I-prune: Item selection for associative classification

I-prune: Item Selection for Associative Classification

Elena Baralis
Politecnico di Torino - Italy
elena.baralis@polito.it
Paolo Garza
Politecnico di Milano - Italy
garza@elet.polimi.it

Abstract

Associative classification is characterized by accurate models and high model generation time. Most time is spent in extracting and post-processing a large set of irrelevant rules, which are eventually pruned. We propose I-prune, an item pruning approach that selects uninteresting items by means of an interestingness measure and prunes them as soon as they are detected. Thus, the number of extracted rules is reduced and model generation time decreases correspondingly. A wide set of experiments on real and synthetic datasets has been performed to evaluate I-prune and to select the appropriate interestingness measure. The experimental results show that I-prune allows a significant reduction in model generation time, while increasing (or at worst preserving) model accuracy. Experimental evaluation also points to the chi-square measure as the most effective interestingness measure for item pruning.

1 Introduction

Given a collection of labelled data, the classification task is the generation of an abstract model of the classes defined by the class labels. The abstract model is a classifier, which may be used to predict the class label for unlabelled (i.e., previously unseen) data. Many different approaches have been proposed to build accurate classifiers, based on decision trees, naive-Bayes classification, neural networks, association or classification rules, and statistical approaches [24]. In the recent past, many associative classification techniques that yield accurate models for structured data have been proposed [2, 3, 7, 8, 9, 13, 18, 19, 20, 25, 27, 30, 32].

In the context of associative classification, the classification task can be reduced to the extraction and selection of the most appropriate set of association rules for the classifier. Association rules for classification are expressed in the form $X \rightarrow C$, where X is a set of items, while C is a class label. Each tuple d in the training dataset can be described as a collection of pairs *(attribute, value)*, plus a class label. Each pair *(attribute, value)* will be denoted as *item* in the paper. A tuple d matches a collection of items X when $X \subseteq d$. The quality of an association rule is usually measured by two parameters, its support, given by the number of tuples matching $X \cup C$ over the number of tuples in the dataset, and its confidence given by the number of tuples matching $X \cup C$ over the number of tuples matching X.

While associative classifiers yield accurate models for structured data, they usually require more computation time to generate the classification model with respect to traditional rule based classifiers (e.g., [11, 21]). Building an associative classifier usually requires two main steps: (1) Classification rule mining and (2) classification rule selection (i.e., rule pruning). During the rule mining phase, all frequent classification rules are extracted. Rule pruning is typically a post-processing phase that discards uninteresting classification rules [3, 12, 17, 19, 20, 25, 32]. Recently, new approaches based on only one step have been proposed [8, 9, 13, 30]. Interesting classification rules are directly mined by pushing some of the selection constraints in the rule mining algorithm. The model building phase is usually slower for classifiers based on a separate post-processing step.

The number of extracted rules, independently of the considered associative classifier, affects both rule mining and rule pruning time. Hence, the extraction and the analysis of many irrelevant classification rules may significantly impact on the model generation time. In this paper, we propose a filtering approach that is integrated in the classification rule mining step and avoids the generation of rules which would typically be discarded by the classification rule selection criteria.

Based on the observation that uninteresting items usually generate irrelevant rules, we propose I-prune, a "clever" item selection technique, which prunes uninteresting items during the rule mining phase. I-prune allows a significant reduction of the model generation time, while increasing (or preserving) the accuracy of the generated model. Since an item may be uninteresting for a class, but interesting for another, the proposed approach selectively prunes different items in different classes. Item interestingness is evaluated by means of several correlation measures (e.g., Chi-square, Yule's Y, Lift), whose effectiveness has been thoroughly evaluated with a large set of experiments. The Chi-square has been identified as the most effective measure in detecting uninteresting items.

Given the subset of frequent and interesting items for a class, only the rules exclusively including interesting items are extracted and considered in the following classification rule selection phase. This approach, which can be easily and efficiently integrated in any classification rule mining algorithm, allows the exclusion of (potentially) irrelevant rules from the model building process. However, analogously to attribute (feature) selection techniques (e.g., [14, 15]) and classification rule pruning techniques (e.g., [19, 20]), the proposed approach may unintentionally prune also interesting rules, which would have been generated by greedily pruned uninteresting items.

