An Information Theoretic Framework for Data Mining

Tijl De Bie
University of Bristol, Intelligent Systems Laboratory
MVB, Woodland Road
Bristol, UK
tijl.debie@gmail.com

ABSTRACT

We formalize the data mining process as a process of information exchange, defined by the following key components. The data miner's state of mind is modeled as a probability distribution, called the background distribution, which represents the uncertainty and misconceptions the data miner has about the data. This model initially incorporates any prior (possibly incorrect) beliefs a data miner has about the data. During the data mining process, properties of the data (to which we refer as patterns) are revealed to the data miner, either in batch, one by one, or even interactively. This acquisition of information in the data mining process is formalized by updates to the background distribution to account for the presence of the found patterns.

The proposed framework can be motivated using concepts from information theory and game theory. Understanding it from this perspective, it is easy to see how it can be extended to more sophisticated settings, e.g. where patterns are probabilistic functions of the data (thus allowing one to account for noise and errors in the data mining process, and allowing one to study data mining techniques based on subsampling the data). The framework then models the data mining process using concepts from information geometry, and I-projections in particular.

The framework can be used to help in designing new data mining algorithms that maximize the efficiency of the information exchange from the algorithm to the data miner.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications—*Data mining*

General Terms

Algorithms, Theory

Keywords

Data mining framework, subjective interestingness, MaxEnt

KDD'11, August 21–24, 2011, San Diego, California, USA.

1. INTRODUCTION

Loosely speaking, we would define data mining as the process of extracting patterns present in data that are ideally interesting to a user, the data miner. There has been considerable debate about the difference and added value of data mining when compared to related disciplines. We believe the distinctiveness of the field of data mining is hidden in the word *interesting*: at its heart is the task of quantifying how interesting patterns are. This has proved to be a hard problem, and attempts to solve it wholly or partially are numerous. For example, the number of clustering objectives is practically uncountable. The number of measures of interest used in frequent pattern mining and association analysis is in the order of a hundred. Even in a setting as well-defined as supervised classification, the accuracy objective can be meaningfully quantified in a large number of ways. The practitioner trying to find her way in this jungle is not to be envied.

Partly in recognition of this problem, and to define the boundaries of this relatively young field, efforts have been made to come up with theoretical frameworks for data mining. A very insightful paper [22] on this topic suggested a list of properties such a theoretical framework for data mining should satisfy. In particular, the paper argues it must encompass all or most typical data mining tasks, have a probabilistic nature, be able to talk about inductive generalizations, deal with different types of data, recognize the data mining process as an interactive and iterative process, and account for background knowledge in deciding what is an interesting discovery.

In [22] various attempts at achieving subsets of these criteria are surveyed, such as reducing data mining to multivariate statistics or to machine learning; regarding it as a probabilistic modeling or as a compression effort; formalizing it from a microeconomic viewpoint with an externally defined utility function; or regarding data mining research as the development of a query language on so-called inductive databases.

The purpose of this paper is to suggest a new framework that usefully defines the boundaries of data mining as different from statistics, machine learning, and probabilistic

modeling.[1] In particular, the subject of our framework is data mining as defined as follows.

DEFINITION 1.1 (DATA MINING). *Data mining is a process of information transmission from an algorithm (called the data mining algorithm) that has access to data, to a data miner who is interested in understanding the data but who has an information processing capacity that is too limited to access it directly. The information transmitted is in the form of patterns of a type or syntax specified by the data miner. Here,* pattern *is broadly defined, as a quantifiable property of the data. The transmission of a pattern results in a reduction of the data miner's uncertainty and misconceptions about the data, and the stronger this reduction, the more interesting the pattern is deemed to be. The data mining process can be iterative, such that patterns are communicated one by one. It can also be interactive, allowing the data miner to influence the data mining algorithm in the course of the process, e.g. steering it toward patterns of a certain syntax or complexity.*

From this definition, it should be clear that we focus on the data miner as much as on the data. Indeed, we contend that *postulating* which patterns are interesting and which ones are not in an 'objective' way, as is common practice, is bound to be of limited practical use. Instead, we believe that what makes a pattern interesting should be part of the object of study of data mining research. The framework we propose in this paper will serve that goal: studying the interaction between the data and the data miner in order to quantify the subjective level of interest in a pattern (see [28, 25] for remarkably early and significant steps in this direction).

Before we embark on the details of our framework for data mining defined in this way, let us point out what our framework is not, and which questions it does not attempt to answer. In doing this, we contrast our framework with the possible frameworks considered in [22].

Our setting is distinctly different from the machine learning setting, in that *interesting* patterns in data may allow one to *predict*, but do not need to do so. To derive PAC-style learning theoretic results [30], statistical assumptions about the source of the data need to be made. As this is done adequately in the machine learning literature, we would rather not make these assumptions and leave it to machine learning studies to do this. Even stronger, in the general setting we regard *the data* as a monolithic entity, not necessarily composed of similar constituent parts typically called *data points*. This is important e.g. in relational data mining (if one wants to avoid using techniques such as propositionalization), where such a reduction is often nontrivial and not unique.

Similarly, our framework is philosophically different from probabilistic modeling. In particular, we consider situations where the data miner's primary concern is understanding the data itself, rather than the stochastic source that generated it. Whenever we consider a probabilistic model for the data, this represents the data miner's belief about what the value of the data may be, rather than a probability distribution from which the data might have been sampled.

Our formalism also differs in important respects from the view of data mining as compression [10, 27]. This popular and powerful view suffers from a few problems. A theoretical problem is its philosophical dependence on the Kolmogorov complexity of the data, which is uncomputable. A more serious practical problem is that, at least in its common form, it is ignorant of the user or her prior information or beliefs about the data.

