Parameter Tuning for Differential Mining of String Patterns

Jérémy Besson[1] Christophe Rigotti[2] Ieva Mitašiūnaitė[2]
Jean-François Boulicaut[2]
[1] Institute of Mathematics and Informatics, Vilnius, Lithuania
[2] University of Lyon, France
{Firstname.Name}@insa-lyon.fr

Abstract

Constraint-based mining has been proven to be extremely useful for supporting actionable pattern discovery. However, useful conjunctions of constraints that support domain driven mining tasks generally need to set several parameter values and how to tune these parameters remains fairly open. We study this problem for substring pattern discovery, when using a conjunction of maximal frequency, minimal frequency and size constraints. We propose a method, based on pattern space sampling, to estimate the number of patterns that satisfy such conjunctions. This permits the user to probe the parameter space in many points, and then to choose some initial promising parameter settings. Our empirical validation confirms that we efficiently obtain good approximations of the number of patterns that will be extracted.

1. Introduction

Knowledge Discovery in Databases (KDD) processes based on pattern discovery have been studied extensively the last decade (e.g., processes based on itemsets and association rules, substrings, episode rules, subtrees, subgraphs, etc.). However, tackling pattern relevancy remains a major problem. Indeed, in real-life contexts, actionable patterns are quite often hidden amongst many irrelevant ones (see, e.g., [3]). Irrelevancy must be understood in terms of both objective and subjective interestingness. For instance, most of the time, avoiding the presentation or even better the computation of known patterns for the analyst is important. A key issue is thus to take the most from domain knowledge at each phase of the KDD life cycle. Data selection and preprocessing quality heavily rely on domain knowledge (e.g., for feature selection and feature construction). The data mining phase itself has to incorporate domain knowledge if we do not want to use all our computational resources for computing a huge amount of irrelevant patterns at the price of missing the relevant ones. Last but not the least, the crucial post-processing and interpretation phase obviously need a strong involvement of domain experts. We believe that many of these problems, especially for improving the data mining and post-processing phase efficiency, can be solved by means of constraint-based data mining systems (see, e.g., [1] for a recent overview in this area).

If constraint-based data mining is the answer, we have to specify the constraints and this can be difficult. Constraints are generally boolean combinations of primitives that need to hold for parameters. When we have found a parameter setting that gives promising results, it is possible to slightly tune the values of the parameters to focus on the most interesting patterns within a typical iterative and interactive mining process. The situation is quite different at the early exploratory mining stage. Indeed, when we start such a process, we may have at hand many primitive constraints that can be combined into a mining query, and such that most of these constraints require at least one parameter value. However, we have a limited insight about the location of the promising areas in the parameter space. A common practice is to count the number of patterns obtained for a few different parameter settings to guess what could be the interesting parameter values for a deeper investigation (during which we will then look more closely at the patterns, e.g., using some interestingness measures). In simple contexts, e.g., when considering a single minimal frequency constraint, a limited number of trials may be sufficient. This is obviously not the case when considering a conjunction of primitive constraints giving rise to a large multidimensional parameter space: we cannot afford to run hundreds or thousands of experiments to probe such a space.

We consider such a parameter tuning situation for a typical constraint-based mining task that supports differential pattern discovery. In this context, important domain knowledge is incorporated by the expert when she/he prepares the two datasets to be used during the differential extraction. We focus on a conjunction of primitive constraints com-

978-0-7695-3503-6/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDMW.2008.118

monly used in differential mining of sub-string patterns and related to the search for emergent patterns [4]. This conjunction, denoted $\mathcal{C}$, is defined informally as follows: *to occur more than a given minimal number of times in a dataset* $\mathbf{r}_1$*, and to occur no more than a given maximal number of times in a second dataset* $\mathbf{r}_2$*, and to satisfy a constraint on the size of the pattern*. Such a conjunction has been shown to be useful in several contexts. For instance, let us assume that $\mathbf{r}_1$ (resp. $\mathbf{r}_2$) contains descriptions of molecules that are active (resp. inactive) against a given organism and that the language of patterns concerns linear fragments (i.e., strings) in such molecules. Then computing solutions of $\mathcal{C}$ can provide an interesting hypothesis within a drug discovery process [7]. A second typical example concerns abstractions of WWW logs $\mathbf{r}_1$ and $\mathbf{r}_2$ for different periods of time (i.e., sets of sequences of events during browsing sessions). Computing solutions of $\mathcal{C}$ can help to identify period-specific patterns (i.e., signatures) [10]. Another recent application of such a conjunction is where $\mathbf{r}_1$ (resp. $\mathbf{r}_2$) contains promoter sequences of genes that are involved (resp. not involved) in a given biological process such that solutions of $\mathcal{C}$ can suggest regulation mechanisms and new transcription factor binding sites [11].

In this paper, our main contribution is a method based on pattern space sampling (and not on data sampling) to estimate the number of patterns satisfying a user-defined constraint. The approach is simple but we demonstrate empirically that it is efficient and accurate for guessing promising parameter settings. To the best of our knowledge, such an idea has not been reported previously. Moreover, we are more flexible than methods based on a global analytical model. To support this claim, we show that it can be combined easily with three different symbol distributions. Furthermore, we believe that because of this flexibility, it has the potential for handling other pattern domains (i.e., other types of patterns and constraints), and to support the integration of various background knowledge in the estimate.

The rest of the paper is organized as follows. Section 2 specifies the mining task, and considers different parameter tuning approaches. The method to estimate the number of patterns satisfying the conjunction of constraints is described in Section 3. Section 4 presents an empirical evaluation. Related work is discussed in Section 5. Section 6 briefly concludes.