The effectiveness of I-prune has been thoroughly evaluated by means of a wide set of experiments, both on real and synthetic datasets. The experimental results show the positive effect of the item pruning approach, both on accuracy and model generation time. In particular, I-prune, coupled with the Chi-square measure, yields on average a time reduction around 30%, while also increasing the accuracy of the generated classification model.

The paper is organized as follows. Section 2 compares our work with previous work on attribute selection and rule pruning, while Section 3 describes the I-prune filtering approach and Section 4 describes its integration in the current associative classification algorithms. In Section 5 the proposed item pruning approach is evaluated by means of a large suite of experiments. Finally, Section 6 draws conclusions.

2 Previous related work

Attribute (or feature) selection techniques [14, 15] look for irrelevant or redundant attributes, which are then pruned. Differently from attribute selection techniques, we propose a technique to prune items, i.e., pairs *(attribute, value)*. Hence, our approach performs a finer granularity pruning with respect to traditional attribute selection techniques. Furthermore, it adapts to each class distribution, by selectively pruning different item subsets in each class. Traditional attribute selection techniques allow the selection of different attributes for each class only if a model for each class is generated. I-prune, instead, generates a single model. Hence, it requires less time, because only one model is generated instead of $|C|$, where $|C|$ is the number of classes.

Traditional attribute selection techniques cannot be straightforwardly adapted to prune items instead of attributes. In particular, by generating one binary variable for every (attribute,value) pair, the number of attributes grows excessively and traditional attribute selection techniques cannot be applied. However, since traditional attribute selection techniques and our approach operate at different pruning granularities, they may be applied jointly. In particular, attribute selection techniques are applied first, to prune useless or redundant attributes. Next, I-prune prunes useless values of unpruned attributes (i.e., useless items).

In the context of associative classification many techniques have been proposed to select high quality classification rules [3, 4, 12, 17, 19, 20, 22, 25, 26, 31, 32]. These greedy approaches, similarly to our filtering approach, potentially discard a subset of interesting rules. However, these approaches post-process classification rules after the extraction task. Hence, model generation may require a significant processing time. We propose a filtering approach integrated in the rule extraction algorithm, which prunes uninteresting items to prevent the extraction of irrelevant rules. Thus, model generation time is significantly reduced. As discussed in Section 4, our approach may be profitably integrated in many extraction algorithms (e.g., CBA [20], CMAR [19], L^3 [3]).

Recently, in order to reduce the model generation time, new associative classifiers have been proposed [8, 9, 10, 13, 29, 30]. These classifiers directly mine interesting classification rules to be included in the model by merging the mining and pruning steps in one single step. This goal is achieved by pushing a set of constraints in the rule mining algorithms, or by applying the filtering rule selection criteria immediately before storing the extracted rules on disk. These approaches allow reducing the search space and hence the model generation time. Our item filtering approach can be profitably integrated also in these associative classification algorithms, thus allowing to further reduce the search space.

Associative classifiers are usually eager classifiers, because a general model is "statically" generated on the training data. Lazy associative classifiers (e.g., [18, 27, 28]) instead generate a new model for each test data (i.e., mine classification rules specifically for a given test instance). Since in this case the rule mining phase is executed multiple times (one time for each new test instance), our filtering approach can be exploited by lazy associative classifiers to significantly reduce the mining phase, and hence the classification time for each test instance.

