Maximum Entropy Models for Iteratively Identifying Subjectively Interesting Structure in Real-Valued Data

Kleanthis-Nikolaos Kontonasios[1], Jilles Vreeken[2], and Tijl De Bie[1]

[1] Intelligent Systems Laboratory, University of Bristol, Bristol, United Kingdom
{kk8232,Tijl.DeBie}@bristol.ac.uk
[2] Advanced Database Research and Modelling, University of Antwerp, Antwerp, Belgium
Jilles.Vreeken@ua.ac.be

Abstract. In exploratory data mining it is important to assess the significance of results. Given that analysts have only limited time, it is important that we can measure this *with regard to what we already know*. That is, we want to be able to measure whether a result is interesting from a subjective point of view.

With this as our goal, we formalise how to probabilistically model real-valued data by the Maximum Entropy principle, where we allow statistics on *arbitrary* sets of cells as background knowledge. As statistics, we consider means and variances, as well as histograms. The resulting models allow us to assess the likelihood of values, and can be used to verify the significance of (possibly overlapping) structures discovered in the data. As we can feed those structures back in, our model enables iterative identification of subjectively interesting structures.

To show the flexibility of our model, we propose a subjective informativeness measure for tiles, i.e. rectangular sub-matrices, in real-valued data. The *Information Ratio* quantifies how strongly the knowledge of a structure reduces our uncertainty about the data with the amount of effort it would cost to consider it. Empirical evaluation shows that iterative scoring effectively reduces redundancy in ranking candidate tiles—showing the applicability of our model for a range of data mining fields aimed at discovering structure in real-valued data.

1 Introduction

When analysing a database, what we already know will strongly determine which results we will find interesting. Rather than analysing results representing knowledge we already have, we want to discover knowledge that is novel from *our* perspective. That is, the interestingness of a data mining result is highly subjective. Hence, in order to identify and mine such results we need theory and methods by which we can measure interestingness, significance, or surprise, from a *subjective* point of view.

With this as our goal, we formalise how to probabilistically model real-valued data using the Maximum Entropy principle [8], where we allow statistics over arbitrary sets of cells—such as, but not limited to, rectangular tiles—as background information. Using our model, we can measure the likelihood of values and value configurations from the perspective of what we already know about the data. As possible background knowledge, we develop theory for two easily considered sets of statistics for characterising distributions, namely means and variances, and histograms.

While being able to measure the significance of data mining results is important in general, it is particularly so for the sub-fields of data mining that discover local structures, such as in (frequent) pattern mining, subspace clustering, subgroup discovery and bi-clustering—each of which areas where we are typically faced with overly large and highly redundant result sets. As our model allows us to incorporate information on any area of the data, we are able to *iteratively* identify the most surprising pattern out of a given collection of candidate patterns; as by subsequently updating our model with this new information, all variations of the same theme will become predictable and hence onward be considered uninformative.

To quantify the subjective informativeness of a local structure we propose an *Information Ratio* measure for submatrices, or tiles, in real-valued data. It trades off how strongly the knowledge of the structure reduces our uncertainty about the data, with the amount of effort it would cost the analyst to consider it. For binary data, De Bie [5], and Kontonasios and De Bie [10] successfully used an information ratio based measure for identifying surprisingly dense tiles in binary data. Here, we generalise this notion to more rich statistics on expected values of tiles in real-valued data.

The topic of measuring the significance of data mining results was first discussed by Gionis et al. [6], who gave a method for testing significance by swap randomizing binary data. Ojala et al. [15] extended swap randomization to real-valued data. With swap randomization, however, only empirical p-values can be determined. Recently, we gave theory for modelling real-valued data by the Maximum Entropy principle [11]. One key advantage, besides speed, of MaxEnt modelling over swap randomization is that analytical modelling allows us to calculate *exact* probabilities.

Unlike the model we propose here, all of the above provide only relatively weak and *static* null-hypotheses. That is, they can not incorporate information beyond row and column margins, and hence can not identify redundancy in light of previous discoveries. Our model, on the other hand, allows for much stronger statistical testing as rich background knowledge can be incorporated, as well as for iterative use.

Empirical evaluation of our model shows that the information ratio measure reliably identifies highly informative tiles, and that by iteratively including the highest ranked tiles in our model, we correctly and non-redundantly identify the subjectively most informative tiles, subgroups, and bi-clusters on both synthetic and real data.

In sum, the main contributions of this paper are as follows:

- We give the first Maximum Entropy model for real-valued data that can incorporate knowledge of certain useful statistics (mean/variance, and histograms) over arbitrary areas of the data, and hence allows for *iterative* identification of the subjectively most informative data mining result;
- We formalise *Information Ratio* scores for contrasting the subjective informativeness of tiles in real-valued data against their descriptive complexity;

Finally, it is important to stress the main goal of this paper is theoretical. It is explicitly *not* our goal to define practical algorithms for the efficient search of non-redundant selection of tiles/subgroups/biclusters/subspace clusters/etc., all of which will likely need specialised solutions. Instead, our aim here is to provide the theoretical foundations that can be used as building blocks to those ends. As such, the information ratio we here propose for tiles in real-valued data is a proof-of-concept, not a general solution.

2 Related Work

We discuss related work along the following lines: measuring significance, iterative data mining, and identifying informative submatrices in real-valued data.

Assessing the significance of data mining results was first discussed by Gionis et al. [6]. The general idea is to check how likely a result is in data that shares basic properties, but is fully random otherwise. Swap-randomisation, essentially a Markov-chain of many random, local, value-swaps, for generating random data that exactly maintains the row and column margins of the original data. By sampling many random datasets (10 000s), we can subsequently obtain empirical p-values for the result at hand.

Ojala et al. generalised this to maintaining means and variances over rows and columns of real-valued data [15,14]. Recently, De Bie [5], and Kontonasios et al. [11] gave theory to instead model real-valued data analytically by the Maximum Entropy (MaxEnt) principle [8], maintaining properties by expectation instead of exactly. The formalisation by De Bie only allowed to maintain row and column means [5], Kontonasios et al. [11] generalised towards means and variances as well as histograms over rows and columns. Key advantages over swap-randomisation include the speed at which random data can be sampled, and that *exact* probabilities and p-values can be calculated.

None of these can incorporate information on arbitrary *submatrices* as background knowledge. As such, they are essentially static null hypotheses, and hence not applicable for identifying which result is most significant in light of previous discoveries.