The microeconomic view of data mining [19] is highly dissimilar but complementary to our framework. It is useful in different settings, namely when the data mining utility function can be formally specified. However, we contend that in many cases it is impractical to specify such a utility function.

Also the inductive database view (e.g. [21, 26]) could possibly be united and lead to synergies with our framework. However, we will not utilize the terminology common in that area of research, as we are not concerned with the inductive learning aspects, nor the query aspects, as much as with the information theoretic aspects of the data mining process.

Probably the key difference of our framework with all theoretical frameworks surveyed by [22] is its primary focus on the user, rather than on the data. Some early papers have taken a similar approach, e.g. [28, 25], but unfortunately the fraction of papers that aim to define interestingness in a subjective manner is small. Still, we believe this is critical to the very identity of data mining, as a field concerned with defining which patterns are interesting, and we consider it crucial to enhance the successful adoption of data mining in practice, especially of exploratory data mining techniques.

The present paper results from various ideas that were published earlier, or are currently under review [11, 12, 4, 5, 6, 8, 20, 7]. Furthermore, relations exist with various prior work [28, 25, 17, 27, 29, 24, 23, 14], but for space reasons we postpone a detailed discussion of these connections to a forthcoming paper. For information theory and information geometry references related to this paper, see e.g. [15, 3, 1].

2. THE USER'S PERSPECTIVE AND A DATA MINING FORMALISM

2.1 A bird's eye view on the framework

We regard data mining as a process that decreases a user's uncertainty or misconceptions about the value of the data $x \in \mathcal{X}$ (see Def. 1.1). This data mining process can be more or less efficient, depending on which patterns are being communicated and the prior beliefs of the data miner. In this sense, a pattern that reduces the data miner's uncertainty more strongly is subjectively more interesting. A central component of our framework is thus a representation of the uncertainty of the data miner about the data (as believed by the data miner).

In this initial simplified bird's eye view, let us assume that the data miner is able to formalize her beliefs in a *background distribution*, denoted P^*. Her interest is to get to know the data x in as cheap a way as possible. A simple approach could be to agree a code with the algorithm, and request the entire data to be sent over encoded in this way. For

[1]Note that the purpose of this paper is fundamentally different from the purpose of [9], which introduces a different kind of framework for data mining. Whereas the purpose of [9] appears to be the establishment of a common taxonomy and language to formalize data mining processes as they are usually implemented today, the purpose of the current paper is to propose a framework for how data mining could and perhaps sometimes should be done differently from this standard practice.

convenience, we will assume the code is an optimal code where Kraft's inequality is an equality, which means that the code lengths can be specified by means of a probability distribution P as $-\log(P(x))$ for data x. We will also refer to such code as a Shannon code with respect to P. It is in the data miner's interest to design the code lengths in this code (or equivalently the probability distribution P) so as to minimize her expectation of the code length, equal to $E_{X\sim P^*}\{-\log(P(X))\}$. It is easy to show that this function is minimized for $P = P^*$, and the expected code length is equal to the entropy $H(P^*) = -E_{X\sim P^*}\{\log(P^*(X))\}$ of P^*. Note that a small entropy does not guarantee that the code length $-\log(P^*(x))$ for the actual data x is small: this depends on the accuracy of the prior beliefs as formalized in P^*. The entropy of P^* could be small due to the data miner being overly confident.

The result of the communication of a pattern to the data miner is that she will update her beliefs about the data, resulting in a new background distribution $P^{*\prime}$. Correspondingly, to ensure efficient communication of the data, the data miner may update the code to be used for communicating the data such that the new code word length for $x \in \mathcal{X}$ is equal to $-\log(P'(x))$ for some distribution P' (again, assuming that Kraft's inequality can be an equality). A rational choice to determine these new code word lengths (or equivalently the distribution P') would be to maximize the expected reduction in code length, equal to $E_{X\sim P^{*\prime}}\{\log(P'(X)) - \log(P^*(X))\}$. It is easy to show that the maximum is achieved for $P' = P^{*\prime}$, and the expected reduction in code length is then equal to the *Kullback-Leibler (KL) divergence* $KL(P^{*\prime}||P^*) = E_{X\sim P^{*\prime}}\{\log(P^{*\prime}(X)) - \log(P^*(X))\}$ between the initial and updated background distributions P^* and $P^{*\prime}$. It is quite fitting that the KL-divergence is also known as the *information gain*.

Then, good data mining algorithms are those that are able to pinpoint those patterns that lead to a large information gain.[2] Simultaneously, an interesting pattern needs to be easy to describe, itself. Describing a pattern requires a code as well, which should also be defined by the data miner. This code provides the data miner with a way to steer the search toward patterns of a certain syntactical complexity or form. It is then the trade-off between the information gain due to the revealing of a pattern in the data, and the description length of the pattern, that should define a pattern's interest to the data miner.

We have not touched upon how the background distributions P^* and $P^{*\prime}$ are determined, or how they become known to the algorithm so that it could work out the Shannon code lengths. Typically, the data miner will not be able to specify these distributions explicitly, as it would be too laborious or the introspective exercise may be prohibitively hard. In the rest of this Section we will discuss how this can be handled, and we will further detail some of the other aspects of the framework.

2.2 The data miner and prior beliefs

We assume that a data miner is interested in getting to know as much as possible about the data at hand, at as little a description cost as possible. If the data miner did not have prior beliefs, this would amount to data compression, e.g. using a universal compression scheme (such as Lempel-Ziv). See e.g. [27, 10] for a discussion of the compression view.

Prior beliefs allow one to specify a compression scheme that reflects the data's complexity as perceived by the data miner. Let us assume that the prior beliefs of the data miner can be summarized using a background distribution P^*. Then we have argued that the optimal code to be used for communicating the data has code word lengths equal to $-\log(P^*(x))$ for $x \in \mathcal{X}$. The more probable the data miner judges the data to be, the shorter the code describing it would be. The code length for x is effectively a measure of the subjective complexity of the data x.