2. Tuning parameters

We consider string database mining and the context of substring pattern extraction under a typical and non trivial conjunction of primitive constraints. Let Σ be a finite alphabet, then a string ϕ over Σ is a finite sequence of symbols from Σ. The language of patterns $\mathcal{L}$ is Σ^*, i.e., the set of all strings over Σ. A string database $\mathbf{r}$ is a multi-set[1] of strings from Σ^*. The length of a string ϕ is denoted $|\phi|$, and ϕ_i represents the i^{th} symbol of ϕ. A substring ϕ' of ϕ is a sequence of contiguous symbols in ϕ, and we note $\phi' \sqsubseteq \phi$.

The primitive constraints we use to specify the string mining tasks are a minimal frequency constraint on one dataset, a maximal frequency constraint on another dataset, and a syntactic constraint. We also consider two kinds of pattern occurrences: exact occurrences and soft ones. The concept of soft occurrence is useful to handle degenerated occurrences of patterns in many real-life application domains (e.g., motif discovery in genomics, browsing patterns for WWW usage mining). Let us now define more precisely these notions and the corresponding frequency constraints.

An *exact occurrence* of a pattern ϕ is simply a substring of a string in $\mathbf{r}$ that is equal to ϕ. The *exact support* of ϕ, denoted $supp_E(\phi, \mathbf{r})$, is the number of strings in $\mathbf{r}$ that contain at least one exact occurrence of ϕ. Notice that multiple occurrences of a pattern in the same string do not change its support.

Let δ be a positive integer, then a *δ-soft occurrence* of a pattern ϕ is a substring ϕ' of a string in $\mathbf{r}$, having the same length as ϕ and such that $hamming(\phi, \phi') \leq \delta$, where $hamming(\phi, \phi')$ is the Hamming distance between ϕ and ϕ' (i.e., the number of positions where ϕ and ϕ' are different). The *δ-soft support* of ϕ is the number of strings in $\mathbf{r}$ that contain at least one δ-soft occurrence of ϕ. It is denoted $supp_S(\phi, \mathbf{r}, \delta)$.

Example 1 *If* $\mathbf{r} = \{atgcaaac, acttggac, gatagata, tgtgtgtg, gtcaactg\}$*, we have* $supp_E(gac, \mathbf{r}) = 1$ *since only string* $acttggac$ *contains* gac*. We also have* $supp_S(gac, \mathbf{r}, 1) = 4$ *because* $atgcaaac$*,* $acttggac$*,* $gatagata$ *and* $gtcaactg$ *contain some 1-soft occurrences of* gac*.*

Definition 1 (Frequency constraints) *In the case of an exact support, given a threshold value f, the minimal (resp. maximal) frequency constraint is* $MinFr(\phi, \mathbf{r}, f) \equiv supp_E(\phi, \mathbf{r}) \geq f$ *(resp.* $MaxFr(\phi, \mathbf{r}, f) \equiv supp_E(\phi, \mathbf{r}) \leq f$*). For a δ-soft support, the constraints are defined as* $MinFr(\phi, \mathbf{r}, f) \equiv supp_S(\phi, \mathbf{r}, \delta) \geq f \wedge supp_E(\phi, \mathbf{r}) \geq 1$ *and* $MaxFr(\phi, \mathbf{r}, f) \equiv supp_S(\phi, \mathbf{r}, \delta) \leq f$*.*

Notice that, in the case of the soft support, our definition of $MinFr$ enforces the presence of at least one exact occurrence, to discard patterns that only occur as degenerated instances (useful, for instance, when looking for transcription factor binding sites).

The generic conjunction of constraints considered in this paper is: $\mathcal{C} \equiv MinFr(\phi, \mathbf{r}_1, f_1) \wedge MaxFr(\phi, \mathbf{r}_2, f_2) \wedge \mathcal{C}_{synt}(\phi)$, where $\mathbf{r}_1$ and $\mathbf{r}_2$ are string databases, f_1 and f_2 are frequency thresholds, and $\mathcal{C}_{synt}$ is a syntactic constraint

[1] The dataset may contain multiple occurrences of the same string.

on the pattern. We have been using the quite simple syntactic constraint $\mathcal{C}_{synt}(\phi) \equiv |\phi| = k$, where k is a user defined size value. However, let us emphasize that our framework can easily be extended to other constraints like $\mathcal{C}_{synt}(\phi) \equiv |\phi|$ *op* k where *op* can be $\leq$ or $\geq$.

As mentioned in the introduction, this generic conjunction has been shown to be useful in many application domains: differential mining appears as a simple but powerful way to integrate domain knowledge and thus to support domain driven data mining.

In an exploratory data mining task based on pattern extraction, one of the most commonly used parameter tuning strategies, in the early exploration stage, is to run a few experiments with different settings, and to simply count the number of patterns that are obtained. Then, using some domain knowledge, the user tries to guess some potentially interesting parameter settings. After that stage, the user enters a more iterative process, in which she/he also looks at the patterns themselves and at their scores (according to various quality measures), and uses her/his knowledge of the domain to focus on some patterns and/or to change the parameters by some "local" variations of their values.

We want to support the early exploratory stage such that the user can guess promising initial parameter settings. We propose to probe the parameter space in a more systematic way, so that it could be possible to provide graphics that depict the extraction landscape, i.e., the number of patterns that will be obtained for a wide range of parameter values. This idea is very simple, and many (if not all) of the practitioners have one day written their own script/code to run such sets of experiments. However, in many cases, the cost of running real extractions for hundreds of different parameter settings is clearly prohibitive.