3 Item selection

The first step of association rule mining algorithms usually selects frequent items, i.e., items that satisfy the minimum support threshold on the whole dataset. Only these items can potentially generate frequent rules. In the case of classification rule mining, only frequent rules whose

	c	$\overline{c}$	$\sum_{row}$
i	$f_{i,c}$	$f_{i,\overline{c}}$	f_i
$\overline{i}$	$f_{\overline{i},c}$	$f_{\overline{i},\overline{c}}$	$f_{\overline{i}}$
$\sum_{column}$	f_c	$f_{\overline{c}}$	N

$$f_i = sup(i) \qquad f_{\overline{i}} = N - sup(i)$$
$$f_c = sup(c) \qquad f_{\overline{c}} = N - sup(c)$$
$$f_{i,c} = sup(i \cup c) \qquad f_{i,\overline{c}} = sup(i) - sup(i \cup c)$$
$$f_{\overline{i},c} = sup(c) - sup(i \cup c) \qquad f_{\overline{i},\overline{c}} = N + sup(i \cup c) - sup(i) - sup(c)$$

Table 1: Contingency table for a generic pair (i, c) (where i is an item and c a class label)

consequent is a class label are considered. Hence, only items which are frequent with respect to at least one class are selected.

During the first step, our item filtering technique also enforces a correlation constraint. In particular, an item is considered interesting, and hence selected, if and only if it is both frequent and correlated with respect to at least one class. Given an interestingness measure and a filtering threshold value, an arbitrary item i is correlated with respect to a class c if *interestingness_measure(i,c)>filtering_threshold*. Many different measures may be exploited by I-prune. Item interestingness measures are further discussed in Section 3.1. Uninteresting items are discarded as soon as they are detected. Thus, the number of extracted rules is reduced and model generation time decreases correspondingly.

Given the subset of frequent and interesting items for a class c, rules are generated for c. Hence, for each class c, only the rules including interesting items are extracted and considered in the following classification rule selection phase. As discussed in Section 2, this approach can be easily and efficiently integrated in many classification rule mining algorithms. As an example, the implementation of I-prune in a prefix-tree based classification algorithm is discussed in Section 4.

The objective of the I-prune item filtering approach is the greedy selection of a subset of interesting items, from which a set of high quality classification rules may be built by means of traditional heuristic approaches. In the classification context, since the whole space of possible solutions is usually not exhaustively analyzable, heuristics are exploited to prune the search space. Some heuristic approaches prune the initial data set (e.g., feature selection algorithms are used to prune potentially irrelevant attributes [14, 15]), while other approaches prune the extracted classification rule sets (e.g., pessimistic error rate based pruning and database coverage [20]). These approaches are usually based on heuristic criteria which are empirically shown to increase classification accuracy. Similarly, I-prune exploits a heuristic criterion, the correlation of items with classes, to prune the search space. This interestingness constraint is applied on the initial set of items to perform early pruning of potentially uninteresting items. Hence, analogously to feature selection techniques and classification rule pruning techniques, I-prune may unintentionally prune also items which might yield useful rules. The effectiveness of our pruning technique is experimentally evaluated in Section 5.

3.1 Measuring item interestingness

Many measures have been proposed for assessing the interestingness of association rules, classification rules, and patterns (e.g., [6], [19], [23]). We considered (a subset of) the most frequently used interestingness measures in the context of association rules to filter items. Among the measures reported in [6], [19], and [23], we selected symmetric measures because we are interested in correlated pairs (item, class). We also considered the Confidence and Conviction asymmetric measures because of their popularity.

We considered the following measures: Chi-square, Conviction, Yule's Q, Yule's Y, Odds

