulated in an information theoretic manner. More importantly, we are able to combine both desiderata, high coverage and orthogonality, in one single objective function, achieving to simplify our problem formulation as well as the mining algorithm.

A. Entropy of Dimensions

Our goal is to define the entropy $H(\delta^i \mid \mathcal{D})$ of the dimension $\delta^i = \{X_0^i, ..., X_m^i\}$ of the itemset space $\mathbb{I}$, on the dataset $\mathcal{D}$. We first define the entropy of the dimension δ^i conditioned on a single transaction t of the dataset.

$$H(\delta^i \mid t) = -\sum_{X^i \in \delta^i} P(X^i \mid t) \log P(X^i \mid t)$$

The probabilities $P(X^i \mid t)$ express the uncertainty that the itemset X^i is present in the transaction t, and are defined later in the section. Averaging over all transactions of the dataset $\mathcal{D}$, we now define the entropy of the dimension δ^i as follows:

$$H(\delta^i \mid \mathcal{D}) = \sum_{t \in \mathcal{D}} P(t) H(\delta^i \mid t),$$

where $P(t)$ is the frequency of each transaction in the dataset. For instance, if all transactions are distinct, then $P(t) = 1/|\mathcal{D}|$.

The conditional entropy of one dimension given another, is calculated similar to an ordinary entropy, but counting only documents assigned to itemsets in a dimension being conditioned. Entropies for each itemset are then aggregated with respect to their probabilities, that is:

$$\begin{aligned} H(\delta^i \mid \delta^j, \mathcal{D}) &= \frac{1}{|\delta^j|} \cdot \sum_{X^j \in \delta^j} \frac{H(\delta^i \mid X^j, \mathcal{D})}{\sum_{t \in \mathcal{D}} P(X^j \mid t)}, \quad \text{where} \\ H(\delta^i \mid X^j, \mathcal{D}) &= \sum_{t \in \mathcal{D}} P(X^j \mid t) \cdot H(\delta^i \mid X^j, t), \quad \text{and} \\ H(\delta^i \mid X^j,\ t) &= -\sum_{X^i \in \delta^i} P(X^i \mid X^j, t) \cdot \log P(X^i \mid X^j, t). \end{aligned}$$

It remains to define the probabilities $P(X^i \mid t)$, which can be interpreted as the probability of an itemset being relevant for a transaction. When computing relevancy probabilities, we may use different set similarity measures, such as *cosine similarity* (1), *Jaccard coefficient* (2) or *binary inclusion/exclusion* (3):

$$P(X^i \mid t): \ \frac{|X^i \cap t|}{|X^i| \cdot |t|} \ (1); \ \frac{|X^i \cap t|}{|X^i \cup t|} \ (2); \ \begin{cases} 1, & X^i \subseteq t; \\ 0, & X^i \not\subseteq t. \end{cases} \ (3);$$

Also note that after computing the set similarity measures we need to normalize them in order to arrive to a valid probability distribution whose values sum up to 1. The following example describes the meaning of different probability distributions.

Example 3: Let us consider a dimension δ^i containing five itemsets:{{`canon`},{`nikon`},{`olympus`},{`pentax`},{`sony`}}. Each transaction t in the dataset may be relevant to one or several itemsets of the dimension, or not relevant at all. Figure 2 shows three different transactions with the following probability distributions: $t_1 =$ {`pentax,camera,test`} is relevant only to {`pentax`}, with a probability 1.0; $t_2 =$ {`pentax,nikon,test`} is relevant to only two cameras, with probabilities 0.5; $t_3 =$ {`dslr,cameras,test`} may be relevant to any camera, thus resulting in equal probabilities and maximal entropy. In this example, entropy reflects the uncertainty of the dimension being relevant to a transaction. When only one itemset is relevant we have low entropy, as in the first case. When none of the itemsets is more relevant, resulting in the unclear choice, the entropy becomes high.

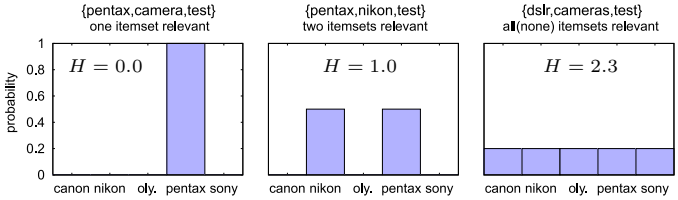

Fig. 2: Entropy for different probability distributions.

B. Problem Formulation

As we mentioned before, the problem we consider is to discover k diverse dimensions that explain well the input dataset. Let us denote by $\Delta = \{\delta^1, \ldots, \delta^k\}$ such a set of k dimensions. Our objective function evaluates the goodness of the dimension set Δ in terms of entropy and diversity. We define those next.

Definition 2 (Entropy of dimension set): Given a set of dimensions $\Delta = \{\delta^1, \ldots, \delta^k\}$, its entropy is defined as the sum of entropies of its dimensions. [2]

$$H(\Delta) = \sum_{\delta^i \in \Delta} H(\delta^i)$$

Definition 3 (Diversity of dimension set): Given a set of dimensions $\Delta = \{\delta^1, \ldots, \delta^k\}$, its diversity is calculated as the sum of conditional entropies over all pairs of its dimensions.

$$DIV(\Delta) = \sum_{\delta^i, \delta^j \in \Delta} H(\delta^i \mid \delta^j)$$

Central to our problem is the concept of mutual information, which we define here for a pair of dimensions δ^i and δ^j.