Instead of running real experiments, a second way is to exhibit an analytical model, that estimates the number of patterns satisfying the constraint $\mathcal{C}$, with respect to the distribution of the symbols and the structure (number of strings and size) of the datasets, and with respect to the values of the parameters used in $\mathcal{C}$. In this approach, the effort has to be made on the design of the model, and in most cases this is a non-trivial task. For instance, to the best of our knowledge, in the literature there is no analytical model of the number of patterns satisfying $\mathcal{C} \equiv MinFr(\phi, \mathbf{r}_1, f_1) \wedge MaxFr(\phi, \mathbf{r}_2, f_2) \wedge |\phi| = k$ when the distribution of the symbols is represented by a first-order Markov chain and when soft-occurrences are used to handle degenerated patterns (even in the simple case where $\delta = 1$). Designing an analytical model to handle this case is certainly not straightforward, in particular because of the specific symbol distribution that has to be incorporated in the model. We propose a third approach based on the following key remark. When a pattern ϕ is given, together with the distribution of the symbols, the structure of the datasets and the values of the parameters in $\mathcal{C}$, we can compute $P(\phi\ sat.\ \mathcal{C})$, the probability that ϕ satisfies $\mathcal{C}$ in this dataset. In most cases, designing a function to compute $P(\phi\ sat.\ \mathcal{C})$ is rather easy in comparison to the effort needed to exhibit an analytical model that estimates the number of patterns satisfying the constraint $\mathcal{C}$. Having at hand a function to compute $P(\phi\ sat.\ \mathcal{C})$, the next step is then to estimate the total number of patterns that will be extracted, but without having to compute $P(\phi\ sat.\ \mathcal{C})$ for all patterns in the pattern space. Therefore, we propose a simple pattern space sampling approach, that leads to a fast and accurate estimate of the number of patterns that will be extracted. Finally, we can compute such an estimate for a large number of points in the parameter space and it provides views of the whole extraction landscape.

3. Estimate based on pattern space sampling

3.1. Symbol distributions and probability to satisfy the constraint

We choose three symbol distributions, to show that our method can be used with different models. However, this choice is not central in the contribution, and depending on the application domain, other dedicated models that would be more accurate could be used to provide a better estimate.

The three models of distributions retained here are:

- $\mathcal{M}_E$: independence of all occurrences of the symbols with equal occurrence frequencies of each symbol;
- $\mathcal{M}_D$: independence of all occurrences of the symbols with different occurrence frequencies of the symbols;
- $\mathcal{M}_M$: a first-order Markov chain.

For each of the three models mentioned above, it is easy to compute the probability for a given pattern ϕ to occur in a string, then to obtain the probability to satisfy a frequency constraint using a binomial law, and to finally determine $P(\phi\ sat.\ \mathcal{C})$.

3.2. Estimate of the number of patterns satisfying the constraint

Let $S_{\mathcal{C}}$ be the set of patterns in $\mathcal{L}$ that satisfy the constraint $\mathcal{C} \equiv MinFr(\phi, \mathbf{r}_1, f_1) \wedge MaxFr(\phi, \mathbf{r}_2, f_2) \wedge \mathcal{C}_{synt}(\phi)$. In this section, we present a simple method to estimate $|S_{\mathcal{C}}|$ by sampling the pattern space and using a function that gives $P(\phi\ sat.\ \mathcal{C})$ for any pattern ϕ.

Let us associate to each pattern ϕ a random variable X_ϕ, such that $X_\phi = 1$ when ϕ satisfies $\mathcal{C}$ and $X_\phi = 0$ otherwise. Then $|S_{\mathcal{C}}| = \sum_{\phi \in \mathcal{L}} X_\phi$. Considering the expected value of $|S_{\mathcal{C}}|$, by linearity of the expectation operator we have $E(|S_{\mathcal{C}}|) = \sum_{\phi \in \mathcal{L}} E(X_\phi)$. Since $E(X_\phi) = 1 \times P(X_\phi =$

$1) + 0 \times P(X_\phi = 0)$, then $E(|S_\mathcal{C}|) = \sum_{\phi \in \mathcal{L}} P(\phi \; sat. \; \mathcal{C})$. Let $S_{\mathcal{C}_{synt}}$ be the set of patterns in $\mathcal{L}$ that satisfy $\mathcal{C}_{synt}$. As $P(\phi \; sat. \; \mathcal{C}) = 0$ for all patterns that do not satisfy $\mathcal{C}_{synt}$, we have $E(|S_\mathcal{C}|) = \sum_{\phi \in S_{\mathcal{C}_{synt}}} P(\phi \; sat. \; \mathcal{C})$.

Computing this sum over $S_{\mathcal{C}_{synt}}$ would be prohibitive, since we want to obtain $E(|S_\mathcal{C}|)$ for a large number of points in the parameter space. Thus we estimate $E(|S_\mathcal{C}|)$ using only a sample of the patterns in $S_{\mathcal{C}_{synt}}$. Let S_{samp} be such a sample, then we use the following value as an estimate of $E(|S_\mathcal{C}|)$:

$$\frac{|S_{\mathcal{C}_{synt}}|}{|S_{samp}|} \times \sum_{\phi \in S_{samp}} P(\phi \; sat. \; \mathcal{C})$$

In practice, many techniques can be used to compute the sample. In the experiments presented in the next section, we use the following process:

- Step 1: build an initial sample S_{samp} of $\mathcal{C}_{synt}$ (sampling with replacement) of size 5% of $|\mathcal{C}_{synt}|$ and compute the estimate of $E(|S_\mathcal{C}|)$.
- Step 2: go on sampling with replacement to add 1,000 elements to S_{samp}. Compute the estimate, and if the absolute value of the difference between the new estimate and the previous one is greater than 5% of the previous estimate, then repeat Step 2.

Since the variables X_ϕ are not independent (the occurrence of a pattern has an impact on the possibility of occurrence of other patterns), there is no straightforward analytical confidence bound of the estimate. It is out of the scope of this paper to further discuss this issue. However, in the next section, we show that the estimate is quite accurate in practice.