Measure name	Formula
Chi-square	$\frac{\left(f(i,c)-\frac{f(i)f(c)}{N}\right)^2}{\frac{f(i)f(c)}{N}}+\frac{\left(f(\bar{i},c)-\frac{f(\bar{i})f(c)}{N}\right)^2}{\frac{f(\bar{i})f(c)}{N}}+\frac{\left(f(i,\bar{c})-\frac{f(i)f(\bar{c})}{N}\right)^2}{\frac{f(i)f(\bar{c})}{N}}+\frac{\left(f(\bar{i},\bar{c})-\frac{f(\bar{i})f(\bar{c})}{N}\right)^2}{\frac{f(\bar{i})f(\bar{c})}{N}}$
Collective strength	$\frac{f(i,c)+f(\bar{i},\bar{c})}{f(i)f(c)+f(\bar{i})f(\bar{c})}\times\frac{N-f(i)f(c)-f(\bar{i})f(\bar{c})}{N-f(i,c)-f(\bar{i},\bar{c})}$
Conviction	$\frac{f(i)f(\bar{c})}{Nf(i,\bar{c})}$
Confidence	$\frac{f(i,c)}{f(i)}$
Cosine	$\frac{f(i,c)}{\sqrt{f(i)f(c)}}$
Jaccard	$\frac{f(i,c)}{f(i)+f(c)-f(i,c)}$
Kappa	$\frac{Nf(i,c)+Nf(\bar{i},\bar{c})-f(i)f(c)-f(\bar{i})f(\bar{c})}{N^2-f(i)f(c)-f(\bar{i})f(\bar{c})}$
Lift/Interest	$\frac{Nf(i,c)}{f(i)f(c)}$
Odds ratio	$\frac{f(i,c)f(\bar{i},\bar{c})}{f(i,\bar{c})f(\bar{i},c)}$
ϕ-coefficient	$\frac{Nf(i,c)-f(i)f(c)}{\sqrt{f(i)f(c)f(\bar{i})f(\bar{c})}}$
Piatetsky-Shapiro's	$\frac{f(i,c)}{N}-\frac{f(i)f(c)}{N^2}$
Yule's Q	$\frac{f(i,c)f(\bar{i},\bar{c})-f(i,\bar{c})f(\bar{i},c)}{f(i,c)f(\bar{i},\bar{c})+f(i,\bar{c})f(\bar{i},c)}$
Yule's Y	$\frac{\sqrt{f(i,c)f(\bar{i},\bar{c})}-\sqrt{f(i,\bar{c})f(\bar{i},c)}}{\sqrt{f(i,c)f(\bar{i},\bar{c})}+\sqrt{f(i,\bar{c})f(\bar{i},c)}}$

Table 2: Definition of all considered interestingness measures

ratio, Kappa, Lift, ϕ-coefficient, Cosine, Piatetsky-Shapiro's, Collective strength, Jaccard, and Confidence. While the formal definition of the Chi-square measure, in the context of classification rules, is reported in [19], the definition of all the other measures is reported in [23]. All the considered measures can be easily computed by exploiting the 2x2 contingency table represented in Table 1. Their definition is provided in Table 2.

To select the items to be pruned, for each pair *(item,class)* our approach computes the corresponding contingency table and checks if the considered item and class are correlated with respect to the used measure. We experimentally evaluated the effectiveness of all the considered measures. The results of our experiments, discussed in Section 5, show that the Chi-square is the most effective pruning measure, because it both reduces the search space (and consequently the computation time), and on average increases accuracy.

4 I-prune filtering algorithm

The I-prune filtering technique can be easily and efficiently integrated in many classification rule mining algorithms (e.g., [19], [20]). In this section, we discuss how it may be integrated in a prefix-tree [16] based classification rule mining algorithm.

Item pruning is performed by the *frequentInterestingItems* function (Algorithm 4.1). For each item, its frequency and correlation with respect to each class are analyzed (lines 4-11). Given a class c, when an item is characterized by (i) a support count local to class c above the minimum support threshold (line 7) and (ii) an interestingness (i.e., correlation) measure with respect to c above the input threshold (line 7), the item is considered correlated to the class. Interesting items are the items which are frequent and correlated with respect to at least one class (lines 13-17).

Algorithm 4.1 Function frequentInterestingItems

Input: Set of items (I)
Input: Set of classes (C)
Input: Minimum support threshold (min_sup)
Input: Interestingness measure ($int\_measure$)
Input: Interestingness filtering threshold (th)
Output: Items which are frequent and pass the interestingness threshold in at least one class.

```
for all item i in Items do
   i.correlatedClasses=∅; /*Empty set*/
end for
/*Compute for each item the classes with respect to which it is frequent and correlated*/
for all item i in Items do
   for all class c in C do
      /*Check if i is frequent and correlated with respect to c*/
      if (i.classSupportCounts[c] > min_sup
         and int_measure(i, c) > th) then
         Insert c into i.correlatedClasses;
      end if
   end for
end for
/*Select interesting items (i.e, frequent and correlated with respect to at least one class)*/
SelectedItems=∅; /*Empty set*/
for all item i in Items do
   if (i.correlatedClasses is not empty) then
      Insert i into SelectedItems;
   end if
end for
return SelectedItems;
```

