A framework for mining interesting pattern sets

Tijl De Bie
University of Bristol, Intelligent Systems Laboratory
Merchant Venturers Building, Bristol, BS8 1UB, UK
tijl.debie@gmail.com

Kleanthis-Nikolaos Kontonasios
University of Bristol, Intelligent Systems Laboratory
Merchant Venturers Building, Bristol, BS8 1UB, UK
kk8232@bristol.ac.uk

Eirini Spyropoulou
University of Bristol, Intelligent Systems Laboratory
Merchant Venturers Building, Bristol, BS8 1UB, UK
enxes@bristol.ac.uk

ABSTRACT

This paper suggests a framework for mining subjectively interesting pattern sets that is based on two components: (1) the encoding of prior information in a model for the data miner's state of mind; (2) the search for a pattern set that is maximally informative while efficient to convey to the data miner.

We illustrate the framework with an instantiation for tile patterns in binary databases where prior information on the row and column marginals is available. This approach implements step (1) above by constructing the MaxEnt model with respect to the prior information [2, 3], and step (2) by relying on concepts from information and coding theory.

We provide a brief overview of a number of possible extensions and future research challenges, including a key challenge related to the design of empirical evaluations for subjective interestingness measures.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications—*Data mining*; I.5.1 [**Pattern recognition**]: Models—*Statistical*

Keywords

Subjective interestingness measures, pattern set mining, prior information, maximum entropy.

1. BACKGROUND

Motivation.

Since the introduction of the Apriori algorithm significant progress has been made in developing increasingly efficient and sophisticated frequent itemset mining algorithms. Today, we have arguably reached the point where progress in this respect has become incremental. Perhaps to a lesser extent the same holds for other pattern mining techniques.

UP'10, July 25th, 2010, Washington, DC, USA.

Instead, in real-life applications of pattern mining new challenges have surfaced (e.g. [15]). Most of these are centered around the observed discrepancy between what is intuitively interesting and the existing objective proxies, such as the frequency of a pattern. Indeed, a strong consensus is growing that finding better objective formalizations of what is intuitively or subjectively interesting is critical for the success of the field. A second problem is that the set of all patterns deemed interesting by any specific interestingness measure contains too many patterns to be convenient for human consumption, many of which are highly redundant.

Matching these two problems, two research avenues are being pursued by the research community. The first problem is addressed by the search for better interestingness measures, even if that comes at an added computational cost when compared to monotonic or anti-monotonic measures (see [8] for an overview). The second problem is addressed by searching for interesting *pattern sets*, rather than for sets of interesting patterns (see e.g. [4, 1, 7, 5, 6]). These lines of research are by no means independent, and many methods attempt to address them simultaneously.

Despite significant recent progress on both these fronts, we believe the challenges cannot be fully met by proposing yet more objective measures with new properties. Indeed, counting only probabilistically inspired objective interestingness measures, in the survey paper [8] the authors list 38 of them. It is clear that expanding this zoo of interestingness measures even further would not increase transparency of the research area. Instead, the recent advances need to be embedded in a flexible and interactive approach.

In this paper, we discuss a framework that tries to achieve this, which we believe addresses both problems discussed above. It relies on formalizing the prior information of the data miner, and contrasting the data with this formal representation of the state of mind of the data miner. In this way, a pattern set can be found that is subjectively interesting, and data mining algorithms become intelligent communication interfaces between the data and the data miner.

Below we mostly consider patterns (such as itemset and tile patterns) in binary databases. However, we wish to stress that our framework is more widely applicable, and we will therefore introduce it in general terms.

Subjective interestingness measures.

The distinction between subjective and objective interestingness measures was first made in [21, 18, 19], and adopted in the survey paper [8]. Subjective interestingness measures as they conceive them are interestingness measures that do

not only depend on properties of the pattern, but also on the class of users of the data mining algorithm. They should quantify at least one of two properties: unexpectedness, or actionability. Both are clearly strongly dependent on the data miner's prior information or goals.

The first attempt at designing a subjective interestingness measure quantifying unexpectedness was made by [21]. They made use of a so-called belief system, which consists of a set of rules with associated degrees of belief, representing what the data miner knows about the data. Then, patterns are deemed more interesting if they strongly affect these beliefs in a Bayesian sense. The approach had some drawbacks, the most important of which is probably that interactions between these rules were hard to control: patterns implied by combinations of rules from the belief system would be deemed unexpected by their system, if they are not implied by any single rule by itself. While the practical implication of this work is perhaps limited for these reasons, its importance as a conceptual breakthrough is hard to overestimate.

Still, since this work, very few other subjective interestingness measures have been proposed (see [8] for an overview). The most promising one is probably from [12], where the authors suggest to model transactions in a binary database using Graphical Models, and use this model to compute the expected support of itemsets. Itemsets are then deemed more interesting if their support deviates more strongly (in absolute sense) from the expected support given this model. The Graphical Model could be designed such that it reflects the prior information of a data miner, although it is less clear how to do this in practice (where data miners may have limited expertise in Graphical Models). Furthermore, it assumes that transactions are independent and identically distributed, often false in practice. And lastly, the absolute difference between expected and observed support may not be the best measure of unexpectedness.

Some other recent approaches claim to take into account prior information to quantify interestingness, such as [5, 6, 9, 17, 11]. All of these are based on hypothesis testing to formalize interestingness, where the null hypothesis is designed to represent the prior information of the data miner. Those based on randomization approaches [9, 17, 11] are computationally demanding but are also more flexible. Still, they are limited to specific types of prior information, such as on row and column marginals [9, 17], and more recently also cluster structure and the frequency of given itemsets [11].

The framework proposed in this paper aims to be flexible, as well as realistically useful in real-life data mining settings.