4. Experiments

4.1. Empirical evaluation of the estimate

To empirically assess our method, we have to check both its efficiency in terms of running time and accuracy. Recall that we may need to estimate the expected number of patterns satisfying a user-defined constraint for a large number of values in parameter domains.

We generated three pairs of random datasets $\mathbf{r}_1$ and $\mathbf{r}_2$, and on each pair we performed a set of experiments. Each pair is based on a different symbol distribution and/or on a different dataset structure. The symbol distributions used for the estimate were the same as the ones used for the generation. For each set of experiments, we present graphics to compare the estimate of the expected number of patterns versus the real number of patterns extracted in the datasets when using the same parameters. In the experiments, we explore different regions in the parameter space, at different scales, and we do not try to focus on parameter ranges that lead to the best estimates.

In the experiments, we consider the extraction of all patterns satisfying $MinFr(\phi, \mathbf{r}_1, f_1) \wedge MaxFr(\phi, \mathbf{r}_2, f_2) \wedge |\phi| = k$, varying the minimal (resp. maximal) support threshold f_1 (resp. f_2) and the size k, using in turn both exact supports and soft supports (with $\delta = 1$). In all graphics, the isolated dots represent the estimates, the dots linked by a line represent the real number of extracted patterns, the minimal support corresponds to the horizontal axis, and the number of patterns corresponds to the vertical axis (for the sake of readability, we use for some of the graphics a log scale axis). The settings used for the three sets of experiments are the following:

- First set of experiments (Figure 1): 4 symbols with distribution $\mathcal{M}_D$ (frequencies of the symbols are 0.4, 0.1, 0.2 and 0.3), datasets $\mathbf{r}_1$ and $\mathbf{r}_2$ contain 100 strings of length 1,000.
- Second set of experiments (Figure 2): 4 symbols with distribution $\mathcal{M}_M$, datasets $\mathbf{r}_1$ and $\mathbf{r}_2$ contain 100 strings of length 1,000. The arbitrary conditional probabilities $Prob(\phi_i = Y | \phi_{i-1} = X)$ used for the Markov chain (with symbols A, B, C and D) are given by the table:

X \ Y	A	B	C	D
A	0.2	0.28	0.18	0.34
B	0.04	0.36	0.3	0.3
C	0.32	0.08	0.2	0.4
D	0.2	0.24	0.24	0.32

- Third set of experiments (Figure 3): 8 symbols with distribution $\mathcal{M}_E$, datasets $\mathbf{r}_1$ and $\mathbf{r}_2$ contain 100 strings of length 30,000. In four of the graphics, the estimates are so close to the real values that the corresponding dots are superimposed.

In all experiments, the estimates closely follow the trends of the real extractions, and in most cases, the estimates are sufficiently accurate to give a reasonable picture of the shape of the extraction landscape. For each experiment, only between $4,000$ and $8,000$ sampled patterns were necessary to converge to a stable estimate, i.e., the difference between two successive estimates is smaller than 5% of the first one.

The experiments were run on a Linux platform with an Intel 2Ghz processor and 1Gb of RAM. The real pattern extractions on the datasets were performed using the $Marguerite$ [9, 10] prototype (implemented in C++) that supports constraint-based string mining with exact and soft