The general concept of iterative data mining was first proposed by Hanhijärvi et al. [7]. A key advantage of the iterative approach is that it naturally eliminates redundancy. Based on MaxEnt, to the end that we can make an informed decision which result we should analyse next, Tatti and Vreeken [17] gave a general framework for measuring the difference of results of arbitrary methods in terms of the information they provide about the data at hand, and gave a proof-of-concept for binary data.

This paper broadly follows the lines of the framework for iterative data mining based on Maximum Entropy modelling of prior beliefs [4]. While that work was mostly abstract, we here bring these ideas closer to practice for a broad class of data and patterns. The Information Ratio was introduced in De Bie [5] (there called Compression Ratio) and Kontonasios and De Bie [10], respectively in the context of exact and noisy tile patterns in binary databases. In the current paper it is extended to real-valued data.

Bi-clustering [13] and sub-space clustering [12] are also concerned with identifying sub-matrices in real-valued data that exhibit structure different from the background. As in pattern mining, these approaches typically result in overly large, and highly redundant result sets [19]. Existing proposals to identify significant results either employ simple null-hypotheses that do not allow for iterative updating, require strong assumptions on the distribution of the data, or cannot deal with overlapping tiles [12].

3 Preliminaries

3.1 Notation and Basic Concepts

As data we consider rectangular real-valued matrices $\mathbf{D} \in \mathbb{R}^{n \times m}$. We denote the set of row indices as $\mathcal{I} = \{1, \ldots, n\}$ and the set of column indices as $\mathcal{J} = \{1, \ldots, m\}$. A matrix element can be referred to using an element e from the Cartesian product of $\mathcal{I}$ and $\mathcal{J}$, i.e.: $e \in \mathcal{I} \times \mathcal{J}$. For $e = (i, j)$, $\mathbf{D}_e$ denotes the matrix element on row i and column j, sometimes also explicitly denoted as $\mathbf{D}_{ij}$. W.l.o.g. we assume the attributes to be normalised between 0 and 1.

Since the patterns we will consider in this paper are local patterns, we need to establish a notation for referring to such subsets. A subset of database entries can be indexed using an *index set*, here defined as a subset of the Cartesian product of $\mathcal{I}$ and $\mathcal{J}$, i.e.: $\mathcal{E} \subseteq \mathcal{I} \times \mathcal{J}$. We will use $\mathbf{D}_{\mathcal{E}}$ to refer to the (multi-)set of matrix values indexed by $\mathcal{E}$.

Special cases of index sets that will turn out to be of particular importance in this paper are index sets referring to: the elements within one particular row i (i.e. $\mathcal{E} = \{(i, j) \mid j \in \mathcal{J}\}$); the elements within one particular column j (i.e. $\mathcal{E} = \{(i, j) \mid i \in \mathcal{I}\}$); and the elements within a tile τ, which is defined as the set of elements in the intersection between a set of rows $\mathcal{I}_\tau \subseteq \mathcal{I}$ and a set of columns $\mathcal{J}_\tau \subseteq \mathcal{J}$.

Central in this paper is the notion of a pattern p, which we define as a triple $p = (\mathcal{E}, \mathbf{s}, \hat{\mathbf{s}})$. Here, $\mathcal{E}$ is an index set; $\mathbf{s}$ is a vector-valued function defined over sets of real-valued elements, called a *statistic*; and $\hat{\mathbf{s}}$ is the value the data miner beliefs to hold over the part of the data matrix indexed by $\mathcal{E}$. In general, this will be the empirical value, i.e.: $\hat{\mathbf{s}} = \mathbf{s}(\mathbf{D}_{\mathcal{E}})$, yet our theory below allows $\hat{\mathbf{s}}$ to take any valid value for $\mathbf{s}$.

We will focus on two statistics in particular. We define these by specifying the components of this vector-valued function.

- The first statistic we will consider has two components $s^{(1)}$ and $s^{(2)}$: resp. the function computing the sum of the set of values it is evaluated on, and the function computing the sum of their squares: $s^{(1)}(\mathbf{D}_{\mathcal{E}}) = \sum_{(i,j)\in\mathcal{E}} \mathbf{D}_{ij}$, and $s^{(2)}(\mathbf{D}_{\mathcal{E}}) = \sum_{(i,j)\in\mathcal{E}} \mathbf{D}_{ij}^2$. These two values (along with the cardinality of $\mathbf{D}_{\mathcal{E}}$) are sufficient to compute the mean and the variance of $\mathbf{D}_{\mathcal{E}}$, so patterns defined in these terms inform a user also on the mean and the standard deviation of a set of database elements.
- The second statistic has $d_{\mathcal{E}}$ components $s_{\mathcal{E}}^{(k)}$, specifying a $d_{\mathcal{E}}$-dimensional histogram for the elements indexed by $\mathcal{E}$. More specifically, given a set of $d_{\mathcal{E}} + 1$ real values $b_{0,\mathcal{E}} < b_{1,\mathcal{E}} < \ldots < b_{d_{\mathcal{E}},\mathcal{E}} \in \mathbb{R}$ specifying the boundary values of the histogram bins used for the set $\mathcal{E}$ (typically with $b_{0,\mathcal{E}} = -\infty$ and $b_{d_{\mathcal{E}},\mathcal{E}} = \infty$), we write $k_{\mathcal{E}}$ for the bin index for any $x \in \mathbb{R}$, i.e. $k_{\mathcal{E}}(x) = \max\{k | x < b_{k,\mathcal{E}}\}$. Then the k'th statistic $s_{\mathcal{E}}^{(k)}(\mathbf{D}_{\mathcal{E}})$ in this set is equal to the number of elements from $\mathbf{D}_{\mathcal{E}}$ that fall between $b_{k,\mathcal{E}}$ and $b_{k-1,\mathcal{E}}$, i.e. $s_{\mathcal{E}}^{(k)}(\mathbf{D}_{\mathcal{E}}) = \sum_{(i,j)\in\mathcal{E}} I(k_{\mathcal{E}}(\mathbf{D}_{ij}) = k)$, with I the indicator function. A further piece of notation that will prove useful is $w_{\mathcal{E}}(k) = b_{k,\mathcal{E}} - b_{k-1,\mathcal{E}}$, denoting the width of the k'th bin.