Definition 4 (Mutual Information):

$$I(\delta^i; \delta^j) = H(\delta^i) - H(\delta^i \mid \delta^j) = H(\delta^j) - H(\delta^j \mid \delta^i).$$

Mutual information of two dimensions is symmetric and is computed by taking the difference between an entropy of the first dimension, $H(\delta^i)$, and its conditional entropy given another one, $H(\delta^i \mid \delta^j)$. The latter entropy expresses the amount of information which one dimension contains about another, and we want this amount to be low (this happens when the conditional entropy of dimension δ^i remains large after we have identified dimension δ^j). In order to evaluate the goodness of the set of dimensions Δ we are summing the mutual information among all pairs of dimensions of the set Δ. We are now ready to formally define our problem.

Problem 1 (Diverse Dimension Decomposition): Given a dataset $\mathcal{D}$, find a set of k dimensions Δ that minimize $f(\Delta)$:

$$f(\Delta) = \left[H(\Delta) - \frac{2\alpha}{k-1} \cdot DIV(\Delta) \right] \quad (1)$$

In the above problem definition, we propose using an optimization function $f(\Delta)$ derived from mutual information.

Additionally, we introduce a parameter α to control the effect of entropy and conditional entropy over the optimization criterion. One can notice that the value of $\alpha = 1$ corresponds

[2]Throughout our paper we assume that all entropies are calculated with respect to the dataset $\mathcal{D}$, omitting it in order to simplify the notation.

to the case when the criterion is based precisely on the pairwise sum of mutual informations, but we may pick any other positive real value. This gives us the possibility to optimize either for information loss (when α is small, e.g., $\alpha = 0$), orthogonality (when α is large, e.g., $\alpha = 1$), or for both (when α takes an intermediate value).

Furthermore, we are able to show that by minimizing the objective function (1) we are also ensuring that the resulting dimensions explain well the underlying dataset. We first define the notion of coverage of a dimension.

Definition 5 (Coverage of a dimension): Coverage $C(\delta)$ of the dimension δ on the dataset $\mathcal{D}$ is the fraction of transactions t in $\mathcal{D}$, for which $t \cap X \neq \emptyset$, for some itemset $X \in \delta$.

Definition 6 (Maximal co-occurrence of a dimension): We define the maximal co-occurrence $R(\delta)$ between any number of itemsets in the dimension δ on the dataset $\mathcal{D}$ as the fraction of transactions t in $\mathcal{D}$ which contain more than one $X \in \delta$.

The following two lemmas are needed in our exposition that minimizing $f(\Delta)$ ensures high coverage.

Lemma 1: If the value of the objective function is less than a threshold, $f(\Delta) \leq \psi$, then

$$H(\Delta) \leq \frac{\psi}{1-\alpha}.$$

Proof: For all pairs of dimensions δ^i and δ^j, we have that $H(\delta^i \mid \delta^j) \leq H(\delta^i)$, what implies that $I(\delta^i; \delta^j) \geq 0$. In case of a pairwise sum, $DIV(\Delta) \leq H(\Delta) \cdot (k-1)/2$. Consequently, if $[H(\Delta) - DIV(\Delta) \cdot 2\alpha/(k-1)] \leq \psi$ we have that $[H(\Delta) - \alpha \cdot H(\Delta)] \leq \psi$, or equivalently, $H(\Delta) \cdot (1-\alpha) \leq \psi$, which proves the lemma. ■

This lemma predicts that for values of $\alpha \geq 1$ the entropy becomes unbounded. In other words, when optimizing solely for orthogonality the quality (entropy) of dimensions may become uncontrollable as some of them can be added to a collection solely because of their high independence to others. This can happen for dimensions that have negative contributions to $f(\Delta)$ because of a high α.

Lemma 2: Let δ be a dimension with m itemsets, and consider the case that the probabilities $P(X^i \mid t)$ take binary values. Then for the coverage $C(\delta)$ of the dimension δ it should be

$$C(\delta) \geq 1 - \frac{H(\delta)}{|\mathcal{D}| \log m}.$$

Proof: Entropy takes its maximum value in the case that a transaction is not covered by a dimension δ. Thus, we have, $H(\delta \mid t) = \log m$. Therefore, the maximal number of not covered transactions would be less than $H(\delta)$ divided by the maximum entropy. Thus, $(1-C(\delta))|\mathcal{D}| \leq H(\delta)/\log m$, which proves the lemma. ■

Lemma 3: If probabilities $P(X^i \mid t)$ are computed using binary similarities, then maximal co-occurrence R between any two itemsets in a dimension δ^i should be less than its entropy per single transaction: $R(\delta) \leq \frac{H(\delta^i)}{|\mathcal{D}|}$.

Proof: For a dimension δ^i, let s be the number of co-occurring itemsets in a transaction t, where $2 \leq s \leq |\delta^i|$. Then $P(X^i \mid t) = \frac{1}{s}$; and $H(\delta^i|t) = -s\frac{1}{s}\log\frac{1}{s} = \log s$. Therefore, the minimal entropy of single co-occurrence would be equal to $\log 2$. The maximal number of how many times the two itemsets may co-occur would be $H(\delta^i)$ divided by min entropy. Therefore $R(\delta)|\mathcal{D}| = H(\delta^i)/\log 2 = H(\delta^i)$. ■

We are now stating the theorem that small values of $f(\Delta)$ imply high coverage. The theorem is a direct consequence of Lemmas 1 and 2.