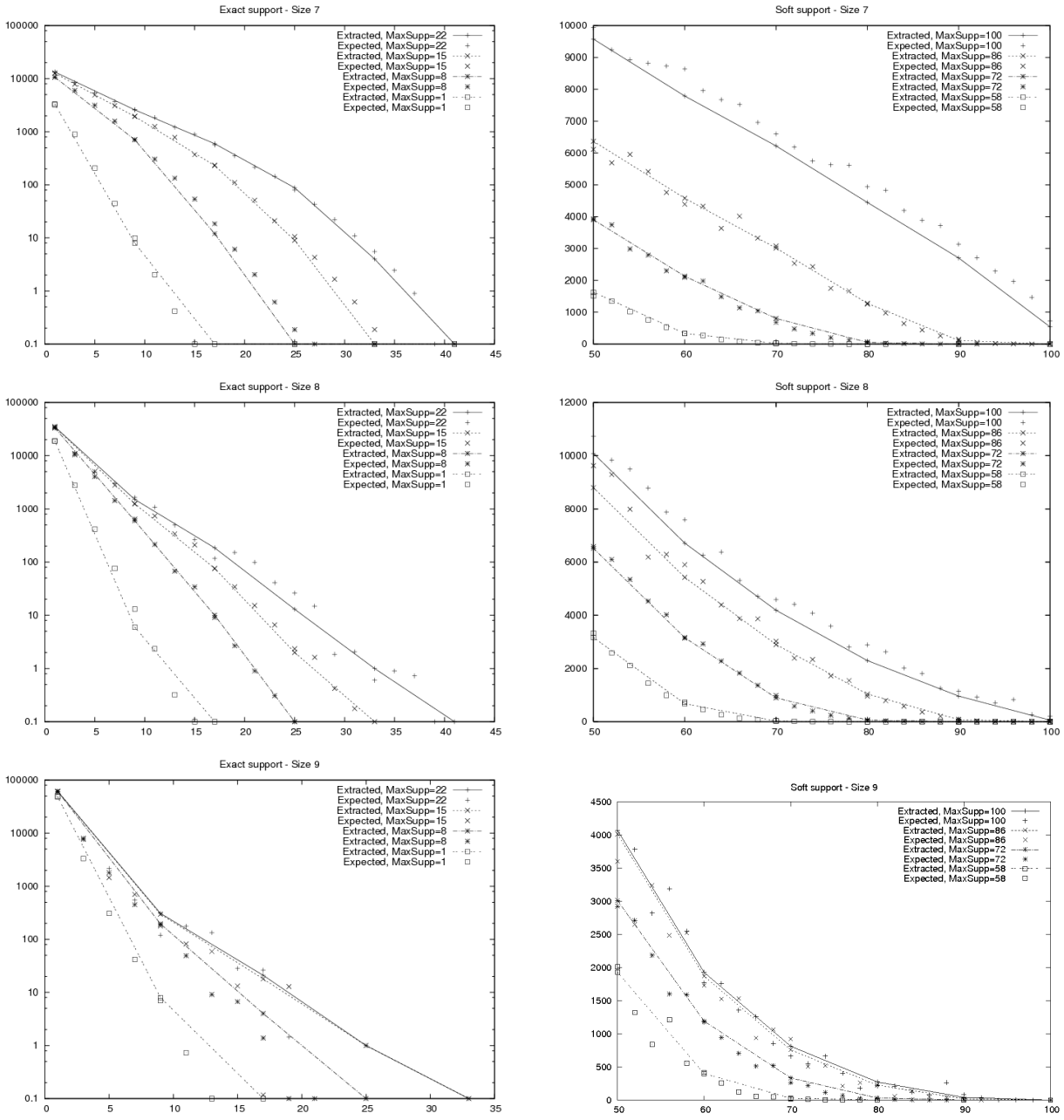

Figure 1. Experiments under $\mathcal{M}_D$ distribution (with symbol frequencies 0.4, 0.1, 0.2 and 0.3). Horizontal axis: minimum support, vertical axis: number of patterns.

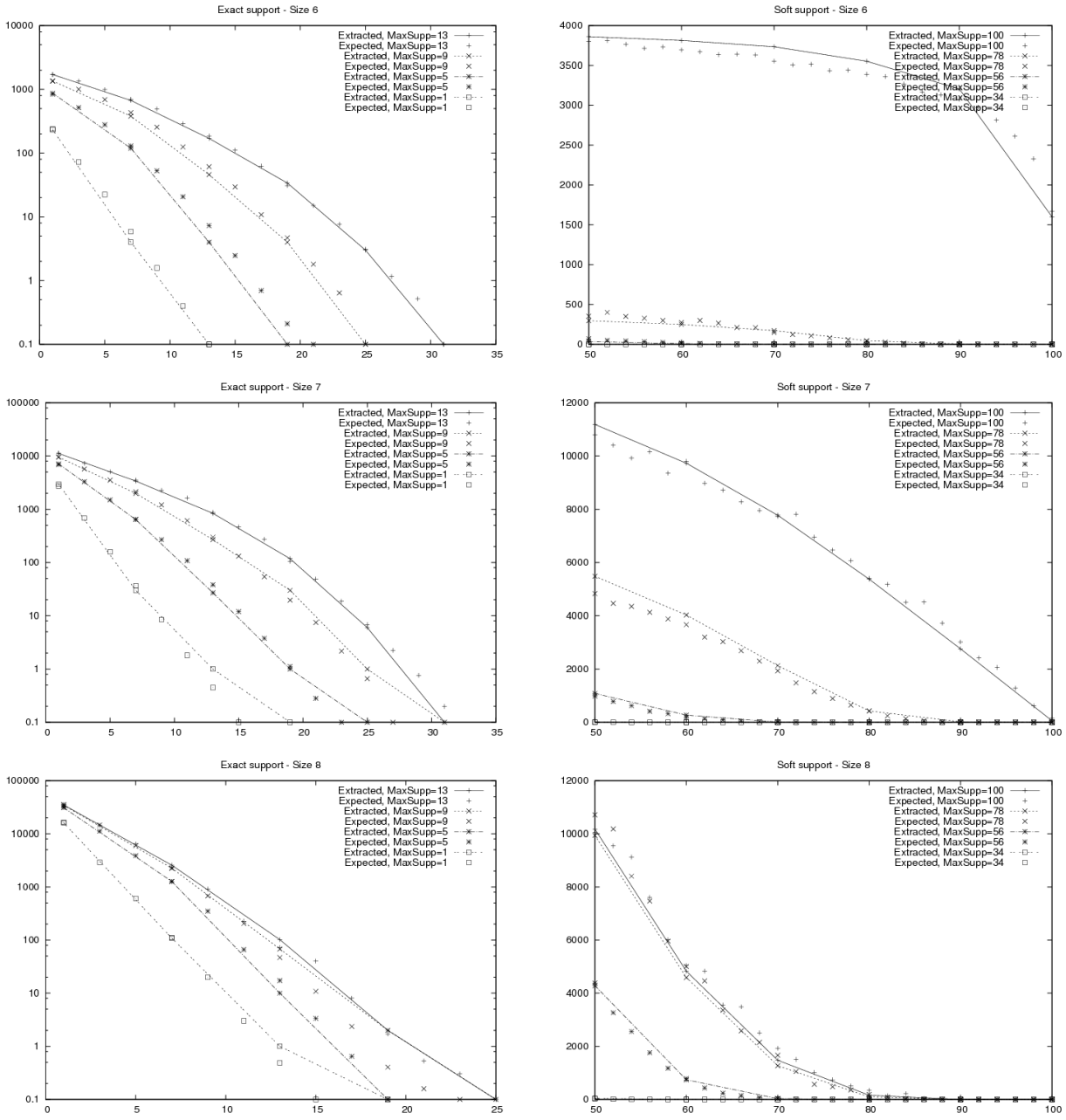

Figure 2. Experiments under $\mathcal{M}_M$ distribution (1st-order Markov chain). Horizontal axis: minimum support, vertical axis: number of patterns.