Given our broad definition of a pattern, we can assume that also background knowledge can be specified in terms of patterns, as formalised above, the data miner expects to be present in the data. The set of all triples specifying such background knowledge will be denoted as $\mathcal{B}$. We will show below how we will use such background knowledge to obtain a probability distribution P for the data, representing the data miner's background knowledge on $\mathbf{D}$. Note that the analogy between patterns and the background knowledge will prove useful in an iterative data mining setting: it allows us to naturally incorporate previously shown patterns into the background knowledge.

All logarithms are base 2, and by convention, $0 \log 0 = 0$.

3.2 The Maximum Entropy Principle, a brief primer

In our approach we make use of maximum entropy models, a class of probabilistic models that are identified by the Maximum Entropy principle [8]. This principle states that the best probabilistic model is the model that makes optimal use of the provided information, and that is fully random otherwise. This makes these models very suited for identifying informative patterns: when measuring quality, we know our measurement is based on our background information only, not on undue bias of the distribution.

Entropy maximisation problems are particularly well-suited to deal with background information on *expected* values of certain properties of the data. For example, the data miner may have an expectation about the value $\hat{f}$ of a certain function f when evaluated on the data. We embed this information in the background distribution by requiring that the expected value of f evaluated on $\mathbf{D}$ is equal to $\hat{f}$:

$$\int P(\mathbf{D}) f(\mathbf{D})\, d\mathbf{D} = \hat{f}$$

Thus, inference of background model P is done by solving the following problem:

$$P^* = \max\nolimits_P - \int P(\mathbf{D}) \log\left(P(\mathbf{D})\right)\ d\mathbf{D}\ , \tag{1}$$

$$\text{s.t.} \qquad \int P(\mathbf{D}) f(\mathbf{D})\ d\mathbf{D} = \hat{f}\ , \quad \forall f\ , \tag{2}$$

$$\int P(\mathbf{D})\ d\mathbf{D} = 1 \quad . \tag{3}$$

where each function f computes a statistic of which the data miner expects the value to be $\hat{f}$. It is clear that in the current context, these functions are determined by the triples $(\mathcal{E}, \mathbf{s}, \hat{\mathbf{s}})$ in the background knowledge. More specifically, for each pattern $(\mathcal{E}, \mathbf{s}, \hat{\mathbf{s}})$ there would be a corresponding set of functions f defined as $f(\mathbf{D}) = s^{(k)}(\mathbf{D}_{\mathcal{E}})$, and $\hat{f} = \hat{s}^{(k)}$, and this for each component $s^{(k)}$ of $\mathbf{s}$.

Entropy maximisation subject to such constraints is a convex problem, which can often be solved efficiently. Furthermore, the resulting distributions are known to belong to the exponential family of distributions, the properties of which are very well understood [18]. In particular, the maximum entropy distribution is of the form:

$$P(\mathbf{D}) = \frac{1}{Z} \exp\left\{ \sum_i \lambda_f f(\mathbf{D}) \right\} \quad ,$$

where Z is a normalisation factor known as the partition function and λ_f is a Lagrange multiplier corresponding to the constraint involving f, the value of which can be found by solving the dual optimisation problem.

4 Maximum Entropy Modeling

In this section we discuss the details of the maximum entropy modelling of the statistics discussed in Sec. 3. More in particular, we discuss how the constraint functions f in

Eq. (2) are to be instantiated, and we will provide the resulting probability distribution as the solution to the *MaxEnt* optimisation problem.

Here, we focus on how to *obtain* the MaxEnt distribution; in the next section, we will show how to *use* this model for identifying subjectively interesting tiles.

4.1 Encoding Means and Variances as Prior Knowledge

As we argued in Sec. 3, the information on the empirical mean and variance of a set of values is equivalent with information on their sum and sum of squares, computed by the functions $s^{(1)}$ and $s^{(2)}$ respectively. Thus, to consider background knowledge on the empirical mean and variance for a set of elements $\mathcal{E}$, we need the following two functions as constraint functions in Eq. (2):

$$f_{\mathcal{E}}^{(1)}(\mathbf{D}) \triangleq s^{(1)}(\mathbf{D}_{\mathcal{E}}) = \sum_{(i,j)\in\mathcal{E}} \mathbf{D}_{ij} \quad , \quad f_{\mathcal{E}}^{(2)}(\mathbf{D}) \triangleq s^{(2)}(\mathbf{D}_{\mathcal{E}}) = \sum_{(i,j)\in\mathcal{E}} \mathbf{D}_{ij}^2 \quad .$$

We will denote the corresponding Lagrange multipliers as $\lambda_{\mathcal{E}}^{(1)}$ and $\lambda_{\mathcal{E}}^{(2)}$ respectively. Since we need these functions for each $\mathcal{E}$ for which background knowledge is available, the total number of constraints of the type of Eq. (2) is equal to twice the number of index sets $\mathcal{E}$ for which the sum of the elements and the sum of the squares of the elements are part of the background knowledge.

Shape of the solution. By solving the MaxEnt optimization problem using Lagrange Multiplier theory we obtain the following probability distribution:

$$P(\mathbf{D}) = \prod_{i,j} \sqrt{\frac{\beta_{ij}}{\pi}} \cdot \exp\left\{ -\frac{(\mathbf{D}_{ij} + \frac{1}{2} \cdot \frac{\alpha_{ij}}{\beta_{ij}})^2}{\frac{1}{\beta_{ij}}} \right\} \ , \tag{4}$$

where the variables α_{ij} and β_{ij} can be expressed as follows in terms of the Lagrange multipliers for the individual constraints:

$$\alpha_{ij} = \sum_{\mathcal{E}:(i,j)\in\mathcal{E}} \lambda_{\mathcal{E}}^{(1)} \quad , \quad \beta_{ij} = \sum_{\mathcal{E}:(i,j)\in\mathcal{E}} \lambda_{\mathcal{E}}^{(2)} \quad .$$

Thus the resulting probability distribution is a product of $n \times m$ independent *Normal* distributions, with means equal to $\mu_{ij} = -\frac{\alpha_{ij}}{2\beta_{ij}}$ and variance $\sigma_{ij}^2 = \frac{1}{2\beta_{ij}}$. Each factor in the product of Eq. (4) corresponds to the probability distribution for one value $\mathbf{D}_{ij}$ in the database. Sampling a full database from P hence comes down to $n \times m$ independent univariate sampling operations. Also, note that the distribution of $\mathbf{D}_{ij}$ depends *only* on the Lagrange multipliers of the subsets $\mathcal{E}$ in which the cell participates, i.e. for which $(i, j) \in \mathcal{E}$, which makes the sampling easier to conduct.

