1. Introduction

In recent years many papers have been publish introducing new algorithms and strategies on how to mine data. In this paper, the focus is on the data miner himself. What is his motivation? What are his goals? What are his initial beliefs about the data?

The author models the data miner’s state of mind as a probability distribution, in order to represent the uncertainty and misconception he might have on the data. This probability function might change, depending on the patterns recognised during the mining process. Data mining can therefore be defined as the process of extracting patterns present in data, that are interesting to the user. This definition is flawed though, since it includes the word ”interesting”, which creates a new topic of discussion: What patterns can be classified as ”interesting”? This has proven to be a hard problem, and attempts to solve it wholly or partially are numerous. To adress this problem, the author has come up a with a theoretical framework for data mining. Such a framework must fulfill various criteria, such as:

- encompassing most typical data mining tasks
- having a probabilistic nature
- being able to talk about inductive generalizations
- dealing with different types of data
- recognising the data mining process as an interactive and iterative process
- accounting for background knowledge in deciding what an interesting discovery is

The author then gives a more precise definition of what data mining is about:
"Definition (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 proposed framework should study the interaction between the data and the data miner in order to define what an interesting pattern is. Particularly, the data is seen as a monolithic entity, not a set of data points.
2. Examples

Let’s say we are studying the effects of an unhealthy way of life on one’s life expectancy. Assume we have a table with the following five columns containing the average values through a person’s life:

1. Numbers of cigarettes smoked;
2. Liters of alcohol consumed;
3. Body Mass Index (BMI);
4. Gramms of hard drugs consumed;
5. Age at death;

Naturally one would assume that a high number in the first 4 columns would significantly lower the expected value in the fifth. This is our initial belief. One would think that a pattern that supports this theory is classified as interesting. However, there might be other patterns present that are more interesting, since they change our view of the data. E.g. a high value in column 2 could indicate a high value in column 4, the reason being that alcohol is usually consumed in high amounts at bars and nightclubs, where chances of being offered hard drugs is significantly higher. This correlation was not included in our initial statement and therefore the pattern might be more informative. Additionally, describing the exact mathematical influence the different columns have on each other under a certain pattern might quite difficult. The data miner has to find a compromise between the mathematical complexity of a pattern and the amount of information it carries.

Another example might be the following. You are given the task of encoding a book written in English. You initially encode the book assuming that Z is the most common letter, so Z will be compressed the most. This is obviously going to yield a bad result. In order to improve, you compute a new, hopefully much better P*, e.g. the distribution with A being the most common letter. This is going be a lot better, and step by step you get closer to the truth, namely the distribution which yields E as the most common letter.
3. A more abstract View

3.1 The User’s Perspective

At the beginning, the miner has a given uncertainty about the value of the data $x \in X$. The stronger a pattern reduces this uncertainty, the more interesting it is. The data miner’s initial beliefs are formalized in the background, denoted $P^*$. The goal is to get to know the data $x$ as cheap as possible. A first approach would be to agree a code with the algorithm and send the entire encoded data this way. We assume the code is optimal and therefore Kraft’s inequality is actually an equality. Therefore the code lengths under the distribution $P$ can be specified by: $-\log(P(x))$ for some data $x$. It is in the miner’s interest to minimize the expectation of the code length, which is $E_{X \sim P^*} \{-\log(P(X))\}$. The function is minimized for $P = P^*$ and the expected length is equal to the entropy $H(P^*) = -E_{X \sim P^*} \{\log(P^*(X))\}$. A small entropy does not guarantee a small code length.