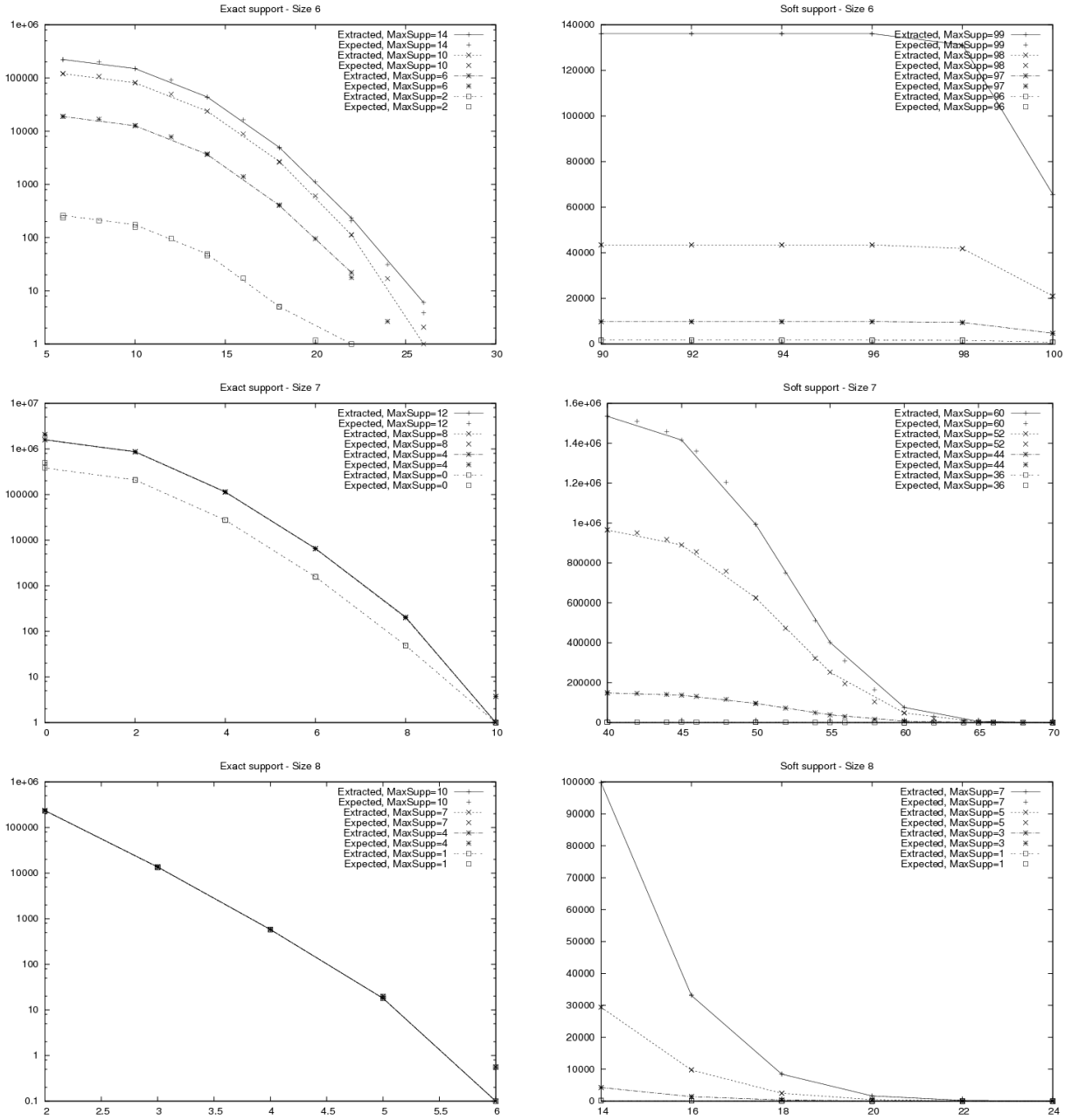

Figure 3. Experiments under $\mathcal{M}_E$ distribution (same frequency for symbols) and string size 30,000. Horizontal axis: minimum support, vertical axis: number of patterns.

supports, and that handles in particular the conjunction of constraints used in this paper. The estimate of the number of patterns, based on the pattern space sampling, was implemented in Perl. For each single extraction with exact support, the running time was about a few tens of seconds to a few minutes. In the case of soft support, an extraction takes from a few tens of minutes to several hours to complete, while for the sampling-based estimate, computing one estimate requires between 1 and 30 seconds.

4.2. Application to Real Data

We use two datasets of DNA sequences that are promoter sequences of genes. Each sequence is a string of 4,000 symbols over an alphabet of size 4 (nucleotides a, c, g, and t). The first dataset $\mathbf{r}_1$ contains promoter sequences of 29 genes and dataset $\mathbf{r}_2$ contains promoter sequences of 21 genes. These two datasets represent two biologically opposite situations of interest, and extracting string patterns that have a high support in $\mathbf{r}_1$ and a small support in $\mathbf{r}_2$ is a way to identify putative binding sites of transcription factors (molecules that bind on the promoter sequences to activate or to repress a gene). The data were prepared and provided by Dr. Gandrillon and his team [2].

We look for patterns satisfying $MinFr(\phi, \mathbf{r}_1, f_1) \wedge MaxFr(\phi, \mathbf{r}_2, f_2) \wedge |\phi| = k$, and for the support we used the soft support definition with $\delta = 1$ (slightly degenerated occurrences). Such a conjunction of constraints, with soft support definition based on the Hamming distance, has been shown to be useful in practice for transcription factor binding sites identification [11].

The estimates were computed with distribution $\mathcal{M}_D$, using as symbol frequencies their respective frequencies in the data (0.23, 0.26, 0.27, 0.24 respectively for a, c, g, and t). In this case, the model is more a description of the random background than a description of the biological organization along the sequences. Representative graphics obtained using these estimates, and depicting portions of the extraction landscape, are presented in Figure 4, on the right. A typical use of such graphics is for instance to look for points, in the parameter space, corresponding to a large support on $\mathbf{r}_1$, but a low support on $\mathbf{r}_2$, a large pattern size, and a rather small number of expected patterns (since here the distribution represents the random background). Such a point, that can be used as an initial guess of the parameters to perform real extractions, is for instance: pattern size $= 10$, minimal support on $\mathbf{r}_1$ of 15 and maximal support on $\mathbf{r}_2$ of 5 (the graphic in the middle on the right indicates that, for this setting, only about 1 pattern due to the random background is expected).

