Aggregated Subset Mining

Albrecht Zimmermann and Björn Bringmann

Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium
{Albrecht.Zimmermann,Bjorn.Bringmann}@cs.kuleuven.be

Abstract. The usual data mining setting uses the full amount of data to derive patterns for different purposes. Taking cues from machine learning techniques, we explore ways to divide the data into subsets, mine patterns on them and use post-processing techniques for acquiring the result set. Using the patterns as features for a classification task to evaluate their quality, we compare the different subset compositions, and selection techniques. The two main results – that small independent sets are better suited than large amounts of data, and that uninformed selection techniques perform well – can to a certain degree be explained by quantitative characteristics of the derived pattern sets.

1 Introduction

Data mining essentially comes in two flavors, *descriptive* mining, finding descriptions of the data, and *predictive* mining, constructing features for effective classification. In predictive mining, class-correlating patterns, patterns showing strong correlation with a class value, are often a good choice. No matter the reliability of statistic measures, however, they can still fall prey to over-fitting, which in turn may harm classifiers. This becomes even more problematic if the amount of patterns is large, or pairs and combinations of patterns reinforce each other's bias.

In a recent work [3], the effect of decreasing redundancy between patterns on the accuracy of classifiers using those particular features was evaluated. While we could show that reducing redundancy – in some cases rather strongly – did in fact improve accuracy, we used the entire data set set for mining patterns which we then filtered. This setting, which is the *standard* data mining setting, is well-suited for descriptive mining. Predictive mining is closer related to machine learning, however, which knows different techniques using parts of the labeled data for verification purposes of found patterns/built classifiers.

We therefore take a page out of the playbook of ML, first mining several sets of correlating patterns, and then using different criteria to create final result sets from them. These are used as features for learning an SVM [5] classifier.

The paper is structured as follows: In the next section, we explain the basic mechanisms for mining patterns, creating subsets of the data for mining and selection purposes, and lay out several selection methods for deriving the final result set. In Section 3, we report on the experimental evaluation of the proposed methods before concluding in Section 4.

2 Mining and Merging Correlating Patterns

We start from set of instances $\mathcal{D}_m$ each being labeled with one of the class labels $\{pos, neg\}$. In this set $\mathcal{D}_m$ we search for patterns drawn from a language $\mathcal{L}$. More specifically for a set of k patterns whose occurrence on the instances correlates best with the presence of the target class according to χ^2 [4]. Further we require the found patterns to be *free* according to [1].

The solutions to the mining task can then be conveniently modeled using

$$\mathcal{T}h_k(\mathcal{D}_m) = \{p \in \mathcal{L} \mid p \text{ among the k-best free patterns on } \mathcal{D}_m \text{ w.r.t. } \chi^2\}$$

As said before, this is the *standard* data mining setting which operates on the full dataset $\mathcal{D}_m$, which we will use as a base-line technique. In the following sections we propose different methods for selecting the final pattern set.

2.1 Using a validation set

The most basic approach consist of using a certain fraction q of the total data $\mathcal{D}_m$ as the actual mining set $\overline{\mathcal{D}}_m$, with size $q \cdot |\mathcal{D}_m|$. The rest would be used as a validation set $\hat{\mathcal{D}}_m = \mathcal{D}_m \setminus \overline{\mathcal{D}}_m$, of size $(1-q) \cdot |\mathcal{D}_m|$. After termination of the mining process on $\overline{\mathcal{D}}_m$, the k_m patterns $\mathcal{T}h_{k_m}(\overline{\mathcal{D}}_m)$ returned by the miner are evaluated on $\hat{\mathcal{D}}_m$ and re-ranked, according to their correlation score χ^2 achieved on this validation set. Out of those the k_s best scoring patterns are returned to the user. It can easily be derived that k_s should be chosen such that $k_s < k_m$ since for $k_m \leq k_s$ the validation scores (and re-ranking) have no effect on the selection of patterns. The final result set is then

$$val_{k_s}(\mathcal{T}h_{k_m}(\overline{\mathcal{D}}_m), \hat{\mathcal{D}}_m) = \{p \in \mathcal{T}h_{k_m}(\overline{\mathcal{D}}_m) \mid p \text{ in the } k_s\text{-best patterns on } \hat{\mathcal{D}}_m \text{ w.r.t. } \chi^2\}$$

Given the use of statistically significant patterns, one would expect a certain robustness against statistical quirks. The degree to which the full distribution can be modeled by a subset could however very well be governed by q. A not unusual choice for q in the machine learning literature is $\frac{2}{3}$.