4.2 Encoding Histograms as Prior Knowledge

In order to encode background knowledge on the histogram within an index set $\mathcal{E}$, we need $d_{\mathcal{E}}$ constraint functions (as explained in Sec. 3, we assume that the histogram for subset $\mathcal{E}$ contains $d_{\mathcal{E}}$ bins, defined by the bin boundaries $b_{0,\mathcal{E}}, b_{1,\mathcal{E}}, \ldots, b_{d_{\mathcal{E}},\mathcal{E}} \in \mathbb{R}$, i.e. we allow number and values of the bin boundaries to be different for different $\mathcal{E}$). These functions are defined as: $f_{\mathcal{E}}^{(k)}(\mathbf{D}) \triangleq s_{\mathcal{E}}^{(k)}(\mathbf{D}_{\mathcal{E}}) = \sum_{(i,j)\in\mathcal{E}} I(k_{\mathcal{E}}(\mathbf{D}_{ij}) = k)$ where $k_{\mathcal{E}}(\mathbf{D}_{ij})$ denotes the bin index for the value $\mathbf{D}_{ij}$ (see Sec. 3). The functions $f_{\mathcal{E}}^{(k)}$ calculate the number of elements from $\mathbf{D}_{\mathcal{E}}$ that fall in the k'th bin of the histogram. We will denote the corresponding Lagrange multiplier as $\lambda_{\mathcal{E}}^{(k)}$.

Shape of the solution. Using basic Lagrange multiplier theory one can show that the resulting probability distribution decomposes to a product of $n \times m$ independent probability distributions, i.e. $P(\mathbf{D}) = \prod_{i,j} P_{ij}(\mathbf{D}_{ij})$. Each component P_{ij} has the form

$$P_{ij}(\mathbf{D}_{ij}) = \frac{1}{Z_{ij}} \cdot \exp\left\{ \sum_{\mathcal{E}:(i,j)\in\mathcal{E}} \sum_{k=1}^{d_{\mathcal{E}}} \lambda_{\mathcal{E}}^{(k)} I\left(k_{\mathcal{E}}(\mathbf{D}_{ij}) = k\right) \right\} \quad ,$$

where Z_{ij} is the decomposed partition function, formally defined as

$$Z_{ij} = \int_0^1 \exp\left\{ \sum_{\mathcal{E}:(i,j)\in\mathcal{E}} \sum_{k=1}^{d_{\mathcal{E}}} \lambda_{\mathcal{E}}^{(k)} I\left(k_{\mathcal{E}}(\mathbf{D}_{ij}) = k\right) \right\} d\mathbf{D}_{ij} \quad .$$

Each component refers to one data cell and it is affected only by the Lagrange Multipliers assigned to the sets $\mathcal{E}$ in which the entry participates.

4.3 Inferring the Model

The values of the λ parameters, which uniquely determine the MaxEnt model, are inferred by solving the Lagrange duals of the respective MaxEnt optimisation problems. These dual optimisation problems are convex, such that they can be solved efficiently using simple and well-known optimisation methods such as Conjugate Gradient (CG, the method of our choice in this paper) providing the gradient can be computed efficiently. Due to lack of space details cannot be shown here, but the theoretical complexity of each CG step is $O(\#\lambda^2)$ (with $\#\lambda$ the number of Lagrange multipliers), dominated by the cost of computing the gradient vector. In practice, however, we observe that run time develops linearly with the number of λs.

5 Measuring Subjective Interestingness

In this section we discuss how to use a MaxEnt model for measuring the subjective interestingness of a pattern. From 5.4 onward we focus on the specific case of tiles, a well-known and intuitive pattern type that lends itself for description in simple terms.

5.1 Quantifying Subjective Interestingness

The main goal of this section is defining a measure for the *subjective* interestingness of a pattern from the user's point of view. That is, how strongly a pattern contrasts to what the user already knows or beliefs about the data.

Loosely speaking, for a given pattern p, we have to define two important properties; the first we refer to as the *Information Content* of a pattern, or, the amount of information the pattern can convey to the user. The second property we refer to as the *Description Length* of a pattern, or, the cost for transmitting this information.

We can assume that a user, given his limited capacity for information processing, will find patterns with larger *Information Content* for a constant *Description Length* more interesting. In other words, the interestingness of a pattern can be formalised as a trade-off between these two quantities. This motivates the *Information Ratio* as a suitable measure of a pattern's interestingness.

Definition. The Information Ratio of a pattern p is the ratio of its *Information Content* and *Description Length*:

$$\mathit{InfRatio}(p) = \frac{\mathit{InfContent}(p)}{\mathit{DescrLength}(p)} \quad ,$$

The Information Ratio was originally introduced in the context of exact tiles in binary data [5]. There it was shown that iteratively mining the tile with the highest *InfRatio* amounts to searching for a set of tiles with maximal total self-information given a limited budget on the overall description length. This was abstracted in [4] by relating iterative data mining in general to the budgeted set coverage problem, further justifying the use of this measure in iterative data mining.

5.2 Information Content of a Pattern

Our goal here is to quantify the information a user can extract from a pattern. We define the $\mathit{InfContent}$ for a pattern p as the number of bits we gain when using p in addition to our current background knowledge when describing $\mathbf{D}$. More formally,

$$\mathit{InfContent}(p) = L(\mathbf{D} \mid \mathcal{B}) - L(\mathbf{D} \mid \mathcal{B} \cup \{p\}) \quad ,$$

where $L(\mathbf{D} \mid \mathcal{B})$ is the number of bits required to describe $\mathbf{D}$ using only $\mathcal{B}$, and $L(\mathbf{D} \mid \mathcal{B} \cup \{p\})$ is the number of bits to describe $\mathbf{D}$ using both $\mathcal{B}$ and p.

We have, for the number of bits to encode $\mathbf{D}$ given background knowledge $\mathcal{B}$,

$$L(\mathbf{D} \mid \mathcal{B}) = \sum_{(i,j) \in \mathcal{I} \times \mathcal{J}} L(\mathbf{D}_{ij} \mid \mathcal{B}) \quad ,$$

where we transmit, in a fixed order, the value of each cell $\mathbf{D}_{ij} \in \mathbf{D}$. To encode a specific value, we use optimal prefix codes [3]. We obtain the probabilities from our maximum entropy model P built using the information in $\mathcal{B}$. We hence have

$$L(\mathbf{D}_{ij} \mid \mathcal{B}) = -\log P_{ij}(\mathbf{D}_{ij}) \quad ,$$

Intuitively, the better our model predicts the value of cell, i.e. the more expected a value is, the fewer bits are needed to encode it.