Algorithm 4.2 I-prune: Extraction of classification rules with item filtering

Input: Dataset (D)
Input: Interestingness measure ($int\_measure$)
Input: Interestingness filtering threshold (th)
Input: Minimum support threshold (min_sup)
Output: Frequent classification rules including interesting items only

/**Count global support and class support for each item in D**/
$Items$=countItemSupports(D);
/**Count global frequency of each class**/
C=setOfClasses(D);
/**Select frequent items which pass the interestingness filtering threshold**/
$SelectedItems$=frequentInterestingItems($Items$, C, min_sup, $int\_measure$, th);
/**Create FP_tree for D using only the items in SelectedItems**/
$FPTree$=createFPTree(D,$SelectedItems$);
/**Call the recursive classification rule mining procedure**/
mineClassificationRules($FPTree$, null, min_sup);

Algorithm 4.3 Procedure mineClassificationRules

Input: FP_Tree (T)
Input: Conditional_path (α)
Input: Minimum support threshold (min_sup)

ht=header table of tree T;
for all item i in ht (in decreasing support rank) **do**
  for all class c in C **do**
    if ($ht.support(i \cup c) > min\_sup$
      and $c \in i.correlatedClasses$ and
      $(\forall j \in \alpha, c \in j.correlatedClasses)$) **then**
      save the rule $\alpha \cup i \rightarrow c$ on disk;
    end if
  end for
  Generate pattern $\beta = \alpha \cup i$ with global and class support counts equal to i support counts;
  Build β's conditional pattern base;
  Build β's conditional pattern base header table ht_β;
  /**Remove items which, with respect to at least one class, are not frequent in ht_β or not correlated**/
  for all item i in ht_β **do**
    if ($\nexists c | ht_\beta.support(i \cup c) > min\_sup$
      and $c \in i.correlatedClasses$) **then**
      remove i from ht_β;
    end if
  end for
  Create β's conditional FP_tree T_β by using only the items included in ht_β;
  mineClassificationRules(T_β, β, min_sup);
end for

The item filtering function fits in the general framework of the classification algorithm outlined in Algorithm 4.2. During the first scan of the database, Algorithm 4.2 computes the frequency of each item in the database (global count) and in each class c (class counts) (line 1). Furthermore, it computes the global frequency of each class (line 2). Then, for each item, the set of classes with respect to which the item is correlated is computed by analyzing the stored support counts. Only items which are both frequent and correlated with respect to at least one class are selected by the *frequentInterestingItems* function (line 3) and used to build the initial FP-tree (line 4). Finally, the recursive rule mining procedure is invoked (line 5).

The rule mining procedure *mineClassificationRules* (Algorithm 4.3) exploits the concept of interestingness during rule extraction. More specifically, every test on the minimum frequency of an arbitrary item i with respect to an arbitrary class c is extended with a test on the correlation of i with c (lines 4 and 12). An itemset generates a new rule for a class c (line 5) only if (i) it is frequent and (ii) all its items are correlated with respect to the current class (line 4). Correlation between each item and all classes is only computed once on the original dataset by the *frequentInterestingItems* function (see Algorithm 4.2, line 3) and is not recomputed during the recursive classification rule mining phase.

The proposed item filtering approach can be easily integrated also in Apriori [1] based extraction algorithms. In this case, when candidate itemsets are generated, every test on the minimum frequency of the itemset is extended with a correlation test. In particular, a candidate itemset is considered only if (i) it is frequent with respect to at least one class, and (ii) it exclusively includes items correlated with the same class.

To validate our approach, we have introduced I-prune in a CBA-like classification algorithm where, for the sake of efficiency, the core rule extraction task is performed by means of the prefix-tree based rule extraction algorithm described above instead of Apriori.

5 Experimental results

The main contribution of I-prune consists in reducing model generation time, while increasing (or preserving) the accuracy of the generated model. To verify the effectiveness of the proposed approach, we performed a set of experiments both on real and synthetic datasets. The experiments addressed the following issues.