Ideally, the data miner directly formalizes her prior beliefs by specifying P^*. However, while we postulate the existence of P^* somewhere deep in the mind of the data miner, requiring her to fully specify it is bound to be impractical. More manageable would be to allow the data miner to just specify (some of) her beliefs in the form of *constraints* the background distribution must satisfy, or equivalently a set $\mathcal{P}$ of distributions P^* must belong to. For example, the data miner could state that the expected value of a certain property of the data is equal to a given value. The resulting $\mathcal{P}$ would be a convex set, as we will assume in the sequel.

Not knowing the exact background distribution, it is unclear then how the code for the data should be designed, i.e. how to choose the distribution P that defines the code lengths as $-\log(P(x))$ for $x \in \mathcal{X}$. We argue it makes most sense to choose the distribution $P \in \mathcal{P}$ of maximum entropy, and we call this the ME (Maximum Entropy) distribution, i.e.:

$$P \quad = \quad \arg\max_{P\in\mathcal{P}} -E_{X\sim P}\{\log(P(X))\}.$$

It should be acknowledged that typically $P \neq P^*$. However, as we will argue in two different ways, P is a good surrogate for P^* for the design of a code for the data.

The first argument is that in the absence of any other constraints, the best estimate for P^* is the ME distribution from $\mathcal{P}$. The motivation of this Maximum Entropy Principle is that any distribution from $\mathcal{P}$ with lower entropy than the ME distribution effectively injects additional knowledge, reducing the uncertainty in undue ways. After estimating the background distribution as the ME distribution P, we can repeat the reasoning from Sec. 2.1 and set the code lengths equal to $-\log(P(x))$ in order to minimize (an estimate of) the expected code length.

The second argument is a game-theoretic one. For a Shannon code with respect to P, the expected code length according to P^* is equal to $-E_{X\sim P^*}\{\log(P(X))\}$. Of course, this quantity cannot be evaluated when P^* is incompletely specified. As fully specifying it is impractical, the data miner and the data mining algorithm could instead agree to play safe and choose P so as to minimize the expected code length *in the worst case* over all possible $P^* \in \mathcal{P}$. Then, P is found as:

$$P \quad = \quad \arg\min_{P} \max_{P^*\in\mathcal{P}} -E_{X\sim P^*}\{\log(P(X))\}.$$

It is well-known that for convex $\mathcal{P}$ the solution P is equal to the ME distribution from $\mathcal{P}$. Hence, in this argument, we do not try to estimate P^*, but instead we design the code

[2]Note that ideally, rather than the *expected* reduction in code length for the data, it should be the *actual* reduction that guides the search for patterns. However, we will show below that focusing on the information gain is equivalent with focusing on the actual reduction in code length.

lengths defined by P in the safest possible way given the prior beliefs specified by the data miner.

In summary, at the start of the data mining process, the most sensible way to quantify the subjective complexity of the data is using the code length in an optimal Shannon code with respect to the ME model P. We will also refer to this distribution as the *surrogate background distribution*.

2.3 Patterns and conditioning

In a data mining process, the data mining algorithm reveals a pattern to the data miner (see Sec. 2.6 for pattern *sets*). For concreteness, let us narrow down our definition of a pattern to mean any property of the data that restricts the set of possible values of the data to a subset of its domain. Then a pattern can be formalized as a constraint $x \in \mathcal{X}'$ for some $\mathcal{X}' \subseteq \mathcal{X}$. Although not fully general, we believe that many data mining algorithms can be studied using this definition of pattern. Nevertheless, in Sec. 4.2 we will outline how the framework could be generalized to include other types of pattern as well.

The result of revealing a pattern to the data miner is that the data miner's beliefs adapt, and thus the background distribution P^* turns into an adapted background distribution $P^{*\prime}$. To exploit this, also the code describing the data should be reconsidered to minimize the expected code length.

As is the case for P^* at the start of the data mining process, the data miner cannot be expected to specify $P^{*\prime}$. All we know about it is that $P^{*\prime}(x) = 0$ for all $x \notin \mathcal{X}'$, as the pattern has revealed to the data miner that $x \in \mathcal{X}'$. To exploit this, it should be possible to update the code describing the data, making it more efficient (at least in expectation, but it would be even better if it is deterministically more efficient, irrespective of the actual value of the data).

We argue that the new code should have Shannon code lengths with respect to a distribution P' that is defined as the distribution P conditioned on the information that the data x belongs to the new support $\mathcal{X}' \subseteq \mathcal{X}$. Formally:

$$\begin{aligned} P'(x) &= P(x|x \in \mathcal{X}'), \\ &= \begin{cases} 0 & \text{for } x \notin \mathcal{X}', \\ \frac{1}{P(X \in \mathcal{X}')} \cdot P(x) & \text{for } x \in \mathcal{X}'. \end{cases} \end{aligned}$$

The reduction in code length (and thus also the reduction in expected description length, called the information gain in Sec. 2.1) of the data achieved by encoding the data w.r.t. P' instead of P is then equal to $-\log(P(X \in \mathcal{X}'))$. We refer to this quantity as the *self-information* of the pattern. In the discussion below, we will refer to the distribution P' as the *updated surrogate background distribution*.

As for choosing P as the ME distribution, there are again two arguments for choosing the updated surrogate background distribution P' as the conditional of P on the fact that $x \in \mathcal{X}'$.