5.3 Estimating *InfContent*

With the above, we can measure the subjective interestingness of a pattern p, and so identify the most informative pattern from a large collection. To do so, however, the above formalisation requires us to infer the maximum entropy model for $\mathcal{B} \cup p$, and this for each candidate pattern p. This is clearly bound to be computationally prohibitive for large collections of candidate patterns. We therefore take a different approach, and instead *estimate* the gain of adding a pattern $p = (\mathcal{E}, \mathbf{s}, \hat{\mathbf{s}})$ to our model by focusing on the information gained regarding the database entries in $\mathcal{E}$. That is,

$$\mathit{InfContent}(p) = L(\mathbf{D}_{\mathcal{E}} \mid \mathcal{B}) - L(\mathbf{D}_{\mathcal{E}} \mid p) \quad ,$$

which is the difference between the number of bits we need under the current model to describe the values of $\mathbf{D}_{\mathcal{E}}$ corresponding to the pattern, and the number of bits we would need to encode this area by solely using the information the pattern p provides. This approximation will be good as long as the pattern contains significantly more information about the database entries $\mathcal{E}$ concerned than the background information—a reasonable assumption given our focus on identifying the most informative patterns.

As discussed in Sec. 5.2, the first term $L(\mathbf{D}_{\mathcal{E}} \mid \mathcal{B})$ can be computed directly. For calculating $L(\mathbf{D}_{\mathcal{E}} \mid p)$ we consider two approaches, corresponding to the statistics discussed in Sec. 3.1. Each results in a different way of calculating the overall $\mathit{InfContent}$:

- Considering the first statistic, a pattern specifies the mean μ and variance σ^2 of the values in $\mathbf{D}_{\mathcal{E}}$. We know [3] that in this case the maximum entropy model for $\mathbf{D}_{\mathcal{E}}$ reduces to the normal distribution $\mathcal{N}(\mu, \sigma^2)$. Hence, with ϕ_{μ,σ^2} the normal density function, we have (with the subscript m to denote *mean and variance*):

  $$L_m(\mathbf{D}_{\mathcal{E}} \mid p) = \sum_{(i,j)\in\mathcal{E}} -\log \phi_{\mu,\sigma^2}(\mathbf{D}_{ij}) \quad ,$$

  by which we encode the values in $\mathbf{D}_{\mathcal{E}}$ by an optimal prefix code proportional to the probability of the entry under this normal distribution.
- Patterns defined in terms of the second type of statistic specify a histogram for the values in $\mathcal{E}$. The maximum entropy model subject to this information reduces to a piecewise-constant density function, uniform with value $\frac{\hat{s}_{\mathcal{E}}^{(k)}}{|\mathcal{E}|} \times \frac{1}{w(k)}$ within the k'th bin of the histogram, where $\frac{\hat{s}_{\mathcal{E}}^{(k)}}{|\mathcal{E}|}$ represents the fraction of values from $\mathbf{D}_{\mathcal{E}}$ specified to belong to the k'th bin. (This follows from the fact that the MaxEnt distribution with a bounded domain is uniform [3].) Hence, we have (with the subscript h to denote *histogram*):

  $$L_h(\mathbf{D}_{\mathcal{E}} \mid p) = \sum_{(i,j)\in\mathcal{E}} L_h(\mathbf{D}_{ij} \mid p) \quad ,$$

  where

  $$L_h(\mathbf{D}_{ij} \mid p) = -\log\left(\frac{\hat{s}_{\mathcal{E}}^{(k_{\mathcal{E}}(\mathbf{D}_{ij}))}}{|\mathcal{E}|}\right) - \log\left(\frac{1}{w(k_{\mathcal{E}}(\mathbf{D}_{ij}))}\right) \quad ,$$

  in which, per entry $\mathbf{D}_{ij}$, the first term represents the description length for the bin index the entry falls in, and the second term for the actual value within that bin.

5.4 Description Length of a Pattern

So far our discussion has been fully general with respect to the choice of $\mathcal{E}$. From now on our exposition will be focused on patterns defined by a tile. Recall that a tile, denoted as τ, is defined as a sub-matrix of the original data matrix. The reason for focusing on tiles is that a tile τ can be described conveniently and compactly by specifying their defining row set $\mathcal{I}_\tau$ and column set $\mathcal{J}_\tau$.

Thus, for the *Description Length* of a tile pattern $p = (\mathcal{E}, \mathbf{s}, \hat{\mathbf{s}})$ where $\mathcal{E} = \{(i,j) \mid i \in \mathcal{I}_\tau, j \in \mathcal{J}_\tau\}$, we have $\mathit{DescrLength}(p) = L(\mathcal{E}) + L(\mathbf{s}, \hat{\mathbf{s}}) \quad = L(\mathcal{I}_\tau) + L(\mathcal{J}_\tau) + L(\mathbf{s}, \hat{\mathbf{s}})$, where $L(\mathcal{I}_\tau) + L(\mathcal{J}_\tau)$ is the number of bits we need to identify the position of the tile. $L(\mathcal{I}_\tau)$ can be computed as:

$$L(\mathcal{I}_\tau) = \log(n) + \log \binom{n}{|\mathcal{I}_\tau|}$$

where the first term accounts for the transmission of the height $|\mathcal{I}_\tau|$ of tile τ. With this information, we can now identify which rows of $\mathbf{D}$ are part of the tile using an index over a binomial. We calculate $L(\mathcal{J}_\tau)$ analogously. Note that as we know the exact counts, encoding through an index over a binomial is at least as efficient as using individual prefix codes [3] as it makes optimal use of the available knowledge.

The quantity $L(\mathbf{s}, \hat{\mathbf{s}})$ scores the cost for transmitting the remaining information conveyed by the pattern. It is straightforward that the way $\mathit{InfContent}$ is calculated dictates a certain approach for the $\mathit{DescrLength}$ as well. Let us discuss these here in turn.

- For mean and variance of a tile as background knowledge we have

$$L_m(\mathbf{s}, \hat{\mathbf{s}}) = 2\log(10^{acc})$$

where we encode $mean(\mathbf{D}_\tau)$ and $var(\mathbf{D}_\tau)$ using a uniform prior.