Theorem 1: Let $\Delta = \{\delta^1, \ldots, \delta^k\}$ be a set of k dimensions and $C(\Delta)$ be their total coverage, defined as $C(\Delta) = \sum_{\delta \in \Delta} C(\delta)$. Finally, let m_0 be the size of the smallest dimension of Δ. If $f(\Delta) \leq \psi$ then for the total coverage we have:

$$C(\Delta) \geq k - \frac{\psi}{|\mathcal{D}| \log m_0 (1-\alpha)}.$$

Proof: According to Lemma 2, the sum of dimensions coverages is greater than:

$$\sum_{\delta \in \Delta} C(\delta) \geq k - \frac{1}{|\mathcal{D}|} \sum_{\delta \in \Delta} \frac{H(\delta)}{\log m} \geq k - \frac{1}{|\mathcal{D}|} \sum_{\delta \in \Delta} \frac{H(\delta)}{\log m_0}$$

Applying our notation and using Lemma 1, we have:

$$C(\Delta) \geq k - \frac{H(\Delta)}{|\mathcal{D}| \log m_0} \geq k - \frac{\psi}{|\mathcal{D}| \log m_0 (1-\alpha)}$$ ■

We can use the above theorem to evaluate the quality of the dimensions, or to limit the number of dimensions in the result, e.g., by conforming to the specified constraint on the minimum coverage.

We next evaluate the dependency of $f(\Delta)$ over the number of dimensions k. Suppose that we have a set of k dimensions Δ, and want to add another dimension δ.

Theorem 2: Adding a candidate dimension δ will improve $f(\Delta)$ as long as its average mutual information (across dimensions Δ) is less than a fraction $(1-\frac{1}{2\alpha})$ of its total information.

Proof: The difference in the optimality value can then be written as follows:

$$diff = H(\delta) - \frac{2\alpha}{k} DIV(\Delta \cup \delta) + \frac{2\alpha}{k-1} DIV(\Delta)$$

$$diff \leq H(\delta) - \frac{2\alpha}{k} [DIV(\Delta \cup \delta) - DIV(\Delta)]$$

$$diff \leq H(\delta) - \frac{2\alpha}{k} \sum_{\delta^k \in \Delta} H(\delta|\delta^k)$$

We are interested in cases when this difference will be negative, what corresponds to improving optimality:

$$H(\delta) - \frac{2\alpha}{k} \sum_{\delta^k \in \Delta} H(\delta|\delta^k) \leq 0$$

$$(1-2\alpha) H(\delta) + \frac{2\alpha}{k} \sum_{\delta^k \in \Delta} I(\delta; \delta^k) \leq 0$$

$$\frac{1}{k} \sum_{\delta^k \in \Delta} I(\delta; \delta^k) \leq (1 - \frac{1}{2\alpha}) H(\delta)$$ ■

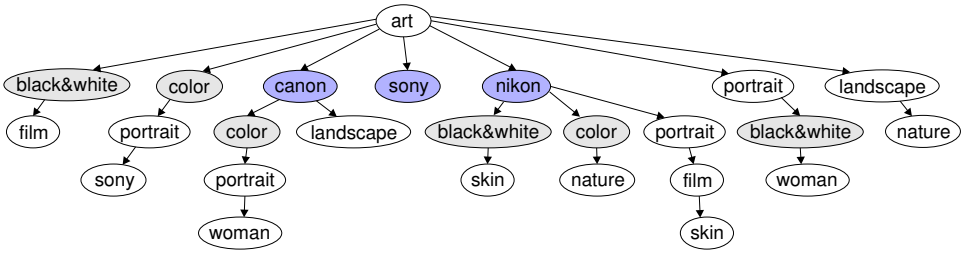

Fig. 3: FP-Tree constructed from the dataset shown in Figure 1. For the sake of simplicity, we omitted frequency counts from the nodes, cross-references among nodes and the header table, showing only the prefix-tree. To avoid having a too large figure, the tree is shown after a pruning of itemsets of frequency less than 2. Nodes highlighted in gray represent items from first-order dimensions, which are blocked and become transparent when considering itemsets for a new dimension (highlighted in blue).

In other words, $f(\Delta)$ will decrease when dimensions in Δ contain on average less than $\beta = 1 - 1/2\alpha$ percent information about δ. This property allows assessing the level of informativeness for newly-added dimensions, and defining criteria for terminating the decomposition. For example, if we stop adding new dimensions when $f(\Delta)$ starts to increase, we ensure that dimensions will not contain more than β percent of ambiguous information.

IV. Algorithm

We observe that Diverse Dimension Decomposition (Problem 1) is NP-hard, by reduction from the Set Partitioning problem, where we want to partition a set into non-overlapping and non-empty parts that cover the entire set. The above operation corresponds to the partition of items in a dimension. Though, our problem is more complex than that, since we are additionally seeking for a partition of the dimensions.