In the first argument, we attempt to estimate the new background distribution $P^{*\prime}$ by P'. In the absence of any other information, it seems sensible to let P' be as similar to P as possible, while incorporating the new information that $x \in \mathcal{X}'$. This difference between the updated and original distributions P' and P can be quantified using the KL divergence $KL(P'||P) = E_{X\sim P'}\{\log(P'(X)) - \log(P(X))\}$. Thus, with $\mathcal{P}'$ the set of distributions that assign zero probability to the set of all $x \notin \mathcal{X}'$, P' can be estimated as:

$$P' = \arg\min_{P' \in \mathcal{P}'} E_{X\sim P'}\{\log(P'(X)) - \log(P(X))\}.$$

This principle of choosing P' is known as the Minimum Discrimination Information Principle, and is a generalization of the Maximum Entropy Principle. For the particular choice of $\mathcal{P}'$, it can be shown that the optimum is indeed achieved for P' as the conditioning of P on the new domain $\mathcal{X}'$, i.e. $P'(x) = P(x|x \in \mathcal{X}')$. After estimating $P^{*\prime}$ as P', it is again easy to show that the shortest code in expectation is the Shannon code with respect to P'.[3]

The second argument is again a game-theoretic one, and avoids the need to estimate the new background distribution $P^{*\prime}$. All we assume about it is that $\sum_{x\notin\mathcal{X}'} P^{*\prime}(x) = 0$, i.e. that $P^{*\prime} \in \mathcal{P}'$, as the pattern revealed to the data miner specifies that $x \in \mathcal{X}'$. We want to find a distribution P' and the associated Shannon code lengths for describing the data, for which the data miner's expectation of the reduction in description length is maximal. This expected reduction in description length is equal to the expected value with respect to $P^{*\prime}$ of the difference between the code length using P and using P', i.e. $E_{X\sim P^{*\prime}}\{\log(P'(X)) - \log(P(X))\}$. It is impossible to maximize this directly, as $P^{*\prime}$ is unspecified. However, we can maximize it in the worst-case over all $P^{*\prime} \in \mathcal{P}'$:

$$P' = \arg\max_{P'} \min_{P^{*\prime}\in\mathcal{P}'} E_{X\sim P^{*\prime}}\{\log(P'(X)) - \log(P(X))\}. \quad (1)$$

It can be shown that the optimum P' is P conditioned on the new domain $\mathcal{X}'$, as suggested (see also Sec. 4.2).

Note that the initial set of prior beliefs may not be true anymore in the updated background distribution $P^{*\prime}$, i.e. it may be that $P^{*\prime} \notin \mathcal{P}$. As a result, also the updated surrogate background distribution P' may not belong to $\mathcal{P}$. This is desirable, as the revealed patterns may help the data miner to fine-tune the prior beliefs, or even to correct outright misconceptions.

REMARK 2.1. *Note that although the description length of the data is reduced by the self-information of the pattern after conditioning the surrogate background distribution, the expected description length may actually increase. This is due to more information becoming available, such that the description length can be estimated more accurately. Thus, patterns that lead the data miner to revise her background distribution in such a way that the expected description length increases are not to be avoided for that reason. We should just be concerned by the amount by which a pattern reduces the description length of the actual data, which is its self-information.*

Data mining could thus be formalized as the process of revealing aspects of the data in the form of patterns that allow the data miner to revise her beliefs in such a way that the data becomes more plausible (i.e. less complex to describe). For patterns defined as constraints of the form $x \in \mathcal{X}'$ for some $\mathcal{X}' \subseteq \mathcal{X}$, this means reducing the set of possible values of the data to a subset of the domain of the data. In doing so, the remaining uncertainty of the data miner is reduced, as reflected in a reduction in the data's description length. Without knowing the data x it would

[3]Note that $KL(P'||P)$ is then equal to the expected difference in coding length using a Shannon code based on P and a Shannon code based on P', where the expectation is taken with respect to the new estimated P'. This should corroborate the choice for the KL-divergence as a way to quantify the difference between P and P'.

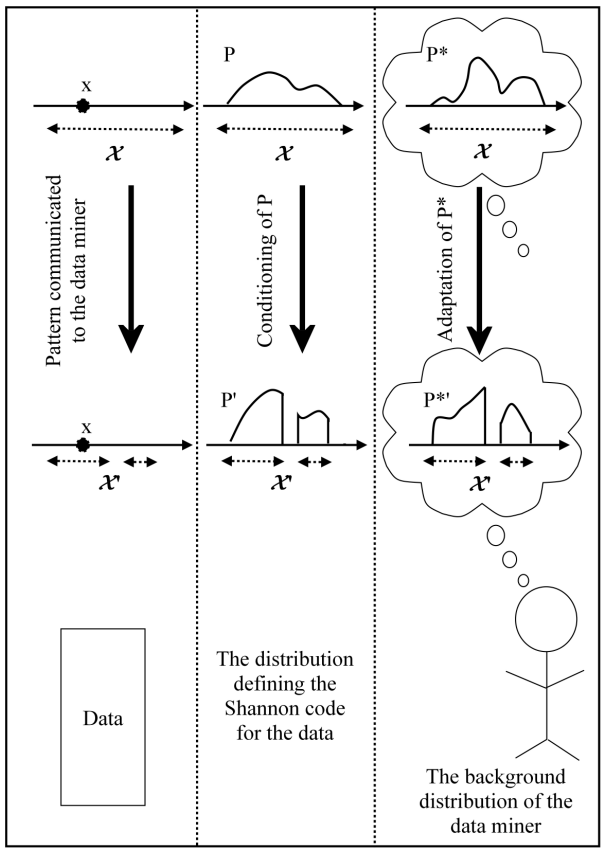

Figure 1: A schematic illustration of the effect of the communication of a pattern reducing the set of possible values for x from $\mathcal{X}$ to $\mathcal{X}' \subseteq \mathcal{X}$. The surrogate background distribution P used to determine the code lengths is conditioned on this new reduced domain, leading to an updated surrogate background distribution P' and associated set of code lengths. The unknown new background distribution $P^{*\prime}$ could be anything consistent with the requirement that it assigns zero probability to $\mathcal{X}\backslash\mathcal{X}'$.

seem we can only reason about the information gain due to a pattern, i.e. the *expected* reduction in code length. However, it turns out that by defining P' as the conditional of P on the presence of the revealed pattern, the reduction in description length is independent of the value of the data x, and equal to the self-information of the revealed pattern. Patterns can then be considered more interesting if their self-information is larger. This is equivalent to what we suggested in Sec. 2.1, considering patterns more interesting if they lead to a larger information gain.