2.2 Aggregating subset results

In the second approach, subsets $\mathcal{D}^i_m$ of $\mathcal{D}_m$ are created, and the top k_m patterns mined from each of them. For the union of their results $\Phi_{all} = \bigcup_i \mathcal{T}h_{k_m}(\mathcal{D}^i_m)$ we know that $|\Phi_{all}| \geq k_m$. All patterns $p \in \Phi_{all}$ are re-evaluated according to some aggregation metric, and a subset (e.g. the top-k_m patterns) returned to the user.

This approach is illustrated in Figure 1, with merge denoting the merging/re-evaluation step. What should be immediately obvious from this figure is that this kind of approach lends itself to distributed/parallel mining, although the merging step needs to be performed on one particular site.

There are two main decisions that influence the result of this approach, namely the choice of subsets and the aggregation metric used. The size of the final set to be returned is obviously also important but has less effect than the afore-mentioned two choices, we believe.

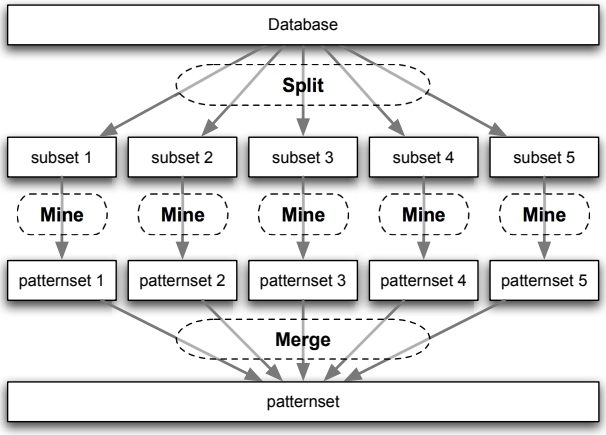

Fig. 1. The aggregated subset mining technique

Formation of subsets We investigate two approaches to forming subsets of $\mathcal{D}_m$. Their main difference lies in whether there is overlap among subsets used or not. The straightest-forward approach consists of segmenting $\mathcal{D}_m$ into f disjunct folds $\mathcal{F}_i$. We define $\hat{\mathcal{D}}_m^i = F_i$ and $\overline{\mathcal{D}}_m^i = \bigcup_{j \neq i} F_j$. In both cases *all* instances in the data have an effect on the final result with the same weight.

Aggregation metrics The goal of any aggregation metric used lies in ranking the patterns in Φ_{all} by using information from all subsets $\mathcal{D}_m^i$ mined on. To this end, we propose three metrics:

1. A first measure takes the form $\mu_{count}(p) = |\{i \ : \ p \in \mathcal{T}h_{k_m}(\mathcal{D}_m^i)\}|$
   Basically μ_{count} counts for each pattern p in how many of the $\mathcal{D}_m^i$ it was found among the top-k_m. This measure only checks whether a pattern was mined at all, however, not what its particular rank was in the respective result sets.
2. Hence, a second metric, μ_{rank} would consist of the following:

$$\mu_{rank}(p) = \frac{1}{f} \sum_i rank(p, \mathcal{D}_m^i)$$

   with

$$rank(p, \mathcal{D}) = \begin{cases} 1 + k_m - \inf\{k \mid p \in \mathcal{T}h_k(\mathcal{D})\} & \textit{if} \ \ p \in \mathcal{T}h_{k_m}(\mathcal{D}) \\ 0 \ \textit{otherwise} \end{cases}$$

   taking a pattern's "local" quality into account.
3. A final metric, μ_{χ^2} would take the form: $\mu_{\chi^2}(p) = (\chi^2$ of p on $\mathcal{D}_m)$ calculating the score for each pattern according to χ^2 on the entire set $\mathcal{D}_m$. This metric is related to the validation approach of 2.1, with the difference that here the data on which the pattern was mined is also used for validation.