- In the case of histograms as background information, we have

$$L_h(\mathbf{s}, \hat{\mathbf{s}}) = L_{\mathbb{N}}(d_{\mathcal{E}}) + \log \binom{10^{acc}}{d_{\mathcal{E}} - 1} + \log \binom{|\mathcal{E}| + d_{\mathcal{E}} - 1}{d_{\mathcal{E}} - 1}$$

where we first encode the number of bins $d_{\mathcal{E}}$ in the histogram, using the MDL optimal code $L_{\mathbb{N}}$ for integers ≥ 1 [16]. This encoding requires progressively more bits the higher the value—by which we explicitly reward simple histograms. In the next term, $\log \binom{10^{acc}}{d_{\mathcal{E}}-1}$, we encode the split points between the bins. These terms account for specifying the histogram, i.e. the statistic $\mathbf{s}$ used.
The last term, $\log \binom{|\mathcal{E}|+d_{\mathcal{E}}-1}{d_{\mathcal{E}}-1}$, encodes $\hat{\mathbf{s}}$: how many observations fall within each bin. We have to partition $|\mathcal{E}|$ entries over $d_{\mathcal{E}}$ possibly empty bins. This is known as a weak composition. The number of weak compositions of k non-negative terms summing up to n is given by $\binom{n+k-1}{k-1}$. Assuming an ordered enumeration, we need $\log \binom{|\mathcal{E}|+d_{\mathcal{E}}-1}{d_{\mathcal{E}}-1}$ bits to identify our composition. Note that $\log \binom{n}{k} = \log \Gamma(n+1) - \log \Gamma(k+1) - \log \Gamma(n-k+1)$ and hence is calculable even for large n and k.

Each of these methods require an accuracy level acc to be specified. Ultimately to be decided by the user, a natural choice is the number of significant digits of the data [9].

5.5 Instantiating *InfRatio*

The discrimination between transmitting mean/variance and histograms allows us to instantiate the $\mathit{InfRatio}$ of a tile τ in two different ways per scheme.

$$\mathit{InfRatio}_s(p) = \frac{L(\mathbf{D}_{\mathcal{E}} \mid \mathcal{B}) - L_s(\mathbf{D}_{\mathcal{E}} \mid p)}{L(\mathcal{E}) + L_s(\mathbf{s}, \hat{\mathbf{s}})}$$

where s is the statistic used as prior knowledge in the modelling of the database.

6 Iteratively Identifying Subjectively Interesting Structure

In Section 4 we discussed how to obtain a Maximum Entropy model for a real-valued database under background information of statistics $\mathbf{s}$ over arbitrary sets of cells $\mathcal{E}$ of the data, and Section 5 proposed $\mathit{InfRatio}$ as a measure for the informativeness of patterns in the data based on their unexpectedness under this background knowledge.

In practice, to discover novel knowledge, we propose to iteratively find the most informative pattern with regard to background knowledge; then present this pattern to the user, and continue our search after incorporating the pattern in the background knowledge and updating the MaxEnt model accordingly.

What background knowledge to start the search from is up to the user; it may be empty, or can consist of already known patterns. In our experiments, we choose to compute the initial Maximum Entropy model with statistics on the row and column distributions as prior knowledge. We then use $\mathit{InfRatio}$ to rank a collection of candidate patterns $\mathcal{F}$, and identify the top-ranked pattern as most interesting. This pattern is henceforth considered prior knowledge. From this point the algorithm iterates with

1. a Maximum Entropy modelling step, using all accumulated background knowledge
2. an $\mathit{InfRatio}$ ranking for the tiles in $\mathcal{F}$ in each iteration.

We terminate when user dependent criteria are met. These criteria can be objective (e.g. top-k, log-likelihood of the data, or a model selection criterion such as MDL, BIC or AIC), subjective (e.g. human evaluation of patterns) or a combination of both.

7 Experiments

In this section we empirically evaluate our MaxEnt model for real-valued data. We stress, however, that the contribution of this paper is theoretical in nature—this section should be regarded as proof-of-concept, not a final application. Though our modelling theory is general for $\mathcal{E}$, for the practical reasons of defining meaningful $\mathit{InfRatio}$, as well as for mining candidate patterns $\mathcal{F}$ we here restrict ourselves to tiles.

7.1 Setup

We evaluate whether iterative ranking helps to correctly identify the most informative tiles on data with known ground truth. Second, for synthetic and real data, we investigate the $\mathit{InfRatio}$ and log-likelihood curves of the top-k identified patterns.

We implemented our model in C++, and make our code available for research purposes.[3] All experiments were performed on a 2.66GHz Windows 7 machine.

As initial background knowledge to our model, we always include statistics over the row and column distributions.

Data In order to evaluate performance with a known ground truth, we use synthetic data. We generate a 500-by-500 dataset, in which we plant four complexes of five overlapping tiles, for which the values are distributed significantly different from the background. We refer to this dataset as *Synthetic*.

Per complex, we plant three tiles of 15-by-15, one of 10-by-10, and one of 8-by-8. Values of cells not associated with a complex are drawn from a Normal distribution with mean $\mu = 0.3$ and variance $\sigma = 0.15$. For cells associated with complex A we use $\mu = 0.15$ and $\sigma = 0.05$; for complex B we use $\mu = 0.2$ and $\sigma = 0.05$; for complex C, $\mu = 0.3$ and $\sigma = 0.05$; and for complex D, $\mu = 0.4$ and $\sigma = 0.05$.

For the set of candidate tiles to be ranked, in addition to the 20 true tiles, we randomly generate 230 tiles of 15-by-15. Per random tile we uniformly randomly select columns and rows. Note that by generating these at random, they may or may not overlap with the complexes, and hence may or may not identify a local distribution significantly different from the background.

We also evaluate on real data. The *Alon* dataset is a 62-by-2000 gene expression dataset [1]. To obtain tiles for this data, using CORTANA[4] at default settings, we mined the top-1000 subgroups of up to 3 attributes. Subgroups naturally translate into tiles; simply consider the features of the rule as columns, and the transactions satisfying the rule as rows. The Arabidopsis thaliana, or *Thalia*, is a 734-by-69 gene expression dataset.[5] For this data we mined 1 600 biclusters using BIVISU at default settings [2].

7.2 Ranking tiles

In our first experiment we evaluate iteratively identifying subjectively informative patterns. In particular, we use *InfRatio* to rank the 250 candidate tiles of the *Synthetic* dataset, starting with background knowledge $\mathcal{B}$ containing only statistics on the row and column distributions. At every iteration, we incorporate the top-ranked tile into the model, and re-rank the candidates.