Accuracy of the classifier. The effect of the I-prune filtering approach on the average accuracy of the generated classification models is analyzed in Section 5.1. Accuracy depends both on the selected interestingness measure and the filtering threshold. This set of experiments allowed us to identify the Chi-square as the most appropriate interestingness measure for I-prune. Based on their effect, measures have been classified in three classes: measures with a positive, negative or no impact on classification accuracy for most datasets.

Model generation time and pruning effect. In the range of filtering threshold values that yield accurate classification models, most measures allow significantly reducing the average training time without impacting on the accuracy of the classification models (see Section 5.2). Since each measure selects different items for pruning, the number of pruned items and rules strongly depends on the considered measure.

Scalability. The scalability of the proposed approach has been analyzed in Section 5.3 by means of a set of experiments on synthetic datasets. Both scalability in the cardinality of the dataset and in the number of attributes have been analyzed.

Dataset	Transactions	Attributes		Classes	Items
		Continuous	Categorical		
Anneal	998	6	32	6	71
Australia	690	6	8	2	49
Auto	205	15	10	7	106
Breast-w	699	10	0	2	29
Census	30162	0	16	2	145
Cleve	303	5	8	2	27
Crx	690	6	9	2	53
Diabetes	768	8	0	2	15
Flare	1066	0	10	2	31
German	1000	7	13	2	58
Glass	214	9	0	7	20
Heart	270	6	7	2	18
Hepatitis	155	6	13	2	33
Horse	368	7	21	2	61
Hypo	3163	7	18	2	53
Ionosphere	351	34	0	2	144
Iris	150	4	0	3	12
Labor	57	8	8	2	29
Lymph	148	3	15	4	59
Mushroom	8124	0	22	2	116
Nursery	12960	0	8	5	27
Pendigits	10992	16	0	10	165
Pima	768	8	0	2	15
Segment	2310	19	0	7	171
Shuttle-s	58000	9	0	7	1109
Sick	4744	7	22	2	58
Sonar	208	60	0	2	42
Soybean s.	47	0	35	4	72
TicTacToe	958	0	9	2	27
Vehicle	846	18	0	4	71
Voting	433	0	16	2	48
Waveform	5000	21	0	3	106
Wine	178	13	0	3	37
Yeast	1484	8	0	10	21
Zoo	101	0	16	7	34

Table 3: UCI datasets characteristics

<table>
<tr><th colspan="2">Interestingness measure</th><th colspan="3">Best average accuracy value</th><th>Tuned threshold value</th></tr>
<tr><th>Group</th><th>Name</th><th>Avg acc%</th><th>Threshold value</th><th>Filter vs without filter win/draw/loss</th><th>Avg acc%</th></tr>
<tr><td></td><td>Without filter</td><td>86.88</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td rowspan="3">1</td><td>Chi-square</td><td>87.14</td><td>0.455</td><td>12/15/8</td><td>88.08</td></tr>
<tr><td>Conviction</td><td>86.90</td><td>0.8</td><td>8/21/6</td><td>87.69</td></tr>
<tr><td>Yule’s Q</td><td>86.90</td><td>-0.6</td><td>10/15/10</td><td>87.70</td></tr>
<tr><td rowspan="8">2</td><td>Yule’s Y</td><td>86.95</td><td>-0.2</td><td>8/13/14</td><td>87.77</td></tr>
<tr><td>Odds ratio</td><td>86.90</td><td>0.6</td><td>8/11/16</td><td>87.72</td></tr>
<tr><td>Kappa</td><td>86.89</td><td>-0.1</td><td>7/16/12</td><td>87.82</td></tr>
<tr><td>Lift/Interest</td><td>86.89</td><td>0.6</td><td>9/15/11</td><td>87.79</td></tr>
<tr><td>ϕ-coefficient</td><td>86.88</td><td>-0.04</td><td>5/21/9</td><td>87.65</td></tr>
<tr><td>Cosine</td><td>86.88</td><td>0.1</td><td>4/23/8</td><td>87.55</td></tr>
<tr><td>Piatetsky-Shapiro’s</td><td>86.88</td><td>-0.04</td><td>5/21/9</td><td>87.89</td></tr>
<tr><td>Collective strength</td><td>86.78</td><td>0.5</td><td>7/20/8</td><td>87.49</td></tr>
<tr><td rowspan="2">3</td><td>Jaccard</td><td>86.89</td><td>0.01</td><td>2/30/3</td><td>87.63</td></tr>
<tr><td>Confidence</td><td>86.86</td><td>0.02</td><td>2/30/3</td><td>87.85</td></tr>
</table>