Selection criteria Due to the fact that $|\Phi_{all}| \geq k_m$, one can simply return the top-k_m after the re-ranking via one of the metrics $\mu(p)$. Thus, given a value k_m, a metric μ, and a set of subsets $\mathcal{M} = \{\mathcal{D}_m^i\}$, our goal is to select $\varphi_{k_m}(\mathcal{M}, \mu) \subseteq \Phi_{all}$ such that the $p_i \in \varphi_{k_m}(\mathcal{M}, \mu)$ are the k_m highest ranked pattern in Φ_{all} w.r.t. μ.

Additionally, this framework does allow for a second k-value (k_s), similar to the one of the validation set approach which is used to define the size of the final result set leading to $\Phi_{all} = \bigcup_i \mathcal{T}h_{k_m}(\mathcal{D}_m^i)$ with $\varphi_{k_s}(\mathcal{M}, \mu) \subseteq \Phi_{all}$.

3 Experimental Evaluation

For the experimental evaluation, we arbitrarily picked 8 data sets from the NCI-60 data set collection [6] and mine sequential patterns on them. Each has about 3500 instances (one outlier having only 2778) with a class distribution of $50-53\%$ (another outlier of 63.7%) for the positive class. We chose $k_m \in \{10, 25, 50, 75, 100\}$, giving a reasonable range of values across which to compare. We evaluated two aspects of the outlined techniques experimentally:

Q1 **Quantitative Analysis:** The effect of different *subset formation* strategies and subset sizes. Specifically, we consider similarity between the pattern sets finally selected and the ones mined in the *standard* setting.

Q2 **Qualitative Analysis:** The effect of different *aggregation* methods on the quality of the pattern sets selected.

To get a robust accuracy estimate, a 10-fold cross-validation was performed. All folds – both for accuracy and selection purposes – were stratified. As mentioned above, an SVM classifier was used for accuracy estimates. SVMs possess certain inherent feature selection capabilities, giving low-relevance features small weights. In addition, an SVM attempts to find a separating hyperplane with a maximal margin to both classes. Both of these characteristics guard against overfitting at the classifier level, allowing to evaluate the quality of the *feature set*. The SVM's C parameter was tuned via a 5-fold cross-validation on the training data, with potential values $2^i, i \in [-2, 14]$.

3.1 Validating patterns on additional data

In the first setting, using a validation set for assessing found patterns' quality, we evaluated two stratified random splits of the mining data, with $q = \frac{2}{3}$ and $q = \frac{4}{5}$, respectively. As to the top-k parameters, $k_m, k_s \in \{10, 25, 50, 75, 100\} : k_m > k_s$.

Quantitative results A useful measure for assessing quantitative characteristics is that of overlap. Given two sets of patterns $\mathbb{S}, \mathbb{S}'$, we simply define $ovlp(\mathbb{S}, \mathbb{S}') = |\mathbb{S} \cap \mathbb{S}'|$. For evaluating the quantitative characteristics of selected pattern sets, we calculate the overlap with the standard mining operation (denoted by *standard* in Table 1): $ovlp(val_{k_s}(\mathcal{T}h_{k_m}(\overline{\mathcal{D}}_m), \hat{\mathcal{D}}_m), \mathcal{T}h_{k_s}(\mathcal{D}_m))$, and with the *non-validated* pattern set: $ovlp(val_{k_s}(\mathcal{T}h_{k_m}(\overline{\mathcal{D}}_m), \hat{\mathcal{D}}_m), \mathcal{T}h_{k_s}(\overline{\mathcal{D}}_m))$ (denoted by *non-validated*).

non-validated $q = \frac{2}{3}$	standard $q = \frac{2}{3}$	k_s/k_m	non-validated $q = \frac{4}{5}$	standard $q = \frac{4}{5}$
3.1 ±1.1005	2.7 ±1.05935	10/25	3.6 ±1.26491	3.7 ±0.823273
0.7 ±0.948683	0.5 ±0.971825	10/50	1.2 ±1.31656	1.4 ±1.17379
0.2 ±0.421637	0.2 ±0.421637	10/75	0.6 ±0.966092	0.7 ±0.948683
0.1 ±0.316228	0 ±0	10/100	0.3 ±0.674949	0.3 ±0.674949
11.1 ±1.96921	10 ±2	25/50	11 ±2.16025	10.9 ±2.55821
6.3 ±1.82878	6.1 ±2.33095	25/75	6.3 ±2.11082	6.3 ±1.88856
4.3 ±0.948683	4 ±1.56347	25/100	4 ±1.33333	4.1 ±1.59513
32.5 ±1.71594	28.5 ±2.4608	50/75	32.6 ±1.7127	30.8 ±1.68655
24 ±1.69967	21.6 ±1.57762	50/100	22.9 ±2.37814	22.5 ±2.83823
56.8 ±1.8738	48.3 ±3.56059	75/100	56.3 ±1.41814	53.4 ±1.77639