This inherent complexity of the problem makes any brute force approach unfeasible, even for relatively small instances of the problem. In the rest of this section, we describe our solution based on a greedy strategy.

Algorithm Outline. Since it is hard to come up with a good initial set of k dimensions for optimization, we propose identifying dimensions one-by-one, as follows. We start by constructing the first, more prominent dimension, according to our objective function $f(\Delta)$. This process begins with an empty single dimension, and on each iteration we decide whether to add new, or grow existing itemsets, according to the strategies discussed below. The construction of each dimension stops either if it is not possible to improve its optimality or if all items have been partitioned. Then, we do the same for the remaining dimensions iteratively, with the only difference that $f(\Delta)$ now takes into account all the previously identified dimensions, optimizing with respect to their orthogonality.

The Data Structure. To store the data for our problem, we adopt a compressed database representation in the form of the well known FP-Tree [7] data structure. In Figure 3, we show an example of such a tree, for the transactional dataset of Figure 1. This structure allows us to perform efficient pruning based on the coverage, co-occurrence and non-overlap (partitioning) requirements, as explained next.

Search Strategies. We now discuss the search strategies that can be used over the FP-tree data structure, as well as the pruning techniques that can be applied on top of those.

– Breadth-first strategy (*expansion*): a) Locate, and remove from further consideration, individual nodes for items that are already in the dimension (according to the non-overlap criterion; for example, nodes, highlighted in gray in Figure 3); b) add one of the remaining available singleton items as a new itemset; we add these items one at a time.

– Depth-first strategy (*refinement*): a) For an itemset in the dimension, locate the correspondent paths in the FP-tree; b) Expand this itemset by adding one item at a time from the available children nodes of its paths.

However, the problem with the above strategies is that neither of them can lead to a good solution, when used independently: the breadth-first strategy may include many singleton items so that refinement (or expansion) of individual itemsets in a dimension is no longer possible; the depth-first strategy may restrict adding new itemsets to the dimension by expanding existing itemsets with their children items.

– Mixed strategy (*expansion + refinement*): Apply the expansion and refinement steps at every iteration. This is the strategy we use in this paper, and we discuss it in more detail in the following paragraphs (refer to Algorithm 1).

Pruning Strategies: We have already described the basic pruning strategy (non-overlap) based on our definition for dimensions. Our more advanced pruning strategy is based on the relationship between entropy and such characteristics as *coverage* and *co-occurrence*, as described in Lemmas 2-3. For each candidate dimension with entropy H, we are interested in obtaining refined dimensions, which do not exceed this value. Thus, we compute the corresponding thresholds for the minimal coverage C and maximal co-occurrence R (according to the above lemmata), and use them for pruning the itemsets which are added or refined.

Algorithm Description: We formulate our optimization problem in a greedy fashion, relying on a mixed candidate generation strategy and an iterative refinement of the candidate set. The complexity of this approach (almost) linearly depends on the size of the candidate set (as seen in Figure 5), which we use as a parameter. Another input of our algorithm is the FP-Tree, optionally containing only the most frequent items.

Algorithm 1: Mining Orthogonal Dimensions

Name : `findNewDimension`
Input : First-order dimensions $\Delta = \{\delta^k\}, k < i$,
Candidate dimensions $candidates = \{\}$,
FP-Tree, memoryBudget
Output: Optimal dimension δ^i_{out} of order i
repeat
  forall the *dimension* $\delta^i \in candidates.unprocessed$ **do**
    forall the *itemset* $\mathcal{I}_i \in \delta^i$ **do**
      forall the *items*
      $I_j \in children(\mathcal{I}_i),\ I_j \notin \delta^i, \Delta$ **do**
        if `validItemset(`$\mathcal{I}_i \cup \{I_j\} \mid \delta^i$`)` **then**
          //add one item to the current itemset
          $\delta^i_{temp} = \{\delta^i \mid \mathcal{I}_i = \mathcal{I}_i \cup \{I_j\}\}$;
          `checkOptimality(`$\delta^i_{temp} \mid \Delta$`)`;
          $candidates.temp.add(\delta^i_{temp})$;
        end
      end
    end
    forall the *items* $I_j \in \mathbb{I},\ I_j \notin \delta^i, \Delta$ **do**
      if `validItemset(`$\{I_j\} \mid \delta^i$`)` **then**
        //add one more item as an itemset
        $\delta^i_{temp} = \delta^i \cup \{I_j\}$;
        `checkOptimality(`$\delta^i_{temp} \mid \Delta$`)`;
        $candidates.temp.add(\delta^i_{temp})$;
      end
    end
  end
  //mark unprocessed as processed
  candidates += candidates.unprocessed;
  //newly generated become unprocessed
  candidates.unprocessed = candidates.temp;
  candidates.temp = {};
  //sort so that most optimal values are first
  candidates.sort();
  //remove candidates exceeding the allocated memory
  repeat
    candidates.remove(candidates.lastElement);
  until $candidates.size > memoryBudget$;
until $candidates.unprocessed.size > 0$;
return δ^i_{out} = *candidates.firstElement*;

refinement

expansion

This initial pruning does not affect the output (as long as the items forming the dimensions are preserved), but significantly reduces the complexity of the problem.

Our general approach starts with an empty set of dimensions, and uses Algorithm 1 to find each new dimension, resulting in the best optimality value when added to the set of previously selected dimensions; up to the specified number k.