We evaluate modelling with means-variances, and with histograms information. For both, we report both the ranking of the first five iterations, and the final top-10 iterative ranking. We depict the id of the top-ranked tile in an iteration in bold. We give the results in Table 1. On the left, we give the *InfRatio* rankings for transmitting means-variances, i.e. using L_m, while on the right we give the rankings for when transmitting histograms information, i.e. using L_h.

Table 1 shows that *InfRatio* consistently ranks the largest and least-overlapping tiles at the top. In the first iteration the top-10 tiles include the largest tiles from each

[3] `http://www.tijldebie.net/software`

[4] CORTANA: `http://datamining.liacs.nl/cortana.html`

[5] `http://www.tik.ee.ethz.ch/~sop/bimax/`

Table 1. Iterative Ranking. The top-10 ranked tiles for the first five iterations, plus the final ranking of the top-10 most informative tiles, from 250 candidate tiles. We give the results for the *Synthetic* dataset both for mean-variance modelling and transmission using L_m (left), resp. for histogram modelling with transmission using L_h (right). Tiles prefixed with a letter denote planted tiles, those denoted with a plain number are random.

	Mean-Variances						Histograms					
Rank	It 1	It 2	It 3	It 4	It 5	Final	It 1	It 2	It 3	It 4	It 5	Final
1.	**A2**	**B3**	**A3**	**B2**	**C3**	**A2**	**A1**	**C2**	**B1**	**C1**	**D1**	**A1**
2.	A4	B4	B2	C3	C4	**B3**	C2	B1	B2	D1	D3	**C2**
3.	A3	B2	C3	C4	C2	**A3**	B1	C1	B3	D3	B2	**B1**
4.	B3	A3	C4	C2	D2	**B2**	C1	C3	C1	B2	D2	**C1**
5.	B4	C3	C2	B4	D4	**C3**	C3	B2	D1	D2	A2	**D1**
6.	B2	C4	B4	D2	D3	**C2**	B2	B3	D3	A2	84	**B2**
7.	C3	C2	D2	D4	D1	**D2**	B3	D1	D2	C3	25	**A2**
8.	C4	D2	D4	D3	A5	**D3**	A3	D3	A2	25	228	**D2**
9.	C2	D4	D3	D1	21	**A5**	A2	D2	C3	84	43	**228**
10.	D2	D3	B1	A5	B5	**B5**	D1	A2	25	228	33	**43**

of the four complexes. The final ranking shows that for each complex large and little-overlapping tiles are selected. Note that due to the random value generation some tiles within a complex may stand out more from the rest of the data than others.

In general, modelling with histograms is more powerful than the means-variances case. Clearly, this does rely on the quality of the used histograms. Here, we use histograms that balance complexity with the amount of data by the MDL principle [9]. As we here mainly consider small tiles, the obtained histograms are likely to underfit the underlying Normal distribution of the data. Nevertheless, Table 1 shows that the largest tiles of each complex are again top-ranked. As an artifact of the above effect, the final ranking does include a few randomly generated tiles—for which we find that by chance their values indeed differ strongly from the background distribution, and hence identify potentially interesting areas.

From this experiment we conclude that our *InfRatio* for real-valued tiles coupled with iterative modelling leads to the correct identification of subjectively interesting patterns, while strongly reducing redundancy.

7.3 *InfRatio* and log-likelihood curves

Next, we examine *InfRatio* and the iteratively obtained rankings on both artificial and real-world data. Evaluating the interestingness of patterns found in real-world data is highly subjective, however. The negative log-likelihood of the data is often considered as a measure of the surprise of the data as a whole. Finding and encoding informative patterns will provide more insight in the data, and so decrease surprise; the negative log-likelihood scores. Since we have a probabilistic model computing these scores is straightforward. In our experiments we expect to see significant decreases of the score after the discovery of a surprising pattern.

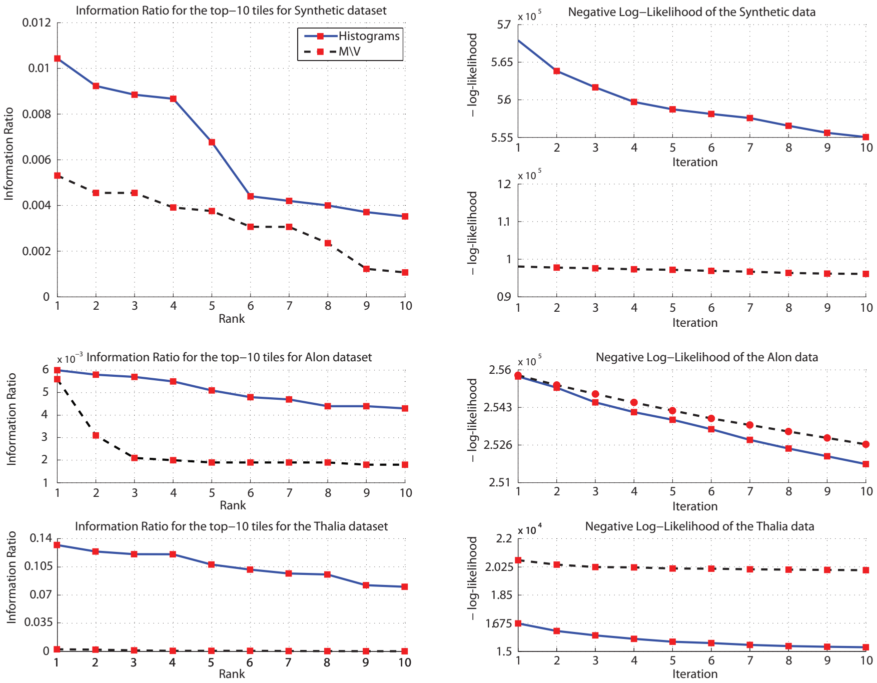

Fig. 1. *InfRatio* values for the top-10 ranked tiles (left column) and negative log-likelihood of the data (right) for the *Synthetic* (top row), *Alon* (middle row), and *Thalia* (bottom row) datasets. The solid blue line corresponds to histograms, the dashed black line to means variance modelling. Note the downward trend for both plot types: the most informative tiles are ranked first.

Figure 1 presents the *InfRatio* and negative log-likelihood scores for the first 10 iterations of our algorithm for resp. the *Synthetic*, *Alon*, and *Thalia* datasets.