The effect of the communication of a pattern to the data miner is that he will now update his beliefs about the data, which results in a new background distribution $P^*’$. He also updates the code used for communicating the data such that the new code length is $-\log(P’(x))$ for $x \in X$. He then attempts to maximize the expected reduction in code length, equal to $E_{X \sim P^*’} \{\log(P’(X)) - \log(P^*(X))\}$. The maximum is achieved for $P’ = P^*’$, and the expected reduction is equal to the Kullback-Leibler divergence $\text{KLD}(P^*’ || P^*)$, which is also known as the information gain.
An illustration of the effect of the communication of a pattern reducing the set of possible values for $x$. A good pattern must therefore offer two qualities: allowing a large gain of information while still being easy to describe.
3.2 Unspecified P*

Under some conditions it might be hard for the data miner to formally specify the initial beliefs P*. More manageable would be to allow the background distribution to be bound by a set of constraints it must satisfy, or equivalently a set Ω of distributions P* must belong to. Going back to the second example. Let’s now assume that the book is not written in English, but in Italian. Instead of just going with the assumption that e is most common letter, we can give a set of assumptions, e.g. all distributions where A, E, I, O, or U are the most common letter. The reason being that the most common letter is usually a vowel. That example should illustrate the idea behind this extension. But not knowing the actual P* causes us a number of problems. For example, how do we chose the P in -log(P(x))? The answer is, of all the distributions in Ω, we take the one with the largest entropy, which we defined in section 3.1.

\[ P = \arg \max_{P \in \Omega} -E_{X \sim P^*} \{-\log(P(X))\} \]

Why is this a good idea? There are 2 arguments that support this choice of P. The first argument is, that any distribution that has not the maximum entropy effectively injects additional knowledge, reducing the uncertainty in undue ways, and we would like to avoid this. We want to reduce the uncertainty by finding patterns, and not by taking wild guesses. Second, since there are many possible P*, it therefore makes sense to optimise for the worst case scenario, i.e.

\[ P = \arg \min_P \max_{P^* \in \Omega} -E_{X \sim P^*} \{-\log(P(X))\} \]

It has been proven that the solution to this equation is the Maximum Entropy distribution in Ω. Typically the found P is not equal to P*.

We can use them exact same reasoning when trying to improve P. We define Ω’ as the set of distributions that assign zero probability to the set of all \( x \notin X' \). This leads to following equation for P’, which looks very familiar:

\[ P' = \arg \max_{P' \in \Omega'} -E_{X \sim P'} \{\log(P'(X)) - \log(P(X))\} \]

It’s interesting to note that P*’ might no longer belong to our initial set of beliefs, because \( \Omega \neq \Omega' \). This is actually a very desirable property, since it allows us to correct any misconceptions we might have had about the data in the beginning.
3.3 Patterns

Since the term "pattern" is key in this paper, it should be defined formally: A pattern is a constrain \( x \in X' \) for \( X' \subseteq X \). This definition is sufficient for now, but can easily be extended.

Consider now the situation after the data miner has received his first feedback and adapted his initial beliefs \( P^* \) to \( P'^* \). It holds that \( P'^*(x) = 0 \) for all \( x \notin X' \), as the pattern has revealed that \( x \in X' \). The data miner can use this information to further improve the describing code. The new code should have Shannon code lengths with respect to a distribution \( P' \) under the assumption that \( x \) belongs to \( X' \subseteq X \), therefore is \( P'(x) = P(x|x \in X') \)

The reduction in code is then \(-\log(P(X \in X'))\). This quantity is defined as the self-information of the pattern \( X' \).

Note that every pattern has to be encoded, which also takes up some space. So we are not just minimizing the length of code, but the sum of code length plus pattern description. The length of a pattern is specified by the miner in advance, which gives him the opportunity to steer the mining process in a certain direction, by assigning very short lengths to these patterns.
4. Conclusion

To summarize, for every pattern we detect, we can reduce the uncertainty we have about the data. The reduction in code length is equal to the self information of the revealed pattern. Therefore, patterns which carry a high amount of self information are the most interesting ones. This is equal to finding a $P^*$ such that the KLD of $P^*$ and $P^*$' is maximum.

We also have seen how to choose a good surrogate distribution $P$ if $P^*$ is not available. It is the Maximum Entropy distribution of $\Omega$, which is the set $P^*$ has to belong to.

Finally we showed how to prioritize certain patterns by assigning them a short encoding- length and thus head our search in a favorable direction.