A schematic illustration of the essential components of the data mining process is given in Fig. 1. This Figure shows what may happen to the true (unknown) background background distribution when it is revealed that the data x belongs to $\mathcal{X}' \subseteq \mathcal{X}$. In particular, all $P^{*\prime}$ needs to satisfy is that its domain is restricted to $\mathcal{X}'$. Also shown is how P', which defines the code lengths for the data after observing the pattern, is obtained from the ME distribution P, simply conditioning it on the reduced domain $\mathcal{X}'$.

2.4 The cost of a pattern

Instead of just being concerned with encoding the data given that a pattern is known to be present, we should be concerned with the length of the joint description of the data and the pattern. Indeed, also with the communication of a pattern to the data miner a description cost is associated.

The code length of each possible pattern should be specified in advance by the data miner. It should reflect the subjective complexity of a pattern, and can be used to favour patterns of a certain syntax or complexity over others by assigning them a smaller code word.

2.5 Effective data mining as a specific type of compression, and subjective level of interest of a pattern

The above completes the essence of our framework for data mining. Given this framework, we can reason about the efficiency of the process. In particular, we believe it should be the goal of a data mining algorithm to search for the pattern that strikes the right balance between the self-information (and hence the reduction in description length of the data) on the one hand, and the description cost for the pattern on the other. There may be various ways to trade-off these two aspects.

For example, assume that the data miner has a certain processing capacity, defined as an upper bound on the pattern description length she can cope with. Then the level of interest of the pattern should be defined by its self-information, as long as its description length under the user-specified code (see Sec. 2.4) is smaller than that upper bound.

On the other hand, if the data miner is truly interested in fully knowing the data (which we believe is less common), it is the overall compression of the data that is of interest. Then the measure of interest should be defined by the reduction in overall coding length achieved, taking into account the need to also encode the pattern. I.e., then the difference between the self-information (the gain in the description length of the data) and the description length of the pattern would be of interest, as it represents the gain in total description length.

2.6 Pattern set mining, iterative data mining, and interactive data mining

In the above, we have assumed that only one pattern is revealed to the data miner. Here, a pattern was defined as anything that reduces the set of possible values for the data from the entire domain $\mathcal{X}$ down to $\mathcal{X}'_i \subseteq \mathcal{X}$. Note that we introduced a subscript i here, which is an index running over the set of patterns of potential interest to the data miner (i.e. those the data miner assigned a finite code word length). Obviously, this implies that a *pattern set* is a pattern itself—for the discussion below let us call this a composite pattern (even though this is a rather badly defined notion). Indeed, a composite pattern defined by a set of patterns with indices i in a set $\mathcal{I}$, corresponds to a constraint $x \in \mathcal{X}'_{\mathcal{I}}$ with $\mathcal{X}'_{\mathcal{I}} = \bigcap_{i\in I} \mathcal{X}'_i$. Clearly, since the intersection operation is commutative and associative, it does not matter in which order patterns are revealed.

Note that the conditioning operator applied to distributions is commutative and associative as well. This means

that the conditioning of the surrogate background distribution based on a composite pattern is the same as the iterative conditioning on the atomic (as opposed to composite) constituent patterns. The framework thus describes pattern set mining in a sound and natural way. Patterns can be combined, leading to the definition of new (composite) patterns.

As such, the notion of a pattern set becomes theoretically less relevant as it can be regarded as a single pattern of different kind. Indeed, the aim of iterative data mining is simply to search for a single composite pattern with a maximal self-information $-\log(P(X \in \mathcal{X}'_{\mathcal{I}}))$, while its description should not be too long. Here, the description length of a composite pattern is the sum of the description lengths of the atomic constituent patterns.

However, from a practical algorithmic perspective, *pattern set mining* remains important as a concept. Interestingly, pattern set mining formalized in this way always boils down to a *weighted set coverage problem* of some kind. Indeed, consider the domain $\mathcal{X}$ of the data as the universe, each element $x \in \mathcal{X}$ of which is assigned a weight $P(x)$ equal to the probability under the initial surrogate background distribution. Furthermore, associate with each pattern a subset of this universe, namely the set $\mathcal{X}\backslash\mathcal{X}'_i$ of elements from the data domain it rides out. Also associate a cost with each pattern, equal to its description length. Then, the search for a pattern set with indices $\mathcal{I}$ such that the union of subsets from the data space associated to the patterns in the pattern set have the largest cumulative weight is equivalent with searching for the pattern set for which $\bigcup_{i\in I} \mathcal{X}\backslash\mathcal{X}'_i = \mathcal{X}\backslash\bigcap_{i\in I} \mathcal{X}'_i = \mathcal{X}\backslash\mathcal{X}'_{\mathcal{I}}$ has the largest total probability under P. This in turn is equivalent to searching for the pattern set with indices $\mathcal{I}$ for which $\mathcal{X}'_{\mathcal{I}}$ has the smallest total probability, i.e. for which $P(X \in \mathcal{X}'_{\mathcal{I}})$ is minimal, or hence for which self-information $-\log(P(X \in \mathcal{X}'_{\mathcal{I}}))$ is maximal, as we aim to do in pattern set mining under our framework.