We observe the *InfRatio* values are non-increasing over the whole range of the ten iterations, for all datasets and coding schemes. An immediate explanation is that in successive iterations more (correct) information is encoded into the model, and hence patterns become at most as surprising as they were in previous iterations. We further observe decreasing negative log-likelihood curves over the ten iterations. The fact that the data gets less surprising in every iteration is an indication that the patterns discovered and encoded in the model are significant. Loosely speaking, we can also say that the more the decrease per iteration the more interesting the corresponding pattern is.

We see that for *Synth* means-variance modelling obtains the best scores (note the very low negative log-likelihood scores). This follows as the data was generated by Normal distributions. For our real datasets, on the other hand, histogram modelling is in its element as these datasets are less well explained by Gaussians.

The modelling time for these datasets, for the first ten iterations range between seconds (*Synthetic*), tens of seconds (*Thalia*, and Histograms on *Alon*), up to hundreds of seconds for *Alon* when modelling means/variances. Scalability experiments (not shown due to lack of space) show run time scales linearly with the number of added tiles.

8 Discussion

The experiments verify that by iteratively identifying the most surprising tile, and updating our model accordingly, we identify the top-k informative tiles without redundancy.

Our model is generally applicable for measuring the significance of results under background knowledge that can be cast in the form of statistics over sets of entries, yet it is currently particularly suited for iteratively measuring the significance of tiles.

We note that our analytical model allows a lot of freedom for constructing measures of surprise, interestingness, or expected performance for tiles. In previous work [11], we formalised how to calculate the expected Weighted Relative Accuracy (WRAcc) of a pattern, such as used in subgroup discovery. More research on how to calculate expected performance on other popular measures is needed.

In this paper, we are not concerned with mining interesting tiles *directly*; That is, we here assume the candidate tiles are discovered externally, and we 'simply' order them; by which it is currently particularly applicable for identifying the informative and non-redundant results in pattern mining, bi-clustering, and sub-space clustering of real-valued data. The development of efficient algorithms for mining surprising tiles will make for exciting future work—likely with specific solutions per setting, as different goals are to be optimised.

Moreover, our modelling theory will allow a generalisation of the recent proposal by Tatti and Vreeken [17] on measuring differences between data mining results to real-valued data—given meaningful translation of results into patterns $p = (\mathcal{E}, \mathbf{s}, \hat{\mathbf{s}})$.

Perhaps most interesting from our perspective is to extend our modelling theory further towards richer types of structure. A straightforward step would be to incorporate into the model not just a statistic over the whole tile, but instead do so per row and/or column. Beyond means-variances, and histograms, there are other important notions of structure such as similarities between rows, as well as correlations between attributes—each of which will further extend the applicability towards the above applications.

In this light it is important to note that our models, and in particular the histogram variant, are already directly applicable for evaluating significances and measuring subjective interestingness on ordinal and integer valued data.

9 Conclusion

We formalised how to probabilistically model a real-valued dataset by the Maximum Entropy principle, such that we can iteratively feed in background information on the distributions of arbitrary subsets of elements in the database. To show the flexibility of our model, we proposed the $\mathit{InfRatio}$ measure to quantify the subjective interestingness of tiles, trading off how strongly the knowledge of the pattern reduces our uncertainty about the data with how much effort would it costs the analyst to consider it.

Empirical evaluation showed that by iteratively scoring candidate tiles, and subsequently updating the model with the most informative tile, we can effectively reduce redundancy in the set of candidate tiles—showing the applicability of our model for a range of data mining fields dealing with real-valued data.

Acknowledgements

Kleanthis-Nikolaos Kontonasios and Tijl De Bie are supported by EPSRC grant EP/G056447/1, the European Commission through the PASCAL2 Network of Excellence (FP7-216866) and a UoB Scholarship. Jilles Vreeken is supported by a Post-Doctoral Fellowship of the Research Foundation – Flanders (FWO).

References

1. U. Alon, N. Barkai, D. A. Notterman, K. Gish, D. M. S. Ybarra, and A. J. Levine. Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, 96(12):6745–6750, 1999.
2. K. Cheng, N. Law, W. Siu, and T. Lau. Bivisu: software tool for bicluster detection and visualization. *Bioinformatics*, 23(17):23–42, September 2007.
3. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience New York, 2006.
4. T. De Bie. An information theoretic framework for data mining. In *KDD*, pages 564–572. ACM, 2011.
5. T. De Bie. Maximum entropy models and subjective interestingness: an application to tiles in binary databases. *Data Min. Knowl. Disc.*, 23(3):407–446, 2011.
6. A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing data mining results via swap randomization. *ACM TKDD*, 1(3):167–176, 2007.
7. S. Hanhijärvi, 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 *KDD*, pages 379–388. ACM, 2009.
8. E. Jaynes. On the rationale of maximum-entropy methods. *Proc. IEEE*, 70(9):939–952, 1982.
9. P. Kontkanen and P. Myllymäki. MDL histogram density estimation. In *AISTATS*, 2007.
10. K.-N. Kontonasios and T. De Bie. An information-theoretic approach to finding noisy tiles in binary databases. In *SDM*, pages 153–164. SIAM, 2010.
11. K.-N. Kontonasios, J. Vreeken, and T. De Bie. Maximum entropy modelling for assessing results on real-valued data. In *ICDM*, pages 350–359. IEEE, 2011.
12. H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM TKDD*, 3(1):1–58, 2009.
13. S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM TCBB*, 1(1):24–45, 2004.
14. M. Ojala. Assessing data mining results on matrices with randomization. In *ICDM*, pages 959–964, 2010.
15. M. Ojala, N. Vuokko, A. Kallio, N. Haiminen, and H. Mannila. Randomization methods for assessing data analysis results on real-valued matrices. *Stat. Anal. Data Min.*, 2(4):209–230, 2009.
16. J. Rissanen. Modeling by shortest data description. *Annals Stat.*, 11(2):416–431, 1983.
17. N. Tatti and J. Vreeken. Comparing apples and oranges - measuring differences between exploratory data mining results. *Data Min. Knowl. Disc.*, 25(2):173–207, 2012.
18. M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Foun. Trends Mach. Learn.*, 1(1-2):1–305, 2008.
19. A. Zimek and J. Vreeken. The blind men and the elephant: on meeting the problem of multiple truths in data from clustering and pattern mining perspectives. *Mach. Learn.*, 2013. In Press.