The most essential part of this algorithm is the greedy dimension optimization procedure `findNewDimension`, which takes as a parameter a set of the first-order dimensions Δ, and an empty set of candidates, and after a finite number of iterations (the first loop) it converges to the single most optimal dimension, which is added to the Δ as the next one.

Algorithm 2: ItemSet pruning method `validItemset`. Entropy for a given dimension will improve only if the new itemset meets coverage and co-occurrence requirements computed using Lemmas 2-3 for the dimension's entropy.

Name : `validItemset`
Input : Dimension δ^i, itemset $\mathcal{I}_i$, FP-Tree
Output: true if itemset is valid, false otherwise
//use Lemma 2 to calculate min coverage
cov_{min} = `Lemma2(`$H(\delta^i)$`)`;
//use Lemma 3 to calculate max co-occurrence
$cooc_{max}$ = `Lemma3(`$H(\delta^i)$`)`;
return (`C(`$\mathcal{I}_i \cup \delta^i$`)` $> cov_{min}$ & `R(`$\mathcal{I}_i, \delta^i$`)` $\leq cooc_{max}$);

More specifically, Algorithm 1 iteratively refines dimensions in the candidate set (the empty initial set is refined only by expansion) and at each iteration performs sorting of candidates according to their optimality. The list of sorted dimensions is then being pruned according to the specified memory budget. By doing this operation, the algorithm ensures that at each step it would refine and check the optimality of only a short list of candidates, which is equal to the memory budget or lower. After all candidates in the list were refined, they are marked as processed (transferred to the main list), and the newly generated list of candidates becomes the next list of unprocessed candidates. The algorithm converges when there are no candidates left in the list, which were not refined. Then it outputs the topmost optimal candidate.

More insights on the algorithm can be obtained by examining Figure 4. In that figure, we depict detailed results of the refinement procedure for two specific domains, namely, "pyramid" and "art", from the `flickr` dataset. In both cases, we focus on the identification of the first dimension, and we depict for each iteration of the Algorithm 1 the size (number of itemsets) of the currently best dimension (bottom graphs), as well as the corresponding entropy value (top graphs).

We observe that for the "pyramid" domain the algorithm quickly increases the size of the dimension by adding more itemsets (seen as diagonal steps) and refining them (seen as horizontal steps), resulting in a significant initial improvement of the entropy of the dimension. Starting at iteration 6, the number of itemsets remains stable, though, the algorithm adds new items to them, leading to further improvements in entropy (which can be observed by the decrease of value on the topmost graph). In contrast, for the "art" domain the algorithm starts with a dimension of good quality (low entropy), which after a single refinement (from iteration 1 to 2) stays on the top of the list of candidates till iteration 9 (the value of entropy does not increase during this interval), while other candidates are being refined. Then, starting from the iteration 10, another candidate refined to a better quality takes its place and shows even better improvement in entropy. Finally, iterations converge and the best dimension is being identified.

We note that the final dimensions identified for the "pyramid" and "art" domains (after 17 and 15 iterations, respectively) are also the optimal single dimension decompositions for these domains.

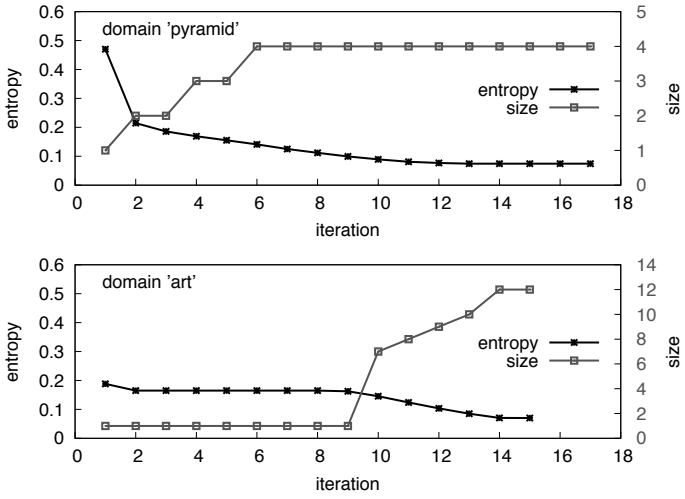

Fig. 4: Optimization stats of the 1^{st} dimension for "pyramid" and "art" (`flickr`).

V. Experimental evaluation

We evaluate our algorithm on two datasets[3] containing tag-annotated resources. The first dataset, extracted from `flickr`-a popular photos sharing website, contains 28 million tagsets (or transactions), obtained by taking annotations for all pictures that contained a specific domain tag, for 34 different domains. To remove noise, we allowed only unique tagsets for each user id. The second dataset contains tagsets from `del.icio.us`, a social bookmarking website. For this dataset, we selected annotations for URLs starting with specific domain names picked from `Yahoo! Directory`. Overall, the `del.icio.us` dataset contains 1.7 million tagsets over 150 domains. The number of unique tags in each of the datasets was about half a million.

For both datasets we performed a limited amount of additional cleaning by removing the domain term, numeric and navigational tags, as well as removing some language variability, based on a custom-built dictionary. No sophisticated preprocessing was applied, so some of the discovered dimensions in our experimental results still contain repetitions due to synonyms and misuse of tags.