2. A GENERAL FRAMEWORK

Below we will first introduce the rationale of our framework. Then we will detail the two basic components: modeling the prior information, and searching for pattern sets that are interesting when contrasted to this prior information.

2.1 Shifting the focus: from the data to the miner

In designing objective interestingness measures, a data mining researcher tries to enter the mind of an imagined practitioner, and attempts to rationalize what may be intuitively of interest to this practitioner. This approach has born fruit in two respects. First, it has helped in understanding which types of interestingness measures are amenable to efficient algorithms. Second, for specific applications, special-purpose interestingness measures are often desirable.

However, the strategy of entering a specific practitioner's mind inevitably falls short of the design of measures that can be applied in a wide range of circumstances, by a wide range of practitioners. In the design of flexible subjective measures, we believe it essential to consider the data mining practitioner as the object of study, no less than the data itself. Such a Copernican revolution, shifting the focus from the data to the data mining process (Fig. 1), is likely to be necessary if we intend to capture subjectivity. Indeed, this can only be achieved if the algorithm is aware of what the data miner wants or does not want to learn about the data.

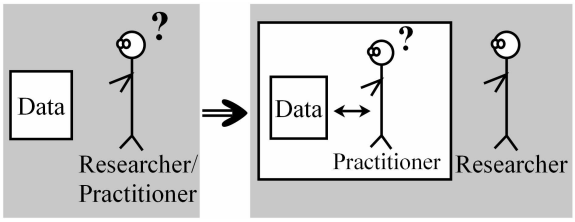

Figure 1: The researcher's point of view when designing objective interestingness measures (left, where he coincides with the practitioner) and subjective interestingness measures (right).

In this paper, we aim to formalize the data mining process as we envisage it with the above considerations in mind. We do this by explicitly modeling the data miner's prior information and hence what is *not* interesting to him. What is of interest to the data miner is then what contrasts with this prior information. Hence, in the actual mining step, a set of patterns is sought that is maximally interesting given the prior information. We believe this approach safeguards the exploratory nature of pattern mining methods better than a more limiting approach that directly specifies what is interesting.

Thus, there are two essential components in our framework: the modeling of the data miner's prior information, and the subsequent search for a pattern set of which the occurrence in the data contrasts with this model of prior information. Below we will fill in this framework more concretely, detailing both these aspects individually.

Throughout this Section, we will complement the theory with an example for the case of binary databases represented by the binary matrix $\mathbf{D} \in \{0,1\}^{m \times n}$, where the prior information is on row and column marginals $\sum_j \mathbf{D}(i,j)$ and $\sum_i \mathbf{D}(i,j)$, and where we are searching for interesting tiles defined by a subset of rows $I \subseteq \{1, \cdots, m\}$ and a subset of columns $J \subseteq \{1, \cdots, n\}$ such that $\mathbf{D}(i,j) = 1$ for all $i \in I$ and $j \in J$. This example is given for concreteness only, and in Sec. 3.3 we aim to make it clear that it can be applied much more generally.

In this paper, random variables will be underlined (e.g. $\underline{\mathbf{D}}$), while deterministic samples of these random variables are not underlined (e.g. $\mathbf{D}$ is a specific instance of the data).

2.2 Formalizing prior information in a probabilistic model

As suggested in [2, 3], we choose to formalize the prior in-

formation in a probability distribution P defined over the data space $\mathcal{D}$. This can be done by setting up a probabilistic model for the data $\mathbf{D}$ that satisfies certain constraints imposed by the prior information. Note that typically, the data $\mathbf{D}$ itself is composed of a set of variables: $\mathbf{D} = \{\mathbf{d}_k, k = 1 : n\}$ with typically $\mathbf{d}_k \in \{0, 1\}$ or $\mathbf{d}_k \in \mathbb{N}$, or $\mathbf{d}_k \in \mathbb{R}$ (e.g. the entries in a database).

The type of prior information we will consider is in the form of expectations about certain functions f_i (further called constraints functions) of the data $\mathbf{D}$:

$$E_P\{f_i(\underline{\mathbf{D}})\} \quad = \quad c_i.$$

EXAMPLE 2.1. *As an example for a binary database $\mathbf{D}$, we will consider two classes of constraint functions, computing the row and the column marginals of the database: $f_i^r(\mathbf{D}) \triangleq \sum_j \mathbf{D}(i,j)$ and $f_j^c \triangleq \sum_i \mathbf{D}(i,j)$. I.e., the constraints are:*

$$E_P\left\{\sum_j \underline{\mathbf{D}}(i,j)\right\} \quad = \quad c_i^r,$$

$$E_P\left\{\sum_i \underline{\mathbf{D}}(i,j)\right\} \quad = \quad c_j^c,$$

where c_i^r and c_j^c are the required expected row and column marginals. This means that we assume that the data miner has certain expectations on each of the row and column marginals as prior information.

Using constraints on the expectations of certain properties of the data, as quantified by the functions f_i, is a flexible way of encoding prior information. We will give more examples in Sec. 3.3.[1]

As in practical settings prior information will not be so rich as to uniquely determine the distribution, an inductive bias needs to be chosen. For various reason discussed in [2, 3] and references therein, it makes sense to choose the distribution of maximum entropy among all those that satisfy the constraints:

$$\begin{aligned} \max_P \quad & -E_P\{\log(P(\underline{\mathbf{D}}))\}, \\ \text{s.t.} \quad & E_P\{f_i(\underline{\mathbf{D}})\} = c_i, \\ & P(\mathbf{D}) \geq 0 \ \forall \mathbf{D} \in \mathcal{D}, \\ & \sum_{\mathbf{D}\in\mathcal{D}} P(\mathbf{D}) = 1. \end{aligned}$$

We refer to the resulting problem as the MaxEnt model.