This shows that pattern set mining can be reduced to a weighted set coverage problem. With an additional upper bound on the overall description length of the pattern set, it is a budgeted version of the weighted set coverage problem. These problems are known to be hard to solve exactly, but can be approximated well using an iterative greedy search approach [18]. In the present context, this amounts to using *iterative data mining* approaches to pattern set mining.

Ideally, the iterative data mining process should be *interactive*, allowing the data miner to change her 'prior' beliefs (a specific pattern may lead to a more global insight or belief), or to change the focus of the search. Both can easily be modeled in our framework: the former by updating the background model (e.g. using I-projections, see below), the latter by adapting the description lengths assigned to the atomic patterns in the course of the iterative data mining process.

2.7 Practical aspects

To allow this framework to be practically useful, algorithmic challenges will need to be overcome. We will touch upon this very briefly in the context of some of the special cases discussed below.

More fundamentally, the framework would loose most of its appeal if mechanisms to allow the data miner to specify prior beliefs, or coding lengths for patterns, were impractical. We believe, however, that such mechanisms do often exist. This will also be demonstrated briefly for the special cases discussed in Sec. 3.

In particular, such prior beliefs and coding lengths can often be specified at a generic intentional level. Also, in many typical scenarios the number of prior beliefs could be very limited (e.g. just on the mean or variance of certain aspects of the data). Even when it is impractical to specify all prior beliefs, the iterative data mining approach would only initially return uninteresting patterns, until the unspecified prior beliefs have been covered by the discovered patterns.

With regards to the coding lengths the data miner needs to specify, let us point out a connection between these coding lengths and a regularization function in machine learning. Both bias the search in a certain direction. In machine learning, the regularizer has been used conveniently and effectively to bias the search toward certain solutions, e.g. to ensure sparsity.

3. SPECIAL CASES

In [22] the author argues that a framework for data mining should be able to describe important existing methods. Equally important is that it provides us with additional insights on such methods, providing an opportunity to improve their usability. Below we will argue at a high level that this is indeed the case for a number of important examples.

3.1 Clustering and alternative clustering

An example of a clustering pattern is one that specifies that the data can be clustered at a certain cost around a specified set of cluster centres. This cost could for example be the average squared distance to the nearest cluster centre, or the likelihood of the maximum likelihood Gaussian mixture distribution. Alternatively, a cluster pattern can be specified by the subset of points belonging to the cluster along with its mean. Clearly, knowing that such a pattern is present reduces the set of possible values for the data.

The degree to which a clustering is of interest to a data miner will strongly depend on prior beliefs the data miner has. For example, for data that is elongated along one dimension, K-means clustering will typically return a result that partitions the data along this dimension. If the data miner had prior knowledge about the overall data shape, this would be a pattern of little interest to the user, and indeed it would have a small self-information.

Recently there has been considerable interest in alternative clustering (see e.g. [2]). This is the task of generating a set of different clusterings, each of which reveals different aspects of the data. A key challenge here is quantifying the redundancy between two alternative clusterings. In our setting, this redundancy could be captured by computing the self-information of the composite pattern composed of the two clustering patterns, and comparing it with the self-information of the two atomic patterns for the two clusterings by themselves. If there is a small difference only, the clusterings are redundant. Alternatively, we could assess the self-information of the second clustering under the updated background model conditioned on the first clustering pattern. If the clusterings are redundant, this self-information will be small.

Note that the user can inject a preference for a specific number of clusters, by ensuring that the coding length for clustering patterns with the specified number of clusters is small.

We have not touched upon computational cost, and we acknowledge that developing practical algorithms to implement the above may be challenging for some types of patterns. Thus, as a proof of concept and a useful result in its own right, we have developed an alternative clustering algorithm based on this framework, for cluster patterns that specify the mean of a specified subset of the data points belonging to the cluster. After working out the details of the framework for this type of pattern, we end up with an algorithm that bears similarities with K-Means clustering and spectral clustering, while it is more flexible in that it can take prior beliefs on the data into account to ensure the clusters are subjectively interesting to the data miner. See [7] for more details of this application of the framework.

3.2 Dimensionality reduction

Principal Component Analysis (PCA) [16] can be described as a method implementing our framework. Imagine prior beliefs of a data miner in the form of the expected value of each of the features (i.e. the mean of all data points), and an isotropic variance of say 1. This gives a ME model equal to an isotropic multivariate normal distribution centred at the assumed data mean $\boldsymbol{\mu}$. With data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ containing the d-dimensional data points $\mathbf{x}_i$ as its n rows, the background distribution is thus:[4]

$$P(\mathbf{X}) \quad = \quad \frac{1}{\sqrt{(2\pi)^{dn}}} \exp\left(-\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x_i}-\boldsymbol{\mu})'(\mathbf{x}_i-\boldsymbol{\mu})\right).$$

In PCA, the type of patterns sought specify the values of the centered data points' projections onto a weight vector $\mathbf{w}$ (we assume, without loss of generality, that $\mathbf{w}'\mathbf{w} = 1$). I.e., such patterns are constraints of the form $(\mathbf{X}-\mathbf{1}\boldsymbol{\mu}')\mathbf{w} = \mathbf{z}$, and there is a possible pattern for each possible weight vector. Clearly, given a pattern for a certain weight vector, the set of possible values of the data is reduced. A priori, none of these patterns is to be preferred, so they should all be given the same coding cost.[5]

It can be shown that the probability density of a pattern $(\mathbf{X}-\mathbf{1}\boldsymbol{\mu}')\mathbf{w} = \mathbf{z}$ under the above background model is given by:

$$P((\mathbf{X}-\mathbf{1}\boldsymbol{\mu}')\mathbf{w} = \mathbf{z}) \quad = \quad \frac{1}{\sqrt{(2\pi)^{n}}} \exp\left(-\frac{1}{2}\mathbf{z}'\mathbf{z}\right).$$

The self-information of the pattern is thus given by:

$$-\log\left(P((\mathbf{X}-\mathbf{1}\boldsymbol{\mu}')\mathbf{w} = \mathbf{z})\right) \quad = \quad \frac{n}{2}\log(2\pi) + \frac{1}{2}\mathbf{z}'\mathbf{z}.$$

Thus, the pattern with the largest self-information is the one that maximizes the sum of squares of the projections of the centered data points onto the corresponding weight vector $\mathbf{w}$—i.e. the one that maximizes the variance of the projections (assuming mean $\boldsymbol{\mu}$) and thus the PCA objective.