Table 4: Average accuracy for each interestingness measure on the UCI datasets

We ran the experiments on 35 datasets of the UCI Machine Learning Repository [5], whose main characteristics are reported in Table 3. Besides the most popular datasets (see [3, 19, 20]), we also considered some larger datasets (e.g., census, shuttle-s). Continuous attributes have been discretized by means of the same technique used by CBA [20]. The synthetic datasets for the scalability experiments have been generated by means of the IBM data generator[1]. The experiments were performed on a 3.2GHz Pentium IV PC with 2.0GB main memory, running Kubuntu 6.06.

5.1 Accuracy

Accuracy measures the ability of the classifier to correctly classify unlabeled data. It is the ratio of the number of correctly classified data over the cardinality of the dataset (including correct and wrong classifications, plus unclassified data). Accuracy has been computed by using a 10 fold cross validation test.

We performed the experiments by considering the 13 measures in Column (2) of Table 4. For each measure, we performed a set of experiments to analyze the accuracy behavior when varying the interestingness measure and the filtering threshold value.

We also performed experiments (not reported here) to compare our approach with the attribute selection techniques in [15]. On average, the accuracy improvement given by I-prune is comparable to that of attribute selection techniques. Since the two approaches operate at a different pruning granularity, they might be profitably exploited together to generate higher quality classifiers. The joint application of the two pruning techniques will be explored as future work.

[1] http://www.almaden.ibm.com/software/quest/Resources

Dataset	Without filter	Accuracy% Chi-square interestingness measure	
		with threshold value = 0.455	with tuned threshold value
Anneal	98.0	**98.1**	98.1
Australia	**85.9**	85.1	86.4
Auto	75.6	**78.1** •	82.0 •
Breast-w	95.9	**96.0**	96.7
Census	84.2	84.2	84.2
Cleve	81.9	**83.2**	83.8
Crx	84.8	**86.1** •	86.5 •
Diabetes	**78.7**	78.5	78.7
Flare	**99.4**	99.1	99.5
German	73.3	**75.1** •	75.1 •
Glass	75.2	75.2	75.2
Heart	83.7	83.7	83.7
Hepatitis	83.2	**83.9**	84.5
Horse	**83.7**	82.9 ◦	84.0
Hypo	96.2	96.2	96.6
Ionosphere	92.0	**92.3**	93.7
Iris	93.3	93.3	94.7
Labor	93.0	93.0	96.5
Lymph	83.1	**85.8**	86.5
Mushroom	96.8	**98.0**	98.0
Nursery	77.0	77.0	83.2 •
Pendigits	93.1	93.1	93.2
Pima	78.3	78.3	78.3
Segment	**93.3**	93.0	93.3
Shuttle-s	99.7	99.8 •	99.8 •
Sick	**95.5**	95.4	95.6
Sonar	78.4	78.4	79.3
Soybean s.	97.9	97.9	100.0
TicTacToe	99.5	99.5	99.5
Vehicle	**71.4**	71.2	71.5
Voting	93.8	**94.5**	95.6
Waveform	81.5	81.5	81.8
Wine	97.8	97.8	97.8
Yeast	**58.6**	56.9	58.6
Zoo	87.1	**88.1**	91.1
Average	*86.88*	*87.14*	*88.08*

Table 5: Detailed accuracy results for the Chi-square measure on the UCI datasets. • (◦) is reported near the accuracy value when the increase (decrease) with respect to the standard classifier is statistically significant

Best accuracy. For each measure, Table 4 reports the best average accuracy value (Column (3)) and the corresponding filtering threshold value (Column (4)) obtained with standard values for the minimum support and confidence thresholds (minsup=1%, minconf=50%). The same filtering threshold value has been used for all datasets.