It is well-known (and easy to prove using Lagrange duality theory) that the solution of the maximum entropy optimization problem takes the form of an exponential family distribution (see e.g. [23]):

$$P(\mathbf{D}) = \frac{1}{Z(\boldsymbol{\lambda})} \exp\left(\sum_i \lambda_i f_i(\mathbf{D})\right),$$

where $\boldsymbol{\lambda}$ denotes a vector containing all λ_i and $Z(\boldsymbol{\lambda}) = \sum_{\mathbf{D}\in\mathcal{D}} \exp\left(\sum_i \lambda_i f_i(\mathbf{D})\right)$ is known as the partition function and ensures normalization. The values of the Lagrange multipliers λ_i can be found by solving the dual optimization problem, which is formally identical to minimizing the negative log-likelihood of data $\mathbf{D}$ that satisfies the constraints $f_i(\mathbf{D}) = c_i$ exactly. Mathematically, this optimization problem is written as:

$$\min_{\boldsymbol{\lambda}} \quad \log(Z(\boldsymbol{\lambda})) - \sum_i \lambda_i c_i.$$

It is worth emphasizing that the exponential family of distributions encompasses most widely used distributions, including the Bernoulli, binomial, Poisson, and Gaussian distributions and many others.

EXAMPLE 2.2. *The MaxEnt model for prior information on row and column marginals on a binary database $\mathbf{D}$ as defined in Ex. 2.1 is given by an exponential family distribution that can be rewritten as a product distribution of Bernoulli random variables, one for each database entry:*

$$\begin{aligned} P(\mathbf{D}) \quad &= \quad \prod P_{ij}(\mathbf{D}(i,j)), \\ P_{ij}(\mathbf{D}(i,j)) \quad &= \quad \begin{cases} \frac{\exp(\lambda_i^r+\lambda_j^c)}{1+\exp(\lambda_i^r+\lambda_j^c)} & \text{if} \quad \mathbf{D}(i,j) = 1, \\ \frac{1}{1+\exp(\lambda_i^r+\lambda_j^c)} & \text{if} \quad \mathbf{D}(i,j) = 0. \end{cases} \end{aligned}$$

Note that although the random variables for the different database entries are independent, their distributions are related by the parameters λ_i^r for the rows and λ_j^c for the columns. These are obtained by solving the optimization problem:

$$\min_{\boldsymbol{\lambda}^r,\boldsymbol{\lambda}^c} \quad \sum_{i,j} \log(1 + \exp(\lambda_i^r + \lambda_j^c)) - \sum_i \lambda_i^r c_i^r - \sum_j \lambda_j^c c_j^c.$$

It is shown in [3] that this problem can be solved remarkably efficiently even for very large databases.

The duality relation between the Maximum Entropy and Maximum Likelihood problems, with the exponential family as a hinge between them, is well known in mathematical statistics. Also in the Graphical Models literature, it has been studied for the special case where the constraint functions f_i are so-called potential functions, i.e. (often indicator) functions that pertain to a typically small subset of the variables $\mathbf{d}_k$ making up the data $\mathbf{D}$. (See [23] for an overview.) In this context, however, we do not constrain ourselves to this situation. Allowing more general functions leads to models such as in Ex. 2.2 where the graphical model representation would be trivial (all random variables $\underline{\mathbf{d}}_i$ making up the data $\underline{\mathbf{D}}$ are independent), but where the distributions of these random variables are related by sharing certain parameters in a non-trivial way.

2.3 Information theory to quantify subjective interestingness

Given a probabilistic model capturing the prior information about the data, we can now attempt to quantify the interestingness to the data miner of a given pattern in the data. Taking account of the prior information, such quantification will be inherently subjective.

Formalizing patterns.

Before we can proceed, we need to define formally what we mean by a pattern.

DEFINITION 2.3. *Let $\pi : \mathcal{D} \to \mathbb{R}$ be a function that we call a* pattern function *and that is an element from the* pattern

[1] Note that hard constraints of the form $g(\mathbf{D}) = c$ can also be imposed in this way, by using an indicator function $f_i(\mathbf{D}) \triangleq \delta(g(\mathbf{D}) = c)$. Then, $g(\mathbf{D}) = c$ with probability one if $E_P\{f_i(\underline{\mathbf{D}})\} = 1$ is imposed as a constraint.

space Π, *i.e.* $\pi \in \Pi$. *A* pattern *in the data* $\mathbf{D}$ *is defined as an equality of the form:*

$$\pi(\mathbf{D}) = \hat{\pi}.$$

We call $\hat{\pi} \in \mathbb{R}$ *the* pattern strength.

For example, in the context of frequent itemset mining, the pattern functions π considered are functions that evaluate as the frequency of an itemset. The set Π of all such functions is determined by the collection of all frequent itemsets. This definition is different from the standard definition in frequent pattern mining: the pattern for us is not the recurring element, but the fact that the element recurs a certain number of times in the data (as expressed by the equality $\pi(\mathbf{D}) = \hat{\pi}$). Let us give another example:

EXAMPLE 2.4. *We define the pattern functions as indicator functions for the presence of a tile, and denote them as* $\pi_{I,J}$ *for a tile with rows* I *and columns* J*. I.e.:*

$$\pi_{I,J}(\mathbf{D}) \quad = \quad \begin{cases} 1 & \text{if } \forall i \in I, j \in J : \mathbf{D}(i,j) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Hence, the pattern strength for a pattern function $\pi_{I,J}$ *is equal to* 1 *if the tile* (I, J) *is present in the data, and* 0 *otherwise.*

We believe the risk associated with using a non-standard definition for a pattern is outweighed by an important benefit: it allows us to deal with a much broader class of problems than just frequent pattern mining. As a result, the framework described in this paper can be transferred easily to other types of patterns, such as tile patterns (see Ex. 2.4), clustering patterns, classification patterns, etc.