A similar reasoning shows that the k patterns with weight vectors equal to the k dominant principal components form a set of k patterns with maximal self-information. Interestingly, the marginal probability densities for patterns corresponding to orthogonal weight vectors are independent, as fixing the values of the projections of all data points on one weight vector does not constrain the possible values of the projections on an orthogonal weight vector. Thus, the self-information of a composite pattern consisting of a set of PCA patterns with orthogonal weight vectors, is equal to the sum of the self-informations of the individual patterns. This is the reason that the greedy set coverage approach (i.e. selecting principal components by a method of deflation of the covariance matrix) is optimal, as is well-known.

Understanding PCA in this way may shed some new light on what PCA is doing. However, the true power is in the framework's ability to show how to adapt PCA for other settings where the prior beliefs may be different (e.g. an anisotropic covariance matrix or an altogether different distribution).

3.3 Frequent pattern mining

In previous work, we and others have already applied the ideas from the suggested framework to searching for interesting tiles and interesting sets of tiles in binary databases. We also pointed out ways to set up ME models for more complex databases and prior beliefs, including for non-binary databases. We also actually implemented the iterative greedy data mining approach, and demonstrated its use on a number of text data sets. For details we refer to the relevant papers [4, 5, 6, 8, 20].

3.4 Community detection

When networks are represented as binary adjacency matrices, community detection is very similar to frequent pattern mining, in particular to the tile mining discussed above. The presented framework may be useful in designing community detection methods that take into account prior knowledge on certain topological properties of the network, such as an a priori known cluster structure, degree distribution, etc.

3.5 Variations on the framework: subgroup discovery and supervised learning

With a minor modification, the framework can be extended to incorporate subgroup discovery (and in a sense supervised learning, although we should reemphasize that investigating generalization is not a goal of our framework).

The modification required is that the background model should be a distribution for the labels conditioned on the actual input data. The prior beliefs are beliefs on the labels conditioned on the input data, such as: "Given that x has a certain property, the expected value of y is equal to bla". Typically, patterns would be expressed in a conditional way as well, i.e. specifying the label as a function of the value of the input data (at least for a subset of the data points). Such a pattern reduces the set of possible values of the set of the labels for the given input data. The effect of observing a pattern is similar to the basic framework: it would lead to a conditioning of the surrogate background distribution.

[4]Note that this is a density function as the domain is continuous, while in the exposition of the framework we focused on distributions on discrete domains. However, the ideas carry over without serious problems.

[5]In this case, the set of patterns is uncountable. As usual in encoding real-valued data, we could therefore discretize the space, and consider a pattern for each of the discretized weight vectors.

4. BROADER CONTEXT AND GENERALIZATIONS OF THE FRAMEWORK

4.1 Significant patterns

In recent years, data mining research, and in particular (frequent) pattern mining research (with 'pattern' defined narrowly here), has made increasing use of the notion of a p-value to quantify interest to a data miner (see e.g. [13, 24, 11, 12, 14]. Here, the null hypothesis is taken to be a distribution not unlike our background distribution, representing any prior beliefs the data miner has.

It is easy to see that this approach is a special case of our framework, for a special choice of pattern, with 'pattern' defined in the general way used in the rest of this paper. In particular, choose a test statistic (e.g. the support of a specific itemset), and define a pattern as the fact that this test statistic in the given data is larger than or equal to a certain value (typically its actual support). Then the probability against the background distribution of this pattern being present in the data is, by definition, equal to the p-value associated to the value of the test statistic, with the surrogate background distribution as null hypothesis. Hence, selecting such patterns based on small p-value is equivalent to selecting them based on large self-information.

We believe it has not been fully clear how to iteratively mine for low p-value itemsets, and heuristic approaches have been developed to update the null hypothesis (surrogate background distribution) to take prior selected patterns into account (see e.g. [12]). Interestingly, our framework suggests how it should be done in a theoretically meaningful way, although it must be admitted that it does not guarantee computational feasibility.

4.2 Generalizing the framework

In motivating the choice for conditioning the surrogate background distribution to take a seen pattern into account, we made use of a game-theoretic argument, leading to Eq. (1). We considered patterns that constrain the updated background distribution $P^{*\prime}$ to the set $\mathcal{P}'$ containing distributions that assign probability zero outside some region $\mathcal{X}' \subseteq \mathcal{X}$. For such patterns, the solution of Eq. (1), i.e. the updated surrogate background distribution P', is equal to the initial surrogate background distribution P conditioned on the new domain $\mathcal{X}'$.

However, in information geometry Eq. (1) is studied more generally [15, 3, 1]. In particular, it is known that for any closed convex set $\mathcal{P}'$, the optimum P' of Eq. (1) is the I-projection of P onto $\mathcal{P}'$. Thus, our framework also applies to pattern types that constrain the updated background distribution to any closed convex set of distributions, not just to the set of distributions restricted to a specified domain.

