Finding High Order Dependencies in Data

Rosa Meo\(^1\) and Leonardo D’Ambrosi\(^2\)

\(^1\) University of Torino, Italy
\(^2\) Regional Agency for Health Care Services - A.Re.S.S. Piemonte, Italy

Abstract. We propose \textit{DepMiner}, a method implementing a simple but effective model for the evaluation of the high-order dependencies in a set \(S\) of observations. \(S\) can be either ordered - thus forming a sequence of events - or not. \textit{DepMiner} is based on \(\Delta\), a measure of the degree of surprise of \(S\) based on the departure of the probability of \(S\) from a referential probability estimated in the condition of maximum entropy.

The method is powerful: at the same time it detects significant positive dependencies as well as negative ones suitable to identify rare events. The system returns the patterns ranked by \(\Delta\); they are guaranteed to be statistically significant and their number results reduced in comparison with other methods.

1 Introduction

In statistics, machine learning and data mining the problem of the determination of set of variables whose values are correlated represents an important knowledge in many fields such as in feature selection, database design, market basket analysis, information retrieval, machine translation, biology, etc. Often in the scientific literature, the study of the dependence between variables is limited to pairs \([8]\) but finding correlations among more than two variables is essential for many commercial and sociological studies (collaborations and interaction networks), bio-medical (interaction among drugs and proteins) and scientific domains.

Furthermore, many observations have a special meaning if considered as sequentially ordered: examples span from the nucleotide sequences of the organisms genome to the series of events generated in a network of interactions (like a social network, a hypertext or a computer network), to the analysis of web server logs or the detection of intrusion attempts in information systems.

Formally, a set of observations \(I = \{i_1, i_2, \ldots, i_k\}\) is named \textit{itemset} where all the elements or items of the set occur together in the same example (or transaction). Here the term transaction denotes a group of events occurring at the same time-stamp and sharing the same order number. Any single observation or item \(i_j\) represents a pair \((\text{Variable}, \text{Value})\) that means that the \textit{Variable} describing one of the attributes of the observation assumes the indicated value. The set of items \(I\) is defined as a non-empty subset of elements from a language \(\mathcal{L}\), the item collection. In this manner, an itemset represents the set of items that occur together and have the same order number.

A sequence \(S\) is an ordered list of itemsets \(S = \langle I_1, I_2, \ldots, I_n \rangle\). A sub-sequence \(S'\) of \(S\), constituted by \(m < n\) itemsets, is denoted by \(S' \subset S\) and is constituted by
a subset of the itemsets of $S$ in which the precedence relationship between any pair of
itemsets is maintained.

In this paper we deal with both itemsets and sequences of itemsets and we are able
to model both of them with the same probabilistic model. We treat an itemset as a
composite event and we represent it in our model by the joint probability that all its
items occur together. Our model is based on the probability of occurrence of events.
We model the probability of the sequences by means of the probability of the events
represented by the sequence members. For the itemsets we record the frequency with
each all the items occur together; for sequences, we record the frequency with which
each of the event sequence occurs in the correct order. In order to unify the two concepts
of itemsets and sequences we use the common term pattern. In practical cases, often
it happens that the set of returned patterns is large and much of the information is
redundant because many of the patterns are returned together with some of their subsets.
The reduction of the redundancy in the result set answers to two major challenges:
reducing the overwhelming size of the mining results and eliminating redundancy in
the information content and the overlapping between the patterns.

Deciding which patterns are redundant is not straightforward. It might depend on
the applications. For instance, the inclusion of patterns with some common elements
could be acceptable because the itemsets might have different meaning or occur in dif-
ferent situations. Instead, the inclusion in the result of both subsets and their super-sets
is not acceptable if the super-sets do not add new information to the information car-
ried by the subsets. In literature, redundant itemsets are detected in many different and
sometimes opposite ways. [3] allows the reduction of the number of the itemsets with
a loss-less compression of the result because all the information lacking from the result
can be restored by application of the concept of closed itemsets. [12] considers the cor-
relation among the items as strong only when all the items are considered together and
when their subsets have instead a weak correlation. On the opposite side, [5] considers
interesting an itemset if all its subsets are closely related to all other subsets.