A. Performance

In the first set of experiments, we report the execution time (Figure 5) and entropy of the best solution found (Figure 6), as a function of the maximum number of candidates considered by our algorithm. We vary the number of items between $8-20$, over the 150 domains of the `del.icio.us` dataset. In the graphs, we report the normalized values, averaged over all the 150 domains, as well as the standard deviation for these values (for most of the points standard deviation is too small and not visible). In order to make the results directly comparable to each other, we first normalize each series using the minimum (maximum) value of its regression line for the time (entropy) graph. Then, we compute the average normalized series, and its deviation.

In Figure 5, we report the averaged normalized execution times versus memory budget. We observe, that an increase in number of items results to an increase in complexity. Overall, the algorithm scales linearly with respect to the memory budget. When the number of items becomes large, the complexity is still determined by the memory budget (remember, that at each iteration the number of refinements is proportional to the size of the candidate set).

In Figure 6, we observe that for a small number of items, an increase in memory brings a considerably larger improvement in entropy, than for larger numbers of items. In the case of 8 items, the series drops until the entropy reaches its minimum for a maximum number of candidates of 32, which corresponds to the optimal solution. For larger number of items, the same effect is observed for a higher setting of the maximum number of candidates.

B. Parameters, Monotonicity, Synthetic Experiments

In order to evaluate various properties of our approach in a controlled environment, we constructed a synthetic dataset by generating itemsets for a number of dimensions closely resembling dimensions found in real datasets. These dimensions contained two to five itemsets of sizes up to three items, and we required exactly two dimensions to be present in each dataset. Following the construction of dimensions, we calculated the frequencies of singleton items by applying *Zipf's* law with a specified parameter z, $f(i^k) \sim 1/k^z$. This distribution was chosen because it is known to resemble word frequencies in real-world datasets. Moreover, it allows to produce frequencies that are close to the uniform (when z is small), or the exponential (when z is large) distributions. Evidently, the first case is more challenging for our problem.

In the subsequent step, we added a uniform noise of level ν to these frequencies, and harmonized them for the items belonging to each of the dimension's itemsets (to account for the observed rule that itemsets in dimensions usually have equally frequent items; for example, both tags in `{eiffel tower}` usually appear together).

Finally, the itemsets were generated by iteratively sampling the distribution of items with respect to the specified dimensions. In this process, we used *Gibbs* sampling first to select a dimension (independently from other dimensions) and then to select the itemset representing it (allowing for co-occurrence with a level of 0.25ν). The rest of items, not covered by any dimension, were distributed with respect to their frequencies.

In these experiments, we restricted the number of items to $n = 16$, which is equivalent to our minimum-support filtering on real datasets. Overall, we were generating 10 thousand itemsets for each dataset to ensure a smooth distribution according to our model.

We assessed the quality of the identified dimensions by comparing them to the dimensions used while generating the dataset. Our similarity measure was based on the *Hamming* distance d between two dimensions (represented as binary

[3]The url of the web page containing the code and the datasets is not available due to the double-blind review process.

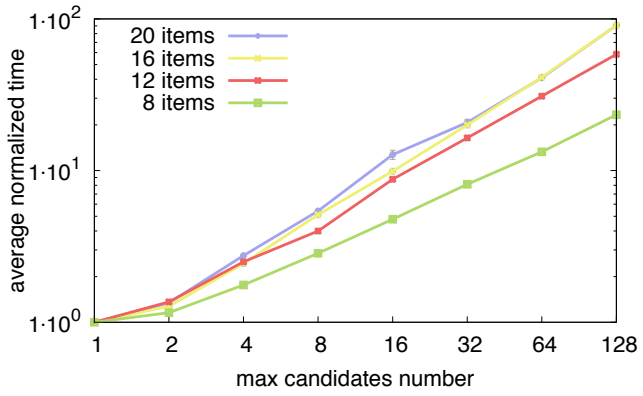

Fig. 5: Time vs memory budget.

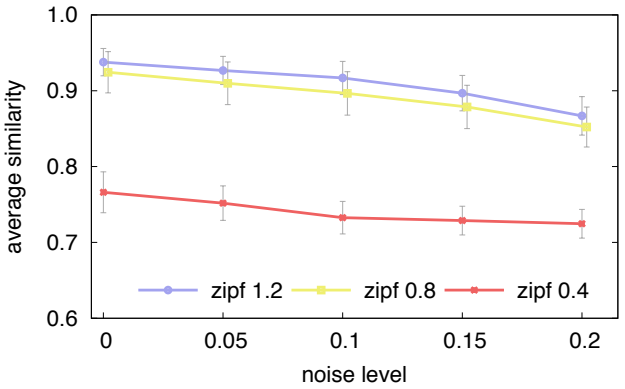

Fig. 7: Similarity to optimal dimensions versus noise.

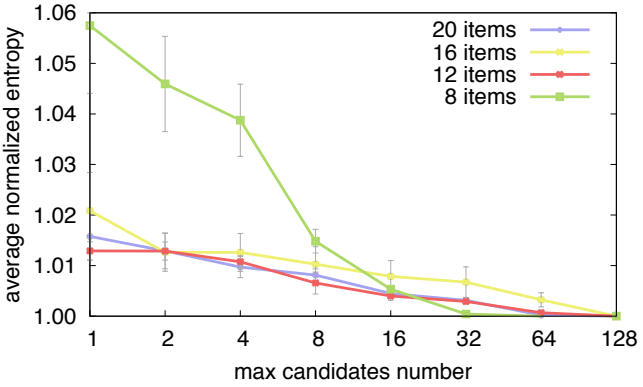

Fig. 6: Entropy vs memory budget.