The self-information of a pattern.

Before defining the interestingness of a pattern $\pi(\mathbf{D}) = \hat{\pi}$, we need to quantify the amount of information in the pattern as perceived by the data miner. This is adequately formalized by the Shannon self-information of the pattern with respect to distribution P that formalized the prior information, defined as the negative log-probability of seeing the observed pattern strength. Formally:

DEFINITION 2.5. *The* self-information *of a pattern* $\pi(\mathbf{D}) = \hat{\pi}$ *is defined as:*

$$I(\pi, \hat{\pi}) = -\log\left(Pr(\pi(\underline{\mathbf{D}}) = \hat{\pi})\right).$$

It is equal to the number of bits required to encode the pattern strength $\hat{\pi}$ *of this pattern under a Shannon optimal code with respect to the MaxEnt distribution* P *for* $\underline{\mathbf{D}}$*.*

The self-information is known to be the code length of a random variable (here the pattern strength $\hat{\pi}$ in the data $\mathbf{D}$) under a Shannon-optimal code with respect to the distribution P. Hence, this quantity also has an interpretation in terms of description length.

EXAMPLE 2.6. *Under the MaxEnt model from Ex. 2.2 and with the tile pattern functions* $\pi_{I,J}$ *from Ex. 2.4, the self-information of a pattern* $\pi_{I,J}(\mathbf{D}) = 1$ *is defined as:*

$$\begin{aligned} I(\pi_{I,J}, 1) &= -\log\left(\prod_{i\in I, j\in J} P_{ij}(1)\right), \\ &= -\sum_{i\in I, j\in J} \log\left(\frac{\exp(\lambda_i^r + \lambda_j^c)}{1 + \exp(\lambda_i^r + \lambda_j^c)}\right). \end{aligned}$$

With the MaxEnt model as a representation of the practitioner's uncertainty about the data, the self-information is equal to the information conveyed to the data miner when he is informed about the fact that a certain tile is present rather than not present in the data $\mathbf{D}$*. It is equal to the required code length if the presence of the tile is described using a Shannon optimal code with respect to the MaxEnt distribution. The self-information would be larger if, given the prior information, the tile is less likely to be present.*

The description length of a pattern.

The self-information $I(\pi, \hat{\pi})$ is the amount of information transmitted to the data miner if he is made aware of the presence of the pattern $\pi(\mathbf{D}) = \hat{\pi}$. The question now arises what the true cost is of communicating this information to the data miner. Can this be done more efficiently than with a Shannon-optimal code with respect to the MaxEnt model? It is clear that this is only possible if there are patterns in the data that are not to be expected given the MaxEnt model of the prior information, and we argue these are precisely the ones the data miner is interested in.

The true cost of conveying a pattern can be quantified by establishing a coding scheme to encode pattern functions $\pi \in \Pi$, and similarly for the pattern strengths $\hat{\pi}$. Then the cost can be defined as the description length $D(\pi, \hat{\pi})$ of the pattern $\pi(\mathbf{D}) = \hat{\pi}$ in this coding scheme. The coding scheme should be chosen so that it reflects the perceived complexity of a pattern. This approach based on coding lengths is convenient, as it will allow us to compare like with like when contrasting this cost with the self-information.

A difficulty with this approach is the design of a code, which can be cumbersome. As a shortcut, however, one could just specify the code lengths directly. When doing so, they must be such that, in principle, a uniquely decipherable code exists with these code word lengths. This means that the code words must satisfy Kraft's inequality [14].

A set of code lengths satisfying Kraft's inequality can be designed conveniently by first defining a distribution Q over the set of patterns that may need to be encoded, and choosing the code lengths of all patterns equal to their negative log-probability under that distribution. (Note that the obtained code lengths may not be integers, but they can still be achieved in the limit if a large number of these patterns are to be encoded using a Shannon-optimal coding under that chosen distribution.)

It should be stressed that the distribution Q and the description length are unrelated to the prior information the data miner holds about the data, and they are also unrelated to any stochastic process from which the data may have been sampled. It is no more than a mathematical construct to help the data miner in quantifying how hard it is for him to grasp a given pattern.

DEFINITION 2.7. *The* description length $D(\pi, \hat{\pi})$ *of a pattern* $\pi(\mathbf{D}) = \hat{\pi}$ *is given by its description length in a code chosen by the data miner, capturing the complexity of patterns as he perceives it.*

It is convenient to compute the description length indirectly, by first specifying a distribution Q *over the space of possible pairs* $(\pi, \hat{\pi})$*. Then the description length of a pattern* $\pi(\mathbf{D}) = \hat{\pi}$*, is given by the negative log-probability of* $(\pi, \hat{\pi})$ *under distribution* Q*:*

$$D(\pi, \hat{\pi}) = -\log(Q(\pi, \hat{\pi})).$$

EXAMPLE 2.8. *To encode a tile (I, J), for each row i and for each column j we need to specify whether or not $i \in I$ and $j \in J$. We will design a coding scheme for tiles using the approach above, i.e. by first defining a distribution Q.*

Let us assume that a data miner finds a tile easier to grasp if it contains less rows and less columns, with no distinction made between different rows or columns. Then, the distribution Q could be defined as:

$$\begin{aligned} Q(\pi_{I,J}, 1) &= p^{|I|+|J|}(1-p)^{m+n-|I|-|J|}q, \\ Q(\pi_{I,J}, 0) &= p^{|I|+|J|}(1-p)^{m+n-|I|-|J|}(1-q), \end{aligned}$$

where the $0 \leq p, q \leq 1$, p is the probability that any row or column belongs to a tile, and q is the probability of a pattern strength equal to 1. After some calculations, this means that the description length of a tile pattern with $\hat{\pi} = 1$ is equal to:

$$D(\pi_{I,J}, 1) = C + (|I| + |J|)D,$$

where $C = -(m+n)\log(1-p) - \log(q)$ and $D = \log\left(\frac{1-p}{p}\right)$.