In data mining there exist computationally efficient methods to discover significant
dependencies and correlations among the items of a frequent pattern [1, 2, 10]. In item-
sets mining, in order to determine the dependencies in patterns composed by more than
2 items, either they make the multi-way independence assumption or they evaluate the
contribution to the overall itemset of each variable separately [6, 11, 12]. The difficulty
stems from the fact that there is not an easy way to determine a referential probability
of an itemset $I$ that represents a condition of independence among the subsets if we do
not suppose independence among all the single variables in $I$. [6] ranks the frequent
itemsets according to the unlikelihood that they appear under the hypothesis that all the
items in the itemset are independent. But the multi-way independence condition gives a
problem: according to this definition of independence, if a dependence already exists in
a subset of $I$, this dependence is “inherited” from the subset to $I$ and to all the super-sets
of $I$ [2]. Thus we do not have a way to distinguish if an intrinsic dependence exists in
an itemset $I$ in addition to the dependencies inherited from its subsets.

We can solve the problem in terms of quantity of information that an itemset pro-
vides. [4] proposes the use of Minimum Description length to identify the interesting
sequences. Similarly, we are interested in identifying the surprising patterns whose
probability, as derived by the number of their observations, has a great departure w.r.t. a referential, estimated probability as determined only by its subsets. We are interested in patterns that add any information to their subsets while we want to discard those itemsets that can be foreseen given the observation of their subsets. We proposed in [9] a solution based on the maximum entropy. The entropy of an itemset $I$ is computed by an estimation of the probability of $I$ computed on the basis of the probability of its subsets. The probability of $I$ at which the entropy is maximum (denoted by $P_E(I)$) corresponds to the probability that the itemset $I$ would have in the condition in which it carries the maximum amount of information in addition to its subsets. The interest measure that we proposed for an itemset $I$ is the departure of the probability of $I$ w.r.t. the referential value computed at maximum entropy: $\Delta(I) = P(I) - P_E(I)$. The higher the departure between the two probabilities, the less the itemset can be correctly foreseen from the observation of its subsets. This departure identifies a dependence between the items and tells us that this dependence is not due to the subsets only. As a consequence the itemset is non-redundant.

$\Delta(I)$ decreases with the increase in the cardinality of itemsets and it is not suitable for the comparison of itemsets of different cardinality. For this purpose in this paper we propose $\Delta_n$, a version of $\Delta$ normalized w.r.t. the probability of the itemset:

$$\Delta_n(I) = \frac{P(I) - P_E(I)}{P(I)}$$

Similarly it happens with a sequence pattern $S$. Its probability is computed on the basis of the occurrence of all its sequence elements (itemsets) when these occurrences respect the correct order in the sequence.

$\Delta_n$ takes both positive and negative values, in the range from $[-\infty, 1]$. Specifically, if the value is positive, it means a positive dependence, i.e., a pattern that is more frequent than expected; if the value if negative it means a negative dependence, i.e., a pattern that occurs rarer than expected.

In this paper we present a method for the computation of the interesting and non redundant patterns based on the above observations. $\Delta_n$ is used as a score function to rank the patterns. In Section 3 we show how we succeeded to determine the significance level of $\Delta_n$ and to identify the significant patterns.

The rest of the paper is organized as follows. In Section 2 we summarize how $\Delta$ is computed. Section 3 shows how to determine the significance level of $\Delta_n$. Section 4 presents an empirical evaluation study on the results of the system on some common data-sets. Finally, Section 5 draws the conclusions.

## 2 Estimation of the referential probability

Here we generalize our model based on $\Delta$ to generic patterns, including also sequences. Consider a pattern $W = \{w_1, w_2, \ldots, w_k\}$ where $w_j$ might represent either a single item or an itemset in an ordered sequence. Henceforth, we will use the generic term pattern element.

Entropy $H(W) = -\sum P(w^+_1, w^+_2, \ldots, w^+_k) \log[P(w^+_1, w^+_2, \ldots, w^+_k)]$ with $w^+_j$ representing the element $w_j$ either affirmed (representing an event present in an observation)
or negated (absent event). Summation ranges over the probabilities of all the combinations of the $k$ elements taken affirmed or negated. $H(W)$ is not computed by assumption that singletons are independent but taking in consideration the actual probability of occurrence of each subset of $W$, as observed from the database. The exclusion-inclusion principle [3] is adopted to compute the entropy of $W$ starting from the probability of the subsets of $W$ and it is a function of the probability of $W$. The estimate of the probability of $W$ is the probability value that maximizes the entropy and corresponds to the case in which the amount of information on the presence of $W$ is minimum given the knowledge on the presence of the subsets of $W$. Notice that in making the estimate we considered only the observed probabilities of the subsets. Thus, if the dependence in a pattern $W$ is intrinsic, due to the synergy between all its elements, then the observed probability of $W$ departs with respect to the estimate. As a result, $\Delta_n(W)$ makes emerge the intrinsic, actual dependencies, existing among all the elements in $W$.