Table 1. Overlap between the validated sets and the non-validated/standard setting, respectively

The pattern set overlap values show that the greater k_m for a given k_s, the smaller the overlap between pattern sets becomes. This means that patterns are ranked rather differently on the validation set although similar underlying distributions should be expected. Overlap is of course higher for comparison against the non-validated set since the validated set is constructed from this. It is interesting however, that the difference between comparison to the standard and the non-validated setting are not that great. Furthermore, there is no big change between the results for $q = \frac{2}{3}$ and $q = \frac{4}{5}$.

Qualitative results Regarding Q2, we use the selected features to encode $\mathcal{D}_m$ as binary vectors and evaluate the SVM's performance. The main focus of our comparison lies on determining which q is better suited to the mining of "good" features, and whether there are particularly well-suited $k_m - k_s$ combinations.

We report the results on a representative data set in Table 2. Unfortunately, the answer seems to be that neither q is a good choice. Using a validation set selects features that are less well suited for classification than mining on the full data. This indicates that randomly splitting the data can give rise to so radically different distributions (hinted at in the quantitative analysis) that top-k selections based on χ^2 becomes meaningless.

3.2 Aggregated pattern selection

For the second setting –using different subsets to mine the data and using aggregation metrics– we chose $f \in \{3, 5, 7\}$, thus allowing for different sizes of $\mathcal{D}_m$. In addition to the standard setting, we compare to a post-processing method which uniformly picks k_s patterns from Φ_{all} at random. Since this method does not use *explicit* information on patterns' quality, nor their relationship, we use it as a baseline to see whether the better informed methods enjoy an advantage.

k_s	10	25	50	75
Standard setting (full $\mathcal{D}_m$)				
$k_m = k_s$	59.752 ± 1.974	60.949 ± 2.361	62.205 ± 3.168	64.859 ± 2.479
Validation setting $q = 0.66$				
k_m=25	54.507 ± 1.358	–	–	–
k_m=50	54.65 ± 2.590	55.134 ± 2.389	–	–
k_m=75	54.565 ± 1.605	55.704 ± 1.690	59.152 ± 3.848	–
k_m=100	52.997 ± 0.768	56.732 ± 2.492	56.017 ± 4.070	57.617 ± 5.830
Validation setting $q = 0.80$				
k_m=25	53.139 ± 1.172	–	–	–
k_m=50	51.972 ± 0.895	55.163 ± 2.890	–	–
k_m=75	52.256 ± 0.061	54.166 ± 2.515	57.928 ± 3.547	–
k_m=100	52.227 ± 0.140	53.452 ± 2.214	58.015 ± 2.450	61.293 ± 2.243

Table 2. Predictive accuracies of the validation settings and the standard setting (top row)