In other words, the conditioning of P in Sec. 2.3 is a special case of an I-projection. We introduced our framework for this special case for reasons of clarity, and also because we believe it is sufficiently general to be applicable to many data mining settings. However, the more general result can be of genuine interest. For example, for computational reasons patterns could be searched for by the algorithm on a subsample of the data, such that they only have a (known) probabilistic relation with the actual data. Interestingly, if this is the case, an I-projection amounts to a Bayes update of the background distribution, rather than a conditioning. Other pattern types could be imagined, with I-projections taking into account the presence of such patterns, possibly amounting to yet different operations on the background distribution.

4.3 Alternative interpretation

We have set up the framework in terms of prior beliefs. Then, patterns are more interesting if they contrast more strongly with these prior beliefs, as quantified using simple information theoretic concepts.

An alternative interpretation of the background distribution P^* is that it models all aspects of the data the data miner is not interested in. A special case of an aspect in which a data miner is not interested is an aspect implied by a prior belief; hence, this alternative interpretation of the background model is in a sense more general.

4.4 Data mining and machine learning

To set out we strongly insisted on distinguishing data mining and machine learning, the former being concerned with identifying interesting patterns, the latter with quantifying predictive performance and designing algorithms that guarantee generalization.

That said, we contend that our framework may be easily integrated with machine learning paradigms—they would simply study different aspects of the same practical problem. For example, we have already highlighted a strong similarity between the description cost (coding length) of a pattern and a regularization cost in machine learning. In fact, all that is needed to add on a learning element would be to make additional statistical assumptions about the data, such as i.i.d.-ness.

5. CONCLUSIONS

In this paper, we introduced a general framework for data mining. It emphatically takes a user perspective, following and modeling the user's state of mind throughout the data mining process, and allowing one to guide the data mining process so as to maximally reduce the data miner's uncertainty and misconceptions about the data. The framework naturally formalizes pattern set mining, iterative data mining, as well as interactive data mining.

By taking a user perspective, fully accounting for any prior beliefs a user may have, we believe our framework is able to capture what is truly of interest to a data miner. Considering the data miner as part of the study object of data mining research, our framework allows to define subjective level of interest of patterns in a formal way.

We argued that various existing methods can be regarded as implementations of this framework, such as PCA, our prior work related to tile mining in binary databases, and our recent work on alternative clustering. We also pointed out that it could be used with great potential in new contexts, such as in subgroup discovery.

We observed that iterative data mining in this framework always amounts to a greedy algorithm for a set coverage type problem attempting to select the best pattern set. However, importantly we did not go into the algorithmic details. As such, our framework is mostly a theoretical ideal, with a few practical instances as indications of its potential. It remains to be investigated if it can be implemented efficiently for more interesting problems. This, and the further development of these ideas for various data mining problems, is subject of our current research.

Acknowledgements

The author is grateful to the anonymous reviewers for useful suggestions, and to Akis Kontonasios, Eirini Spyropoulou and the organizers and participants of the Mining Patterns and Subgroups workshop in Leiden, 2010, for many useful and insightful discussions and critiques related to this work. This work is supported by the EPSRC grant EP/G056447/1.

6. REFERENCES

[1] S.-I. Amari and H. Nagaoka. *Methods of information geometry*. The American Mathematical Society, Oxford University Press, 2000.

[2] E. Bae and J. Bailey. Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *Proceedings of the International Conference on Data Mining (ICDM)*, 2006.

[3] I. Csiszár and P. C. Shields. *Information theory and statistics: a tutorial*. now Publishers Inc., Delft, The Netherlands, 2004.

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

[5] T. De Bie. Finding interesting itemsets using a probabilistic model for binary databases. Technical report, University of Bristol TR-123930, 2009.

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

[7] T. De Bie. Subjectively interesting alternative clusters. Technical report, University of Bristol TR-133090, 2011.

[8] T. De Bie, K.-N. Kontonasios, and E. Spyropoulou. A framework for mining interesting pattern sets. *SIGKDD Explorations*, 12(2), 2010.

[9] S. Dzeroski. Towards a general framework for data mining. In *Lecture Notes in Computer Science volume 4747*, pages 259–300. springer, 2006.

[10] C. Faloutsos and V. Megalooikonomou. On data mining, compression, and kolmogorov complexity. *Data Mining and Knowledge Discovery*, 15:3–20, 2007.

[11] A. Gallo, T. De Bie, and N. Cristianini. MINI: Mining informative non-redundant itemsets. In *Proceedings of Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 2007.

[12] 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.

[13] 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):14, 2007.

[14] 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 (KDD09)*, pages 379–388, 2009.

[15] P. Harremoës and F. Topsøe. Maximum entropy fundamentals. *Entropy*, 2001.

[16] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[17] 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 (KDD04)*, pages 178–186, 2004.

[18] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70, 1999.

[19] J. Kleinberg, C. Papadimitriou, and P. Raghavan. A microeconomic view of data mining. *Data Mining and Knowledge Discovery*, 2(4), 1998.

[20] K.-N. Kontonasios and T. De Bie. An information-theoretic approach to finding informative noisy tiles in binary databases. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, 2010.

[21] H. Mannila. Inductive databases and condensed representations for data mining. In *Proceedings of the 1997 International Symposium on Logic Programming*, pages 21–30, 1997.

[22] H. Mannila. Theoretical frameworks for data mining. *SIGKDD Explorations*, 2000.

[23] H. Mannila. Randomization techniques for data mining methods. In *Proc. of the 12th East European Conference on Advances in Databases and Information Systems (ADBIS08)*, page 1, 2008.

[24] 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 (SDM08)*, pages 494–505, 2008.

[25] 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 (KDD98)*, pages 94–100, 1998.

[26] L. D. Raedt. A perspective on inductive databases. *SIGKDD Explorations*, 4(2):69–77, 2002.

[27] A. Siebes, J. Vreeken, and M. van Leeuwen. Item sets that compress. In *SIAM Conference on Data Mining*, 2006.

[28] 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 (KDD95)*, pages 275–281, 1995.

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

[30] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27, 1984.