3 Setting a threshold for $\Delta$

Another problem that we have to solve is how large must be $\Delta_n$ such that a pattern is deemed significant. We use a null model in which there are not dependencies between the variables. The null model is generated empirically via a randomization of the original data-set. Randomization is generally accepted as a way to allow a statistical test on significance of results [7]. Randomization occurs by independently shuffling the variable values among the examples. As a result, the new data-set will have the same marginal probabilities of the single variables but the dependencies between them are spoiled.

In a successive step, we compute patterns both from the real and the randomized data. Next, we compute the minimum negative value of $\Delta_n$ (denoted by UB) and the maximum positive value of $\Delta_n$ (denoted by LB) in randomized data. Then, we use UB as an upper bound for rare patterns in real data and use LB as a lower bound for the frequent patterns in real data. This is a sort of statistical test on $\Delta_n$ that accepts as dependent a pattern if its $\Delta_n$ is higher (resp. lower) than the maximum (resp. minimum) $\Delta_n$ of the patterns extracted from the randomized data.

Consider the public data-set Mushroom (downloadable from the repository of UCI). After randomization, we observed the maximum value of $\Delta_n = 0.04$ and the minimum value of $\Delta_n = -0.03$. In real data, the maximum is $\Delta_n = 0.85$ and the minimum is $\Delta_n = -0.45$. Thus in Mushroom the positive dependencies are more abundant and marked than the negative dependencies. In Figure 1 we show an output of DepMiner, with the ranking of patterns (itemsets) extracted from Mushroom. In green the significant itemsets are shown (whose value of $\Delta_n$ exceeds the observed range of values in randomized data). In yellow, instead, the other itemsets. Notice that both rarer and more frequent itemsets are interesting.

4 Experimental evaluation

We run a set of experiments on 5 real data-sets (from FIMI and UCI Machine Learning repositories) and on 2 real data-sets coming from NASDAQ stock exchange index (from
### Table 1. DepMiner output: itemsets ranking with significant itemsets in green.

<table>
<thead>
<tr>
<th>Delta/Po</th>
<th>Delta</th>
<th>Freq.</th>
<th>Itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0188</td>
<td>0.0075</td>
<td>3256</td>
<td>Class=poisonous, ring-number=one, bruises?=no</td>
</tr>
<tr>
<td>0.0138</td>
<td>0.0059</td>
<td>3272</td>
<td>stalk-surface-above-ring=smooth, ring-number=one, Class=edibility</td>
</tr>
<tr>
<td>0.0121</td>
<td>0.0045</td>
<td>3032</td>
<td>stalk-surface-below-ring=smooth, ring-number=one, Class=edibility</td>
</tr>
<tr>
<td>0.0055</td>
<td>0.0023</td>
<td>3376</td>
<td>stalk-surface-above-ring=smooth, Class=edibility, gill-size=broad</td>
</tr>
<tr>
<td>0.0007</td>
<td>0.0003</td>
<td>3216</td>
<td>Class=edibility, gill-size=broad, odor=none</td>
</tr>
<tr>
<td>-0.0006</td>
<td>-0.0002</td>
<td>3128</td>
<td>bruises?=bruises, gill-space=close, stalk-surface-above-ring=smooth</td>
</tr>
<tr>
<td>-0.0014</td>
<td>-0.0006</td>
<td>3728</td>
<td>veil-color=white, Class=edibility, gill-size=broad</td>
</tr>
<tr>
<td>-0.0020</td>
<td>-0.0008</td>
<td>3128</td>
<td>veil-color=white, ring-type=pendant, gill-size=broad</td>
</tr>
<tr>
<td>-0.0026</td>
<td>-0.0013</td>
<td>3964</td>
<td>gill-attach=free, stalk-surface-above-ring=smooth, stalk-below-ring=smooth</td>
</tr>
<tr>
<td>-0.0049</td>
<td>-0.0021</td>
<td>3488</td>
<td>gill-attach=free, ring-number=one, Class=edibility</td>
</tr>
<tr>
<td>-0.0051</td>
<td>-0.0021</td>
<td>3368</td>
<td>gill-attach=free, ring-number=one, ring-type=pendant</td>
</tr>
<tr>
<td>-0.0061</td>
<td>-0.0022</td>
<td>3016</td>
<td>stalk-surface-above-ring=smooth, ring-type=pendant, gill-size=broad</td>
</tr>
</tbody>
</table>

January 2001 to May 2009) and from the Italian lottery (with data on the numbers drawn from 1939). The lottery data-set is important in order to check the behavior of $\Delta_n$ on complete random data where even the marginals were uniform.