f	Overlap	$\lvert\Phi_{all}\rvert/k_m$	$\max_{p\in\Phi_{all}} \mu_{count}(p)$	$\max_{p\in\Phi_{all}} \mu_{rank}(p)$	$\min_{p\in\Phi_{all}} \mu_{rank}(p)$
			$k_m = 10$		
3	3.5 ±1.178	2.000 ± 0.200	3 ± 0.000	7.800 ± 1.033	0.333 ± 0.000
5	1.1 ±0.567	3.260 ± 0.302	4.2 ± 0.632	6.260 ± 0.766	0.200 ± 0.000
7	0.3 ±0.483	4.630 ± 0.434	4.7 ± 0.483	4.814 ± 0.919	0.143 ± 0.000
			$k_m = 25$		
3	8.7 ±2.213	2.136 ± 0.163	3 ± 0.000	22.800 ± 1.033	0.333 ± 0.000
5	4.7 ±1.494	3.428 ± 0.204	4.8 ± 0.422	19.420 ± 1.459	0.200 ± 0.000
7	3.7 ±1.159	4.668 ± 0.305	5.8 ± 0.632	15.614 ± 1.355	0.143 ± 0.000
			$k_m = 50$		
3	19.6 ±2.756	2.114 ± 0.131	3 ± 0.000	47.800 ± 1.033	0.333 ± 0.000
5	11.3 ±1.702	3.290 ± 0.208	5 ± 0.000	43.820 ± 2.165	0.200 ± 0.000
7	8.9 ±1.370	4.454 ± 0.367	6.6 ± 0.699	36.700 ± 2.203	0.143 ± 0.000
			$k_m = 75$		
3	32 ±3.126	2.056 ± 0.123	3 ± 0.000	72.800 ± 1.033	0.367 ± 0.105
5	19.7 ±2.496	3.167 ± 0.171	5 ± 0.000	68.760 ± 2.299	0.200 ± 0.000
7	15 ±1.763	4.389 ± 0.283	6.7 ± 0.675	60.171 ± 4.035	0.143 ± 0.000
			$k_m = 100$		
3	43.3 ±5.375	2.018 ± 0.128	3 ± 0.000	97.800 ± 1.033	0.333 ± 0.000
5	26.7 ±2.945	3.114 ± 0.171	5 ± 0.000	93.760 ± 2.299	0.200 ± 0.000
7	20.5 ±1.715	4.321 ± 0.234	6.8 ± 0.422	84.071 ± 5.767	0.143 ± 0.000

Table 3. Quantitative characteristics for pattern sets mined on $\hat{\mathcal{D}}_m^i$

Quantitative results Regarding Q1 and given similar results for all data sets, we report quantitative characteristics of Φ_{all} in Tables 3 and 4 on one example. For both alternatives regarding construction of the $\mathcal{D}_m$ we list the minimum and maximum μ_{count} and μ_{rank} for patterns in Φ_{all}, $|\Phi_{all}|/k_m$, and $ovlp(\varphi_{k_m}(\mathcal{M}, \mu_\sigma), \mathcal{T}h_{k_m}(\bigcup \mathcal{M}))$. We would expect that:

- $ovlp_{\overline{\mathcal{D}}_m} \geq ovlp_{\hat{\mathcal{D}}_m}$ – Larger $\mathcal{D}_m$ give similar results as the standard setting
- $|\Phi_{all,\hat{\mathcal{D}}_m}|/k_m \gg |\Phi_{all,\overline{\mathcal{D}}_m}|/k_m$ – Smaller $\mathcal{D}_m$ give a larger variety of patterns
- $\min_{p\in\Phi_{all}} \mu_{count}(p) > 1$ – No pattern appears in only one result set
- $\max_{p\in\Phi_{all}} \mu_{count}(p) \approx f$ – The best patterns generalize over most $\mathcal{D}_m$
- $\min_{p\in\Phi_{all}} \mu_{rank}(p) > 1/f$ – No pattern is always ranked worst
- $\max_{p\in\Phi_{all}} \mu_{rank}(p) \approx k_m$ – The best patterns generalize over most $\mathcal{D}_m$, appearing with a high ranking

The evaluation shows that most of our expectations hold, the only serious exceptions being our assumptions about the "worst" patterns – which often appear