For the sake of completeness, in Figure 4 on the left, we give the real numbers of extracted patterns. In practice, these graphics are not easily accessible to the user, since in these experiments the running time of a single extraction ranges from several tens of minutes to several hours (while for an estimate, only a few tens of seconds is needed). Even though the global trends correspond to the estimates, they are important differences in some portions of the parameter space (these differences could be expected since the distribution used does not incorporate complex biological knowledge). However, the estimates can still be used to help to choose initial parameter values in an exploratory mining stage, and moreover, finding such differences, when running the real extractions, can also be a useful piece of information in itself. For example, for the setting *pattern size* $= 10$, *minimal support* $= 15$ and *maximal support* $= 5$, we have about 100 patterns really extracted, while we expected only one. If we suppose that the pattern space sampling can provide reasonable estimates when the data satisfy distribution $\mathcal{M}_D$ (as supported by the experiments of Section 4.1), then the 100 patterns obtained are likely to be due to a particular unknown structure in the data and not to the distribution $\mathcal{M}_D$ only. This suggest that it makes sense to look for patterns in this region of the parameter space, since we can expect to obtain some interesting/useful patterns (not only patterns due to the random background captured by $\mathcal{M}_D$).

5. Related work

Estimating the expected number of patterns that satisfy a constraint is in general much more difficult than estimating the probability that a given pattern satisfies such a constraint. This second problem has received a lot of attention, leading to many statistical measures to assess the interestingness of the patterns. Concerning the first problem, only a few solutions have been proposed. [12] and [8] analyze the feasible distributions of frequent itemsets (also of closed itemsets for [8] and of maximal itemsets for [12]). [12] focuses on the kind of distributions one can expect for various kinds of datasets. They answer the question whether there exists a frequent or maximal frequent itemset collection that has a given number of frequent itemsets of a given length. [8] computes the average number of frequent (closed) itemsets using probabilistic techniques. These authors especially focus on minimal frequency threshold and how it influences the number of extracted patterns, considering fixed and/or proportional thresholds. Another approach has been proposed by Geerts et al. [5], providing a tight upper bound on the number of candidate patterns that can arise while mining frequent patterns in a level-wise setting. Given the current level and the current set of frequent patterns, they propose a tight bound of the maximal number of candidate patterns that can be generated on the next level. In the domain of string mining, [6] designs an estimate of the number of patterns due to the random background, and that are likely to be extracted with respect to