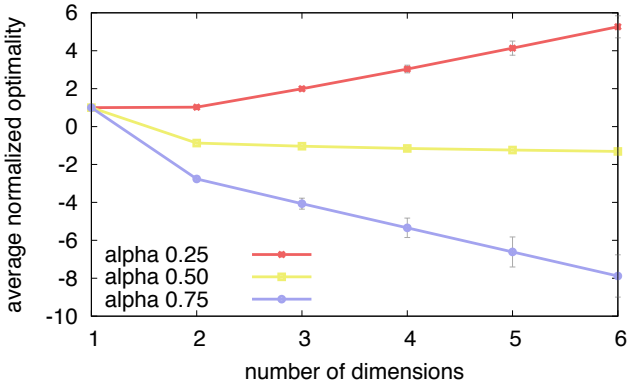

Fig. 8: $f(\Delta)$ dependency on the number of dimensions.

vectors) divided by the total number of items: $sim(\Delta;\Delta_0) = 1 - d(\Delta;\Delta_0)/n$. This measure takes values in the range $[0;1]$, with higher values indicating stronger similarity: a value of 1 means that the algorithm correctly identified the planted dimensions. We note that this measure does not account for the varying significance of items, which is not favoring our approach, since including low-support items in the dimensions represents a challenge, even without the additional noise.

The evaluation of quality against noise for different parameters z is shown in Figure 7. In gray lines we plot the 0.95 confidence intervals for average values.

We can see that regardless of the noise added, our method is able to reconstruct almost perfectly the optimal dimensions for a wide range of distributions. As expected, the similarity between the identified and the optimal dimensions decreases on average with growing noise, and is significantly lower for smaller parameters z (more uniform items distribution).

In Figure 8 we evaluate the monotonicity of $f(\Delta)$ over the number of dimensions k, for different values of α parameter. It is clearly visible that for small values of α optimality gets higher (worse), while for large values every new dimension improves optimality (albeit, not the quality of extracted dimensions). For our experiments we chose $\alpha = 0.5$, since it provides a good balance between orthogonality and interestingness, and allows to rely on Theorem 2 (controlling the decomposition) for a wide range of data distributions.

C. Qualitative Results

We now report results on a qualitative evaluation of the proposed approach. We ran our algorithms on a set of different domains from `flickr` and `del.icio.us` datasets: "eiffel tower", "art", "hollywood", "pyramid", and "spain" for `flickr`; "nytimes.com", "lifehacker.com", "dpreview.com", "apple.com", "microsoft.com", and "ixbt.com" for `del.icio.us`. We use a 3% minimum support threshold on items for all domains. The results of this experiment are summarized in Table II, where for each domain we report the top dimensions identified by our algorithm. We should note, that because of the fixed minimum support threshold, for some of the domains all available items are allocated to the first few dimensions, thus resulting in the varying number of dimensions being identified. In every case, we limit this number to the 3 top dimensions.

The dimensions reported by our algorithm are successfully describing the different concepts under each domain. For example, under the "eiffel tower" domain, we have as first dimension the *Eiffel Tower in Paris and Las Vegas*[4], as second dimension *holidays in Paris*, and as third dimension *architecture*, all of which are different concepts related to "eiffel tower". Similarly, the "dpreview.com" domain in the del.icio.us dataset is described by the concepts of *photographic camera reviews*, *digital [photography]*, and *shopping*.

[4]The city of Las Vegas (NV, USA) hosts a replica of the Eiffel Tower.

δ^i	collection of itemsets for δ^i (flickr)
	domain **"jaguar"**
1	{automobile}, {zoo}
2	{etype}, {auto}
	domain **"eiffel tower"**
1	{paris france europe tower}, {lasvegas}
2	{night seine}, {holiday travel}
3	{architecture}
	domain **"pyramid"**
1	{egypt giza cairo sphinx}, {louvre paris museum glass}, {mexico maya ruins}, {sanfrancisco transamerica}
2	{france sky}, {travel teotihuacan}
3	{architecture night}, {chichenitza}
	domain **"hollywood"**
1	{losangeles california sign}, {star film actor}
2	{us universalstudios}, {hollywoodboulevard night}
3	{theatre party sunset}, {canon street}
	domain **"art"**
1	{painting drawing}, {graffiti streetart}, {sculpture museum}, {newyork}, {color}, {photo}, {street}
	domain **"spain"**
1	{barcelona catalonia}, {madrid europe}, {andalusia granada}, {seville}, {valencia}, {holiday travel}
2	{architecture}

δ^i	collection of itemsets for δ^i (del.icio.us)
	domain **"nytimes.com"**
1	{news politics}, {food health}, {science}, {article}, {business}, {technology}
	domain **"dpreview.com"**
1	{photo camera review}, {dslr}
2	{digital}
3	{shopping}
	domain **"lifehacker.com"**
1	{howto lifehacks tips},{software windows tools freeware}
2	{firefox internet}, {linux utilities}, {email extensions}, {mp3 download}, {organization toread}, {photography}
	domain **"apple.com"**
1	{mac osx software},{ipod itunes music},{video quicktime}, {movies trailers},{iphone},{podcast podcasting},{technology}
2	{macosx howto}
	domain **"microsoft.com"**
1	{windows software tools}, {.net programming}
2	{security xp}
3	{utilities}
	domain **"ixbt.com"**
1	{hardware software news computers russian}, {photo photography}
2	{article}
3	{reviews}

TABLE II: Top dimensions for different domains in flickr and del.icio.us.