f	Overlap	$\lvert\Phi_{all}\rvert/k_m$	$\max_{p\in\Phi_{all}} \mu_{count}(p)$	$\max_{p\in\Phi_{all}} \mu_{rank}(p)$	$\min_{p\in\Phi_{all}} \mu_{rank}(p)$
			$k_m = 10$		
3	6.6 ±1.074	1.440 ± 0.150	3 ± 0.000	9.033 ± 0.508	0.367 ± 0.105
5	6.5 ±0.849	1.430 ± 0.125	5 ± 0.000	9.420 ± 0.416	0.260 ± 0.135
7	6.6 ±1.429	1.400 ± 0.176	7 ± 0.000	9.500 ± 0.318	0.157 ± 0.045
			$k_m = 25$		
3	13.6 ±1.429	1.644 ± 0.125	3 ± 0.000	24.033 ± 0.508	0.333 ± 0.000
5	13.6 ±1.264	1.660 ± 0.080	5 ± 0.000	24.420 ± 0.416	0.220 ± 0.063
7	12.7 ±1.766	1.684 ± 0.115	7 ± 0.000	24.500 ± 0.318	0.157 ± 0.045
			$k_m = 50$		
3	28.1 ±2.424	1.648 ± 0.088	3 ± 0.000	49.033 ± 0.508	0.333 ± 0.000
5	29.1 ±1.286	1.632 ± 0.067	5 ± 0.000	49.420 ± 0.416	0.200 ± 0.000
7	29.8 ±1.549	1.578 ± 0.066	7 ± 0.000	49.500 ± 0.318	0.171 ± 0.090
			$k_m = 75$		
3	45.6 ±1.837	1.599 ± 0.082	3 ± 0.000	74.033 ± 0.508	0.433 ± 0.161
5	47 ±2.494	1.545 ± 0.068	5 ± 0.000	74.420 ± 0.416	0.280 ± 0.103
7	47.3 ±2.830	1.512 ± 0.073	7 ± 0.000	74.500 ± 0.318	0.157 ± 0.045
			$k_m = 100$		
3	61.8 ±3.521	1.535 ± 0.067	3 ± 0.000	99.033 ± 0.508	0.333 ± 0.000
5	64.3 ±3.128	1.480 ± 0.064	5 ± 0.000	99.420 ± 0.416	0.320 ± 0.140
7	65.6 ±3.893	1.456 ± 0.082	7 ± 0.000	99.500 ± 0.318	0.157 ± 0.045

Table 4. Quantitative characteristics for pattern sets mined on $\overline{\mathcal{D}}_m^i$

f	3					5					7				
k_s	10	25	50	75	100	10	25	50	75	100	10	25	50	75	100
$\hat{\mathcal{D}}_m$, chi	4	0	11 ∘	3	4	5	1	14 ∘	1	4	4	1	13 ∘	3	1
$\hat{\mathcal{D}}_m$, random	41	41	44	43	53 ∘	53	**56**	50	**59** ∘	47	44	50	**54**	**58** ∘	51
$\hat{\mathcal{D}}_m$, rank	**51** ∘	43	39	**49**	34	**54**	51	**52**	52	**61** ∘	44	**54** ∘	52	52	**52**
$\hat{\mathcal{D}}_m$, top	44	**52** ∘	**49**	40	35	52 ∘	44	47	45	37	**49** ∘	44	38	39	36
$\overline{\mathcal{D}}_m$, chi	8	8	7	6	13 ∘	9	7	17 ∘	7	11	7	7	10 ∘	7	10
$\overline{\mathcal{D}}_m$, random	28	50 ∘	44	43	34	34	44 ∘	34	38	30	25	48 ∘	44	38	41
$\overline{\mathcal{D}}_m$, rank	38	26	42 ∘	34	32	29	24	26	35	36 ∘	40 ∘	32	33	30	36
$\overline{\mathcal{D}}_m$top	43 ∘	42	26	40	27	23	33 ∘	28	26	24	35 ∘	29	22	32	23
baseline	31	26	26	30	**56** ∘	29	28	20	25	38 ∘	40 ∘	23	22	29	38

Table 5. Total accuracy wins for aggregation techniques and the baseline approach for combinations of different k_m-values and f settings

in only one $\mathcal{T}h_k(\mathcal{D}_m)$. This indicates that even when using correlation measures, different data sets quickly lead to differing mining results. It is interesting to see that *overlap*, $|\Phi_{all}|/k_m$, and $\max_{p\in\Phi_{all}} \mu_{rank}(p)$ are rather stable for the $\overline{\mathcal{D}}_m$ setting for a given k_m, but depend on the value of f for the $\hat{\mathcal{D}}_m$ setting.

Qualitative results Given the findings above, the more interesting question is which of the proposed techniques select patterns which are useful features for classification. Again, we used an SVM and 10-fold cross-validation to estimate the quality of pattern sets. Inasmuch as differences in accuracy were almost never significant, we omit the actual accuracy estimates here. Instead we report how the different methods (each time a combination of $\mathcal{D}_m$ composition and selection method) compare given a fixed $k_m \in \{10, 25, 50, 75, 100\}$ and $f \in \{3, 5, 7\}$ (Table 5). Note that the table shows the total number of wins for each approach.