For $p > 0.5$, it holds that $D > 0$, and the description length increases linearly with the circumference of the tile. The parameter p allows the data miner to zoom in to small tiles (smaller p), or zoom out to larger tiles (larger p). In the experiments below, we chose p equal to the density of the database, i.e. equal to the probability that a randomly selected row contains a 1 in a randomly selected column. We further chose q equal to 1, such that only pattern strengths equal to 1 would be considered.

The interestingness of a pattern.

The interestingness of a pattern can now be determined by comparing the description length of the pattern with its information content. In particular, we suggest to define the interestingness of a pattern as follows:

DEFINITION 2.9. *The* interestingness *of a pattern $\pi(\mathbf{D}) = \hat{\pi}$ is defined as the ratio of the self-information over the description length:*

$$interestingness(\pi, \hat{\pi}) = \frac{I(\pi, \hat{\pi})}{D(\pi, \hat{\pi})}.$$

Intuitively speaking, this quantifies the compression ratio of the information in the pattern by reporting it as a pattern in the code representing the data miner's intuition of simplicity.

EXAMPLE 2.10. *For the tile example, the interestingness measure will be larger if it covers as many (improbable) entries as possible (i.e. if it has a large surface), while having a circumference that is as small as possible.*

Interesting pattern sets.

In practice, a data miner will rarely be satisfied with just the single most interesting pattern. It is likely that the data miner has a certain finite processing capability, determining an upper bound u on the total description length of all patterns reported. Given this upper bound, the data miner would like to receive as much information as possible when contrasted with his prior information. This information can be captured adequately by the self-information of the pattern set, defined as:

DEFINITION 2.11. *The* self-information *of a pattern set is defined as the negative log-probability that these patterns are present in the data under the MaxEnt model. Formally, with pattern functions $\pi \in \Pi_s \subseteq \Pi$ and associated pattern strengths $\hat{\pi} = \pi(\mathbf{D})$ observed in the data $\mathbf{D}$:*

$$I\left(\{(\pi, \hat{\pi}), \pi \in \Pi_s\}\right) = -\log\left(Pr(\pi(\underline{\mathbf{D}}) = \hat{\pi}, \forall \pi \in \Pi_s)\right)$$

with respect to the MaxEnt distribution for $\underline{\mathbf{D}}$.

To maximally satisfy the data miner, the data mining algorithm should thus solve the following optimization problem:

$$\begin{aligned} \max_{\Pi_s \subseteq \Pi} \quad & I\left(\{(\pi, \hat{\pi}), \pi \in \Pi_s\}\right), \\ \text{s.t.} \quad & \sum_{\pi \in \Pi_s} D(\pi, \hat{\pi}) \leq u. \end{aligned}$$

The most interesting pattern set subject to the imposed constraint on the description length is then defined by the optimal set of pattern functions Π_s.

This optimization problem is unfortunately a combinatorial one, and it is hard to solve in general. However, in some cases it may be easy to solve it or at least to solve it approximately. Let us illustrate this with an example.

EXAMPLE 2.12. *The self-information of a pattern set with tile-patterns $\pi_{I,J} = 1$ with $\pi_{I,J} \in \Pi_s$ is given by:*

$$\begin{aligned} & I\left(\{(\pi_{I,J}, 1), \pi_{I,J} \in \Pi_s\}\right) \\ = & -\log\left(\prod_{i,j:\exists \pi_{I,J} \in \Pi_s : i \in I \& j \in J} P_{ij}(1)\right), \\ = & -\sum_{i,j:\exists \pi_{I,J} \in \Pi_s : i \in I \& j \in J} \log\left(\frac{\exp(\lambda_i^r + \lambda_j^c)}{1 + \exp(\lambda_i^r + \lambda_j^c)}\right). \end{aligned}$$

Hence, the problem can be phrased as follows. Given is the set of entries in the database and a collection of subsets of this set as covered by the tiles. Each entry has a certain weight ($-\log(P_{ij}(\mathbf{D}(i,j))$), and each subset has a certain cost ($D(\pi, \hat{\pi})$). The pattern set mining task can then be formulated as the search for a collection of subsets maximizing the sum of the weights of the entries in its union, subject to an upper bound on the sum of the costs of the subsets in the collection.

When the tiles in the database are precomputed using an existing itemset miner (e.g. CHARM), this is an instance of the weighted budgeted maximum coverage problem, which is NP-hard but can be solved approximately to an approximation ratio of $1 - \frac{1}{e}$ using a greedy algorithm. In this algorithm, the k'th tile pattern is selected as the one that maximizes the ratio of the sum of the weights of the newly covered entries divided by its description length.

We have applied this method to two abstract databases after stop-word removal and stemming (turned into binary databases by considering rows as texts and columns as words). The first dataset contains all KDD abstracts between 2001 and 2008, which amounts to 843 documents and 6154 unique stemmed words. The second dataset contains all ICDM abstracts up to 2007, amounting to 859 documents and 5006 unique stemmed words. The 15 tiles first selected in this greedy algorithm are shown in the left column of Tab. 1. Only tiles corresponding to closed itemsets and a support of at least 5 were considered (as mined by CHARM [24]).