In Table 2 we include the total number of examples, $\text{minsup}$ threshold, the total number of itemsets generated ($N$), execution times to compute $\Delta_n$ (in seconds).

### Table 2. Experimental results.

<table>
<thead>
<tr>
<th>data-set</th>
<th>$\text{minsup}$</th>
<th>itemsets (N)</th>
<th>$\text{Dep/N}$</th>
<th>$\text{SDep/N}$</th>
<th>$\text{NDI/N}$</th>
<th>time(s)</th>
<th>$\gamma(\text{DM,MINI})$</th>
<th>$\gamma(\text{RDM,RMINI})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accidents</td>
<td>35%</td>
<td>65,500</td>
<td>13.5%</td>
<td>0.1%</td>
<td>22.93%</td>
<td>4294</td>
<td>-0.84</td>
<td>0.16</td>
</tr>
<tr>
<td>Chess</td>
<td>75%</td>
<td>20,582</td>
<td>1.2%</td>
<td>0.34%</td>
<td>2.11%</td>
<td>135</td>
<td>-0.95</td>
<td>-0.91</td>
</tr>
<tr>
<td>Nasdaq</td>
<td>0.14%</td>
<td>242</td>
<td>95.8%</td>
<td>46.69%</td>
<td>100%</td>
<td>107</td>
<td>-0.05</td>
<td>0.27</td>
</tr>
<tr>
<td>Kosarak</td>
<td>1.01%</td>
<td>21,934</td>
<td>82.5%</td>
<td>10.39%</td>
<td>95.55%</td>
<td>2221</td>
<td>-0.56</td>
<td>0.28</td>
</tr>
<tr>
<td>Mushroom</td>
<td>22.15%</td>
<td>14,189</td>
<td>82.5%</td>
<td>95.55%</td>
<td>100%</td>
<td>115</td>
<td>-0.94</td>
<td>-0.29</td>
</tr>
<tr>
<td>Retail</td>
<td>4.53%</td>
<td>22,524</td>
<td>79.7%</td>
<td>5.9%</td>
<td>99.56%</td>
<td>1322</td>
<td>0.02</td>
<td>0.55</td>
</tr>
<tr>
<td>Lottery</td>
<td>0.006%</td>
<td>91,499</td>
<td>99.1%</td>
<td>0%</td>
<td>100%</td>
<td>5804</td>
<td>0.81</td>
<td>0.77</td>
</tr>
</tbody>
</table>

We performed two experiments: the first one on the compression capability and the second one on the capability of DepMiner to determine the dependencies in contrast to methods that assume the multi-way independence condition.

1. In this experiment, to be further conservative, we compare DepMiner results at many levels. We denote as itemsets clearly non independent, the itemset whose $\Delta_n \neq 0$. Thus we include in the results both these latter ones and the itemsets whose $\Delta_n$ is acceptable by the significance test on the lower and upper bounds obtained in randomized data. We include in Table 2 three ratios: the ratio between the number of itemsets with $\Delta_n \neq 0$ and N ($\text{Dep/N}$), the ratio between the significant dependencies and N ($\text{SDep/N}$) and the ratio between non derivable itemsets ($\text{NDI}$) obtained by the competitor method [3] and N ($\text{NDI/N}$). These ratios quantify the
volume of found dependencies in data and clearly demonstrate the increased ability of DepMiner to reduce redundancies than NDI.

2. In the second experiment we compare the results of DepMiner with MINI [6], in order to determine the difference between DepMiner and another method based on the multi-way independence assumption. The last columns of Table 2 report the result of a comparison between DepMiner ranking (denoted by DM) and MINI. As said, our method does not consider dependencies inherited by the subsets; it coincides with the multi-way independence assumption only for patterns with cardinality 2. In order to measure the correlation between our ranking (DepMiner) and MINI's we adopted an objective measure: $\gamma$ [8]. Since the methods differ in the referential probability estimate, $\gamma$ quantifies the effect of this difference. In Table 2 we also compared the two rankings computed on randomized data (RDM and RMINI). All the values reported by $\gamma$ denote disagreement. The amount of discrepancies decreases ($\gamma$ increases) if we move from real data to randomized data (since the high-order dependencies are spoiled during randomization). Furthermore, on complete random data (Lottery) the two methods agree.

5 Conclusions

We have presented DepMiner, a method for the extraction of significant dependencies between the values assumed by database variables. DepMiner gave good results in comparison with [6] and demonstrated a superior capability to compress results than NDI [3].

References