Each number denotes how often a particular technique has performed better than any other on any data set. We evaluated 9 techniques against each other on 8 data sets. Thus any given approach can have maximally 64 wins. Bold values denote the best-performing technique, given a k_m and value of f, while a circle (∘) shows for which k_m a technique performed best, given f.

The first, somewhat surprising, insight is that using large, overlapping $\mathcal{D}_m$, which should recreate phenomena over different mining situations, does not lead to good pattern selection. $\overline{\mathcal{D}}_m$ settings never perform best for a given k_m and usually perform better if only relatively few patterns are selected, suggesting that resampling does too little to counteract bias. Given that resampling forms the basis for, e.g., Bagging [2] techniques, we did not expect this outcome.

It is also noticeable that the standard approach produces suboptimal pattern sets. Only once is this baseline approach best, for $f = 3$, meaning relatively large folds where *informed* selection techniques such as *count* and *rank* do not enjoy a large advantage. Even there it is closely followed by the random selection - essentially the least informed one. This means that an unwritten paradigm of data mining (using large amounts of data to the fullest leads to meaningful patterns) turns out to be questionable in this case.

The random technique is the big winner of the entire comparison, given its simplicity. While reducing redundancy entirely by chance, it performs well in 4 of 15 settings. It is outperformed by *rank* (7 wins), but *count* is weaker (3 wins). Moreover, adding up *all* wins by technique, *random* outperforms *rank* and *count*, slightly for $\hat{\mathcal{D}}_m$ settings, more pronounced for $\overline{\mathcal{D}}_m$. So the information which

patterns generalize well over different subsets does not give a strong advantage in our case study. However, the variety of patterns caused by several subsets is helpful. Re-evaluating patterns' χ^2 score does again not work satisfactory.

4 Conclusions

In this work, we investigated ways of using data for pattern mining to produce good features for classification of complex data. Two main insights arise from the experimental evaluation: 1) usual assumptions on how to best use data in data mining turn out to be questionable. Neither the standard data mining setting (using large data sets to smooth over-fitting effects), nor a single mining and validation set, nor re-sampling techniques producing overlapping mining sets to uncover true underlying phenomena proved to be the most effective use. The best usage we observed consisted of splitting data into small, *independent* subsets instead, mining patterns on these and evaluating those patterns' generalization capability on *different* subsets. 2) the actual selection method matters far less than could be expected. Given a large enough variety of patterns, picking patterns at random proved to be rather effective, as proved the average rank selector, which picks patterns that were highly ranked at least once, even if not in *all* subsets. Using a validation set (either independent or involving the data patterns were mined from) for reassessing the χ^2-score did not work satisfactory.

An unexpected boon of these results is that pattern mining can apparently be easily parallelised without having to fear the loss of valuable information in terms of patterns. Quite contrary, we have seen that merging pattern sets extracted from small independent data sets improves the merit of the found patterns.

There are still several open questions to pursue w.r.t. the evaluated techniques. As we have observed, the interplay between k_m and k_v for the validation set technique has an effect on the composition of resulting pattern sets, and different k_m seem to favor certain aggregation techniques. It would therefore be valuable to perform stability studies, e.g. investigating whether final pattern sets stabilize for a certain value of k_m. Additionally, there are potential further selection criteria which time and space constraints did not allow us to investigate.

References

1. J. Boulicaut and B. Jeudy. Mining free itemsets under constraints. In M. E. Adiba, C. Collet, and B. C. Desai, editors, *IDEAS*, pages 322–329, 2001.
2. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
3. B. Bringmann and A. Zimmermann. The chosen few: On identifying valuable patterns. In *ICDM*, pages 63–72. IEEE Computer Society, 2007.
4. B. Bringmann, A. Zimmermann, L. De Raedt, and S. Nijssen. Don't be afraid of simpler patterns. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *PKDD*, pages 55–66. Springer, 2006.
5. T. Joachims. Making large-scale support vector machine learning practical. In *Advances in kernel methods: support vector learning*, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.
6. S. J. Swamidass, J. H. Chen, J. Bruand, P. Phung, L. Ralaivola, and P. Baldi. Kernels for small molecules and the prediction of mutagenicity, toxicity and anticancer activity. In *ISMB (Supplement of Bioinformatics)*, pages 359–368, 2005.