KDD

Mining interesting pattern sets (current paper)	$\|I\|$	Tiling databases as described in [7]	$\|I\|$
machin support svm vector	25	data paper	389
art state	39	algorithm propos	246
labeled learn supervised unlabeled	10	data mine	312
associ mine rule	36	base method	202
express gene	25	result show	196
frequent itemset	28	problem	373
graph larg network social	15	data set	279
column row	13	approach	330
algorithm faster magnitud order	12	model	301
algorithm data paper propos real synthetic	27	present	296
answer question	18	larg	286
nearest neighbor	13	applic	271
classifi featur machin support text vector	9	perform	266
precis recal	14	real	255
decis tree	33	inform	240

ICDM

Mining interesting pattern sets (current paper)	$\|I\|$	Tiling databases as described in [7]	$\|I\|$
classifi machin support vector	24	algorithm data	338
analysi discriminant lda linear	10	paper propos	237
associ database mine rule	28	data mine	279
bayes naiv	23	show	370
algorithm discov frequent mine pattern	28	base	369
nearest neighbor	20	result	359
art state	22	approach	349
cluster data dimensional high subspace	11	method	346
account take	19	set	343
play role	14	problem	330
document text word	14	present	305
exampl learn train	17	perform	265
algorithm em expect maximization	8	model	239
frequent item itemset mine	18	larg	221
classifi decis tree	20	algorithm propos	271

Table 1: This table reports the sets of words (columns) J as well as the number of documents $|I|$ containing all these words for the top-15 selected tiles (I, J) for two methods: the tile-mining approach described in this paper (left column) and the tiling databases approach (right column).

3. DISCUSSION

Below we will first try to further elucidate our framework by providing interpretations and clarifying some of the choices we have made. Then we will first discuss some relations with prior work. Finally, we will provide an overview of the extensions and various instantiations of our framework that are subject of current work and that pose interesting challenges for future work.

3.1 Interpretations and remarks

The nature of the data.

In introducing the general framework, we have intentionally treated the data $\mathbf{D}$ as a monolitic block. Our intention with this is to emphasize that our focus is broader than *data sets*. Our framework should be able to handle data that cannot elegantly be cast in a set, such as networks or relational databases. A set often suggests that the elements are commensurable or comparable, perhaps even sampled i.i.d. from some distribution. In this paper, we do not want to make such suggestion or unrealistic assumptions.

The nature of the prior information.

The term prior information may be somewhat misleading, and perhaps more accurately we could also have chosen *prior expectations*. Indeed, the prior information may be wrong (if the data miner is ill-informed), and we believe our framework deals with this in an appropriate way. If the prior information is incorrect, patterns that correct for this will be flagged up as interesting, which is desirable in practice.

Another remark with regard to prior information is that it may seem impractical to list what the data miner already knows. However, we believe that in many cases the most important prior information can be stated at a meta-level, in general terms. For example, the prior information in our running example was on all row and column marginals, specifying $n + m$ constraints in a description of just a few words.

The code describing patterns.

There is a strong connection between a code to describe patterns, and a syntactic choice for the patterns. Indeed, fixing a syntax for the patterns is similar to fixing a code. Simpler patterns in the syntax are be easier to parse and hence probably easier to understand for a data miner.

A communication metaphor.

Our framework can be described using a communication metaphor between the data (Alice) and the data miner (Bob), whereby the data mining algorithm is the intermediary interfacing with both. See Fig. 2 for a graphical illustration. The goal in this communication protocol is to communicate the data as efficiently as possible (i.e. with the shortest possible description), by relying on any prior information the data miner may have. In the first instance good compression can be achieved by relying on a Shannon-optimal code with respect to the MaxEnt model specified by this prior information. However, if the data miner believes or hopes that patterns of a certain easily understandable syntactic form are present in the data, he may ask the Alice to communicate these separately, potentially reducing the overall coding length. In a data mining context, of course only the patterns would be sent, not the rest of the data.

3.2 Relations to prior work

Tiling databases.

The work on tiling databases [7] fits in most closely with our framework, and can be described as a specific instantiation of it. One of the goal in that work was to come up with a collection of a fixed number of tiles (the pattern set) that covers as many database entries as possible. They already observed that set covering techniques can be used to efficiently solve this problem to a guaranteed approximation ratio. The results on two textual datasets described above are shown in Tab. 1.

To see how this method can be viewed as an instance of our framework we need to specify two things: the prior information used, and the code for encoding the patterns. Let us first consider the prior information. Since each database entry is given the same weight, the prior information used is empty, or perhaps non-informative such as assuming that all row marginals are equal, and also all column marginals. As for the coding scheme for the patterns, the same cost is attributed to each of the tiles, such that no distinction is made between tiles with a small or a large circumference.

In this light, it is easy to understand the difference in output between the results of the tile mining method discussed in the running example and the tile mining method presented in [7]. Many tiles found by our method achieve a balance between number of words and documents, since encoding long stretched out tiles comes at a greater cost than compact square tiles. Furthermore, they are less susceptible to common uninformative words, preferring tiles with uncommon words (and although this is harder to see, also preferring tiles overlapping with shorter documents).

KRIMP.

Another related method is KRIMP [20], which attempts to describe the database by constructing a code table of itemsets and encoding the database by making use of this code table. Our approach bears some clear similarities to KRIMP, notably the reliance on coding and description length ideas. However, like with other objective interestingness measures, KRIMP seems less flexible in its current form, and there seems to be no direct way of incorporating properties of the data miner.

Maximum entropy based significance of itemsets.

The maximum entropy principle has been used before for the purpose of designing an interestingness measure for itemsets [22]. Here, the frequency of an itemset is contrasted with the expected frequency based on the frequencies of its subsets. To compute this expected frequency, maximum entropy modeling is used. While this is potentially useful in various applications, it is still an objective measure, that cannot be fine-tuned to suit particular data mining practitioners or tasks.

3.3 Extensions and further work

We are currently working on extending the above ideas in various ways, applying the framework to more general data types, for more general pattern types, and for more general types of prior information. Of course, there are significant interactions between these extensions, but for convenience let us discuss them one by one. After that, we will discuss some other interesting challenges for future work.