The results of this experiment demonstrate that our approach can effectively identify the diverse concepts related to some domain, in an automatic fashion. Finally, we observe that our algorithm provides meaningful results, even when operating on noisy datasets, such as flickr and del.icio.us, which contain a large number of non-useful tags.

VI. Conclusions and Future Work

Motivated by applications on repositories of annotated resources in the collaborative tagging domain, we introduce the problem of diverse dimension decomposition in transactional databases. In particular, we adopt an information-theoretic perspective on the problem, relying on entropy for defining a single objective function that simultaneously captures constraints on coverage, exclusivity and orthogonality.

We present an approximate greedy method for extracting diverse dimensions, that exploits the FP-tree representation of the input transactional dataset and clever pruning techniques. Our experiments on datasets of tagged resources from flickr and del.icio.us confirm effectiveness and efficiency of our proposal. The assessment on synthetic and artificially noisy data confirms that our method is able to reconstruct the "true" dimensions, and it withstands noise.

In our future investigations, we plan to study and compare both empirically and theoretically, different probability measures, which not only control "the look" of identified dimensions, but can even lead to their different semantics. This becomes especially relevant in the context of our optimization problem, where a probability space may well determine the overall shape of the optimization objective and its convergence. We also plan to have a user study for evaluating the discovered dimensions in different domains. A possibility is also that of developing a vertical application exploiting our method for mining diverse dimensions in order to detect, in unsupervised and automatic fashion, collection of web sites with diverse content from del.icio.us.

References

[1] A. J. Knobbe and E. K. Y. Ho, "Maximally informative k-itemsets and their efficient discovery," in *KDD*, T. Eliassi-Rad, L. H. Ungar, M. Craven, and D. Gunopulos, Eds. ACM, 2006, pp. 237–244.

[2] ——, "Pattern teams," in *PKDD*, ser. Lecture Notes in Computer Science, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds., vol. 4213. Springer, 2006, pp. 577–584.

[3] N. Tatti, "Probably the best itemsets," in *KDD*, B. Rao, B. Krishnapuram, A. Tomkins, and Q. Yang, Eds. ACM, 2010, pp. 293–302.

[4] J. V. Michael Mampaey, Nikolaj Tatti, "Tell me what i need to know: succinctly summarizing data with itemsets," in *KDD*, 2011.

[5] H. Heikinheimo, E. Hinkkanen, H. Mannila, T. Mielikäinen, and J. K. Seppänen, "Finding low-entropy sets and trees from binary data," in *KDD*, 2007.

[6] C. Zhang and F. Masseglia, "Discovering highly informative feature sets from data streams," in *DEXA*, 2010.

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

[8] B. Sigurbjörnsson and R. van Zwol, "Flickr tag recommendation based on collective knowledge," in *WWW*, 2008.

[9] R. van Zwol, B. Sigurbjörnsson, R. Adapala, L. G. Pueyo, A. Katiyar, K. Kurapati, M. Muralidharan, S. Muthu, V. Murdock, P. Ng, A. Ramani, A. Sahai, S. T. Sathish, H. Vasudev, and U. Vuyyuru, "Faceted exploration of image search results," in *WWW*, 2010.

[10] M. Grahl, A. Hotho, and G. Stumme, "Conceptual clustering of social bookmarking sites," in *LWA 2007: Lernen - Wissen - Adaption*, 2007.

[11] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina, "Clustering the tagged web," in *WSDM 2009: Proc. of the 2nd ACM Int. Conference on Web Search and Data Mining*, 2009.

[12] M. van Leeuwen, F. Bonchi, B. Sigurbjörnsson, and A. Siebes, "Compressing tags to find interesting media groups," in *CIKM*, 2009.

[13] F. Bonchi, C. Castillo, D. Donato, and A. Gionis, "Topical query decomposition," in *KDD*, 2008.

[14] B. Carterette and P. Chandar, "Probabilistic models of ranking novel documents for faceted topic retrieval," in *CIKM*, 2009.

[15] R. L. Santos, C. Macdonald, and I. Ounis, "Exploiting query reformulations for web search result diversification," in *WWW*, 2010.

[16] G. Capannini, F. M. Nardini, R. Perego, and F. Silvestri, "Efficient diversification of web search results," *PVLDB*, vol. 4, no. 7, pp. 451–459, 2011.

[17] F. Korn, A. Labrinidis, Y. Kotidis, and C. Faloutsos, "Quantifiable data mining using ratio rules," *VLDB J.*, vol. 8, no. 3-4, pp. 254–266, 2000.

[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, October 1996.

[19] F. Verhein and S. Chawla, "Geometrically inspired itemset mining," in *Proc. ICDM 2006*, 2006, pp. 655–666.

[20] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New york, NY, USA: Wiley & Sons, 1991.