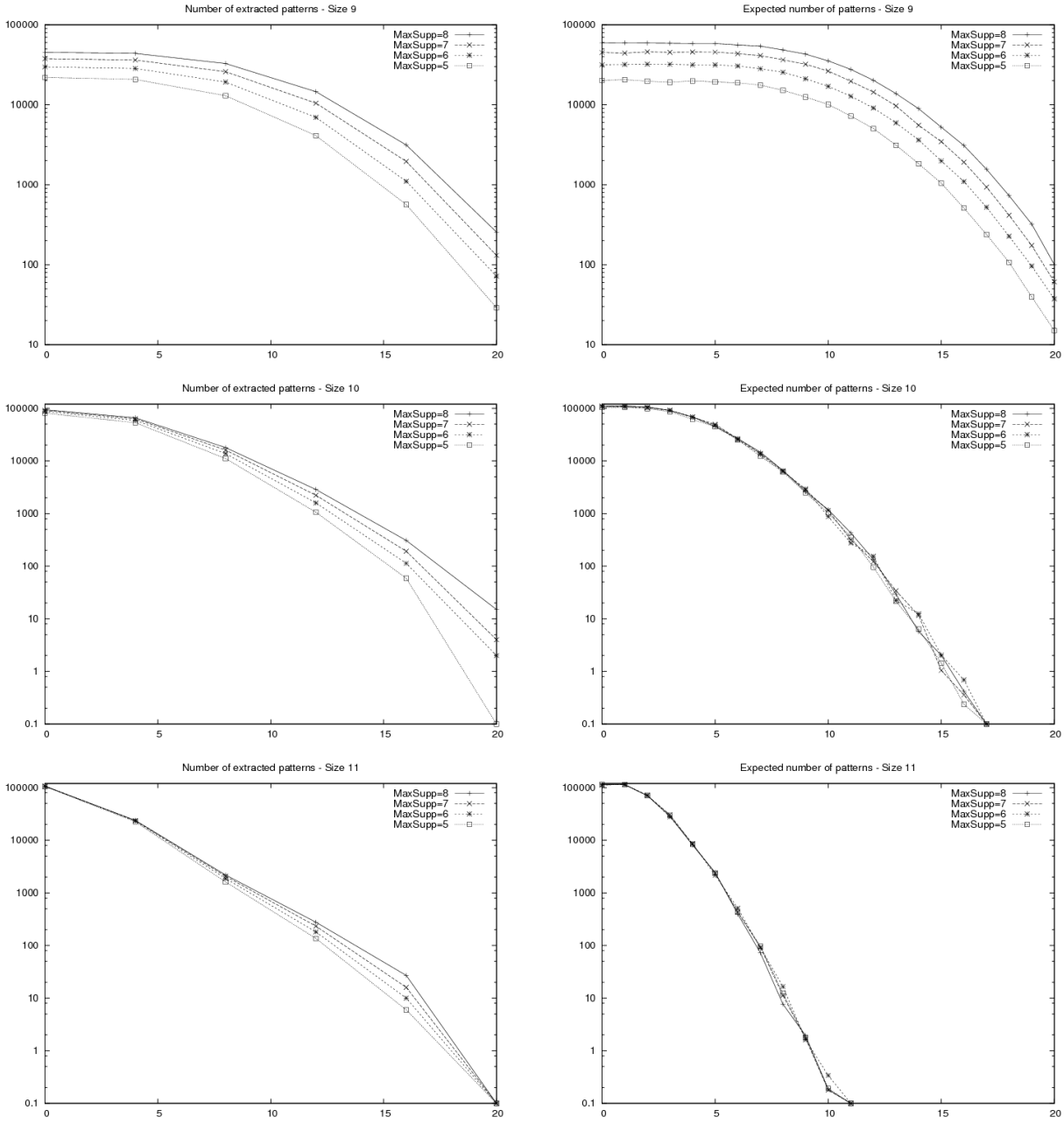

Figure 4. Expected and real number of extracted patterns, using two promoter sequence datasets. Horizontal axis: minimum support, vertical axis: number of patterns.

a frequency constraint and according to the structure of the dataset. These proposals are all based on a global analytical model, i.e., an interesting approach that needs however to develop complex and specific models. As a result, they cannot be easily extended to handle complex conjunctions of constraints, to incorporate different symbol distributions or different semantics for pattern occurrences. To the best of our knowledge, no method has been proposed to estimate the number of patterns satisfying a constraint while avoiding to develop a global analytical model. Our approach requires only to know how to compute for a given pattern its probability to satisfy the constraint (this can be obtained in many situations), and it remains efficient in practice by adopting a pattern space sampling scheme.

6. Conclusion

Using constraints to specify subjective interestingness issues and to support actionable pattern discovery has become popular. Constraint-based mining techniques are now well studied for many pattern domains but one of the bottlenecks for using them within Knowledge Discovery processes is the extraction parameter tuning. This is especially true in the context of differential mining where domain knowledge is used to provide different datasets to support the search of truly interesting patterns. From a user perspective, a simple approach would be to get graphics that depict the extraction landscape (i.e., the number of extracted patterns for many points in the parameter space). We developed an efficient technique based on pattern space sampling, that provides an estimate on the number of extracted patterns. This has been applied to non trivial substring pattern mining tasks, and we demonstrated by means of many experiments that the technique is effective. It provides reasonable estimates given execution times that enable to probe a large number of points in the parameter space. Notice that domain knowledge is also exploited here when selecting the distribution model. Future directions of work include to adapt the approach to other pattern domains and to different constraints. Another interesting aspect to investigate is the use of more sophisticated sampling schemes (e.g., [13]), that could be incorporated in the approach when more complex syntactical constraints are handled (e.g., a grammar to specify the shape of the patterns).

Acknowledgments. This work is partly funded by EU contract IQ FP6-516169 (Inductive Queries for Mining Patterns and Models) and by the French contract ANR-MDCO-14 Bingo2 (Knowledge Discovery For and By Inductive Queries). We thank Dr. Olivier Gandrillon from the Center for Molecular and Cellular Genetics (CNRS UMR 5534) who provided the DNA promoter sequences.

References

[1] J.-F. Boulicaut, L. De Raedt, and H. Mannila, editors. *Constraint-Based Mining and Inductive Databases*, volume 3848 of *LNCS*. Springer, 2005.

[2] C. Bresson, C. Keime, C. Faure, Y. Letrillard, M. Barbado, S. Sanfilippo, N. Benhra, O. Gandrillon, and S. Gonin-Giraud. Large-scale analysis by SAGE revealed new mechanisms of v-erba oncogene action. *BMC Genomics*, 8(390), 2007.

[3] L. Cao and C. Zhang. Domain-driven actionable knowledge discovery in the real world. In *Proceedings PAKDD'06*, volume 3918 of *LNCS*, pages 821–830. Springer, 2006.

[4] G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proceedings ACM SIGKDD'99*, pages 43–52, 1999.

[5] F. Geerts, B. Goethals, and J. V. den Bussche. Tight upper bounds on the number of candidate patterns. *ACM Trans. on Database Systems*, 30(2):333–363, 2005.

[6] U. Keich and P. A. Pevzner. Subtle motifs: defining the limits of motif finding algorithms. *Bioinformatics*, 18(10):1382–1390, 2002.

[7] S. Kramer, L. De Raedt, and C. Helma. Molecular feature mining in HIV data. In *Proceedings KDD'01*, pages 136–143, 2001.

[8] L. Lhote, F. Rioult, and A. Soulet. Average number of frequent (closed) patterns in bernouilli and markovian databases. In *Proceedings IEEE ICDM'05*, pages 713–716, 2005.

[9] I. Mitasiunaite and J.-F. Boulicaut. Looking for monotonicity properties of a similarity constraint on sequences. In *Proceedings of ACM SAC'06 Data Mining*, pages 546–552, 2006.

[10] I. Mitasiunaite and J.-F. Boulicaut. Introducing softness into inductive queries on string databases. In *Databases and Information Systems IV*, pages 117–132. IOS Press, 2007.

[11] I. Mitasiunaite, C. Rigotti, S. Schicklin, L. Meyniel, J.-F. Boulicaut, and O. Gandrillon. Extracting signature motifs from promoter sets of differentially expressed genes. Technical report, LIRIS CNRS UMR 5205, INSA Lyon, France, 2008. 23 pages. Submitted.

[12] G. Ramesh, W. Maniatty, and M. J. Zaki. Feasible itemset distributions in data mining: theory and application. In *Proceedings ACM PODS'03*, pages 284–295, 2003.

[13] F. Železný. Efficient sampling in relational feature spaces. In *Proceedings ILP'05*, volume 3625 of *LNCS*, pages 397–413. Springer, 2005.