Ongoing work.

We have introduced our framework for general data $\mathbf{D}$, as we believe it is likely to be useful for data types different from just binary databases. A first possible extension is toward non-binary databases, such as categorical, integer-valued, and real-valued data (see also [2, 3]). More importantly, we are currently working on instantiating this framework in a flexible way for relational databases. This will allow us to mine for interesting patterns in relational databases in the spirit of the recent papers [16, 10], which have given a new and promising twist to pattern mining research.

Concerning the types of pattern, in a recent paper [13] we have discussed an instantiation of the framework for noisy tiles, with promising empirical results. Other extensions toward frequent itemsets might be of interest as well.

The prior information we have considered in the running example in this paper was restricted to the row and column marginals. Other types of prior informations we are currently considering are the density of certain areas in a binary database, and the support of certain given itemsets. The connection between MaxEnt optimization and the Graphical Models literature will allow us to use results from that community to achieve these goals, such as the Junction Tree algorithm and other techniques for inference and maximum likelihood parameter fitting [23].

An extension similar to this one was made earlier in [11] for randomization approaches to assess data mining results [9]. They introduced different randomization strategies maintaining different properties of a binary dataset besides the row and column marginals, in particular the clustering structure and the frequency of certain itemsets. They suggested this allows data mining to be done in an iterative fashion, updating the randomization model each time a new pattern is reported (and thus becomes part of the prior information). Our framework could accommodate such iterative strategy as well as an alternative to mining pattern sets, as soon as it can handle more complex types of prior information.

Other challenges.

We mentioned earlier that the prior information does not need to be correct for the framework to be useful. However, it would run into problems if the prior information were inconsistent. If this is the case, the method needs to be adapted e.g. by allowing each of the constraints to be vio-

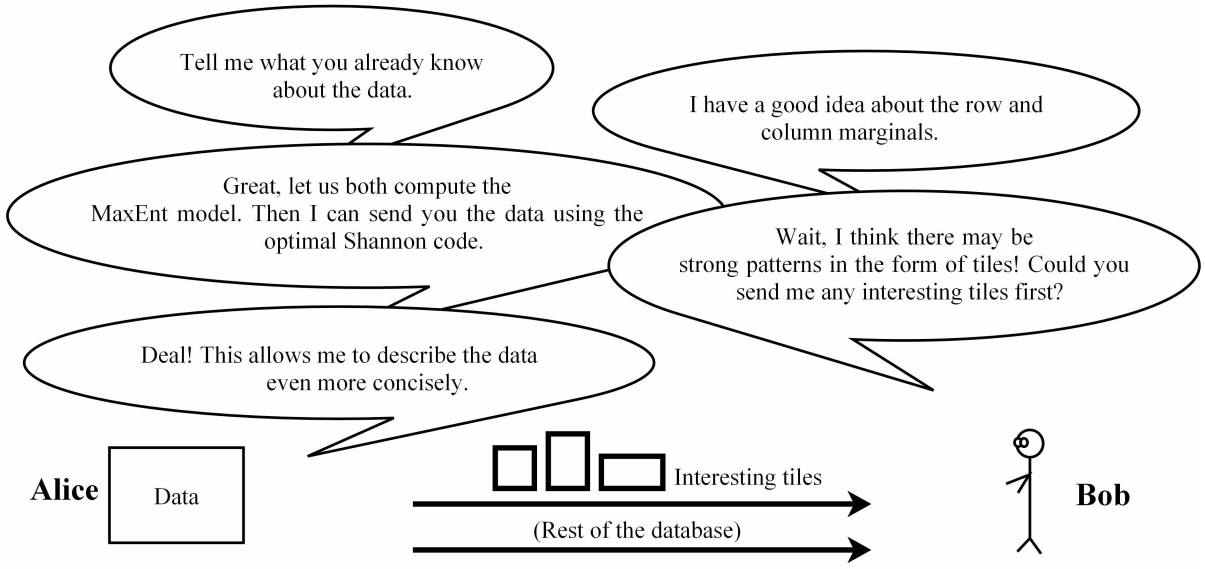

Figure 2: A data mining algorithm implementing our framework can be viewed as an interface moderating the communication between the data miner (Alice) and Bob (the data), trying to help convey the data from Alice to Bob as efficiently as possible. It takes into account what Bob already knows (or thinks to know), as well as the syntactic form of the patterns he believes the data may contain. This figure illustrates that for the running example on mining interesting tiles.

lated by a small amount ε_i. Then, $C \sum_i \varepsilon_i$ can be subtracted from the entropy objective with C akin to a regularization parameter, such that a trade-off is achieved between maximizing the entropy and achieving a good fit to the prior information.

A second important challenge we wish to highlight is the design of efficient algorithms that mine pattern sets as considered in this paper. For the tile mining example, we are currently using a two-step approach, where first all tiles are mined and subsequently they are selected in a greedy way (and thus sorted in the order in which they were selected). It is likely that more efficient algorithms can be devised.

Another challenge is to find out if and how actionability [21] can be incorporated into this scheme. We suspect there may be relations between particular coding schemes for the patterns and certain properties of how they are going to be used, such as the cost of exploiting a pattern or (perhaps equivalently) the profit in exploiting the pattern. However, it is as yet unclear to us if this is the case, or if such connection would be helpful at all.

In this paper, we have chosen to put our framework on statistical foundations. However, it is conceivable that the prior information can be captured in a knowledge base of (possibly probabilistic) logical rules instead. The broad ideas of the framework would stay in place. Doing this would bring the framework closer to the work of [21].

Agreeing on an empirical evaluation strategy.

Finally, a bottleneck we believe this area of research is faced with is the lack of a suitable consensus over how subjective interestingness measures can be assessed empirically, within the scope of a scientific paper. In this paper, we have opted to present some empirical results on a textual data set. The motivation for this is that text is intelligible, certainly if we are familiar with the corpus, and we can assess if the method would provide us with useful insights had we not been familiar with it. However, this risks to raise the wrong impression that the method is supposed to compete with text mining methods (whereas it does not exploit any properties of text at all and any competition would be unfair). Another type of evaluation that is often seen is to use the pattern sets as features for classification. We believe that the relevance of this is limited, as a poor classification accuracy only shows that the features are unrelated to the label, not that they are not interesting. As a result, authors have often resorted to quantitative surrogates such as size of the pattern set and computation times, which are sometimes relevant but usually besides the point when the goal is to design subjective interestingness measures. Therefore, agreeing on an appropriate empirical evaluation strategy is in our opinion critical to progress in this field.

4. CONCLUSIONS

In this paper, we have sketched a possible framework for mining interesting pattern sets. In designing it, our intention was to set it up such that it can operate as an intelligent interface between the data miner and the data, considering both on an equal footing. We designed it as generally as possible, and it is not confined to any particular type of data, pattern, or prior information.

That being said, instantiating the framework for new settings is not always trivial, and issues of computational tractability may arise. This forms perhaps the most important challenge for future research around this framework.

Acknowledgements

This work is supported by the EPSRC grant EP/G056447/1, and by the European Commission through the PASCAL2 Network of Excellence (FP7-216866). KNK is also supported by a University of Bristol Centenary Scholarship.

5. REFERENCES

[1] B. Bringmann and A. Zimmermann. The chosen few: on identifying valuable patterns. In *Proc. of 7th IEEE International Conference on Data Mining (ICDM)*, pages 63–72, 2007.

[2] T. De Bie. Explicit probabilistic models for databases and networks. Technical report, University of Bristol TR 123931, arXiv:0906.5148v1, 2009.

[3] T. De Bie. Maximum entropy models and subjective interestingness: an application to tiles in binary databases. *Data Mining and Knowledge Discovery*, to appear, 2010.

[4] L. De Raedt and A. Zimmermann. Constraint-based pattern set mining. In *Proc. of the 2007 SIAM International Conference on Data Mining (SDM)*, pages 237–248, 2007.

[5] A. Gallo, T. De Bie, and N. Cristianini. MINI: Mining informative non-redundant itemsets. In *Proceedings of the 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 438–445, 2007.

[6] A. Gallo, A. Mammone, T. De Bie, M. Turchi, and N. Cristianini. From frequent itemsets to informative patterns. Technical report, University of Bristol TR 123936, 2009.

[7] F. Geerts, B. Goethals, and T. Mielikäinen. Tiling databases. In *Discovery Science (DS)*, pages 278–289, 2004.

[8] L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys*, 38(3):9, 2006.

[9] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing data mining results via swap randomization. *ACM Transactions on Knowledge Discovery from Data*, 1(3), 2007.

[10] B. Goethals, W. Le Page, and M. Mampaey. Mining interesting sets and rules in relational databases. In *Proc. of the 25th ACM Symposium on Applied Computing (ACM SAC)*, pages 996–1000, 2010.

[11] S. Hanhijarvi, M. Ojala, N. Vuokko, K. Puolamäki, N. Tatti, and H. Mannila. Tell me something I don't know: Randomization strategies for iterative data mining. In *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 379–388, 2009.

[12] S. Jaroszewicz and D. A. Simovici. Interestingness of frequent itemsets using bayesian networks as background knowledge. In *Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 178–186, 2004.

[13] K. Kontonasios and T. De Bie. An information-theoretic approach to finding informative noisy tiles in binary databases. In *Proc. of the 2010 SIAM International Conference on Data Mining (SDM)*, pages 153–164, 2010.

[14] L. G. Kraft. A device for quantizing, grouping, and coding amplitude modulated pulses. Technical report, Massachusetts Institute of Technology, 1949.

[15] K. Lemmens, T. Dhollander, T. De Bie, P. Monsieurs, K. Engelen, B. Smets, J. Winderickx, B. D. Moor, and K. Marchal. Inferring transcriptional modules from chip-chip, motif and microarray data. *Genome Biology*, 7(R37), 2006.

[16] M. Ojala, G. Garriga, A. Gionis, and H. Mannila. Evaluating query result significance in databases via randomizations. In *Proc. of the 2010 SIAM International Conference on Data Mining (SDM)*, pages 906–917, 2010.

[17] M. Ojala, N. Vuokko, A. Kallio, N. Haiminen, and H. Mannila. Randomization of real-valued matrices for assessing the significance of data mining results. In *Proc. of the 2008 SIAM International Conference on Data Mining (SDM)*, pages 494–505, 2008.

[18] B. Padmanabhan and A. Tuzhilin. A belief-driven method for discovering unexpected patterns. In *Proc. of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 94–100, 1998.

[19] B. Padmanabhan and A. Tuzhilin. Small is beautiful: discovering the minimal set of unexpected patterns. In *Proc. of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 54–63, 2000.

[20] A. Siebes, J. Vreeken, and M. van Leeuwen. Item sets that compress. In *Proc. of the 2006 SIAM International Conference on Data Mining (SDM)*, pages 393–404, 2006.

[21] A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proc. of the 1st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 275–281, 1995.

[22] N. Tatti. Maximum entropy based significance of itemsets. *Knowledge and Information Systems*, 17(1):57–77, 2008.

[23] M. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

[24] M. Zaki and C. Hsiao. CHARM: An efficient algorithm for closed itemsets mining. In *Proc. of the 2002 SIAM International Conference on Data Mining (SDM)*, pages 457–473, 2002.