The first row of Table 4 reports the accuracy of the model generated without item filtering. We refer to this configuration as the standard classifier. With respect to the standard classifier, 8 measures allow generating (slightly) more accurate models (improvement between +0.01% and +0.26%), 3 measures have no impact on the average classification accuracy, while only 2 measures generate slightly less accurate models (between -0.02% and -0.10%). Hence, most measures have a mildly positive impact on the average accuracy of the generated models.

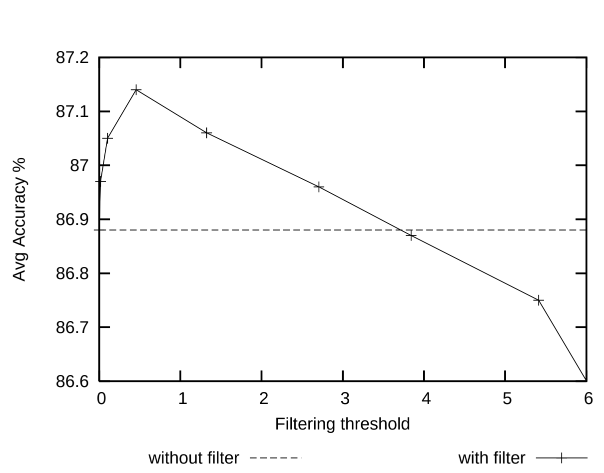
For the best value of each measure, we computed the number of datasets where the accuracy of the generated classifier is higher/equal/lower than the accuracy of the standard classifier. For each measure, Column (5) of Table 4 reports the number of wins, draws, and losses of the generated classifier with respect to the standard classifier. According to these results, measures can be divided in three groups (reported in Column (1)): (1) Measures with a positive impact on a significant number of datasets (e.g., Chi-square), (2) measures with a negative impact on a significant number of datasets (e.g., Yule's Y, Odds ratio), and (3) measures with no impact on most datasets (e.g., Jaccard, Confidence).

Measures in the first set properly prune, on average, useless items on many datasets. Accuracy is improved with respect to the standard classifier for a significant number of datasets, and also average accuracy is high. Differently, measures in the second set prune useful items for many datasets, while pruning uninteresting (or harmful) items for other datasets. These measures have a negative impact on accuracy for many datasets. However, some of these measures (e.g., Yule's Y) are characterized by an average accuracy on the 35 datasets that is higher than or equal to that of the standard classifier. This average improvement is yielded by significant improvements on few datasets (e.g., Yule's Y yields +1.4% on the sonar dataset), which allow balancing low accuracy on other datasets. Finally, measures in the third set prune few items and affect only few datasets. Hence, these measures are not effective for our purposes.

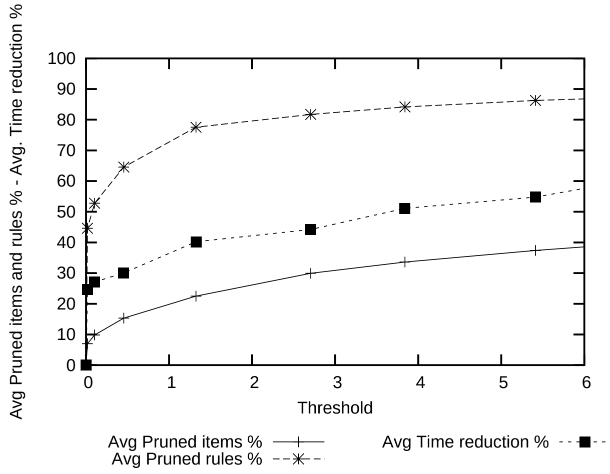
Average accuracy values in Column (3) of Table 4 were obtained by setting the filtering threshold to the same value for all datasets. Separately tuning the filtering threshold for each dataset yields higher accuracy. Column (6) of Table 4 reports, for each measure, the average accuracy on the 35 datasets when the filtering threshold is separately tuned for each dataset. Tuning yields an accuracy improvement between +0.61% and +1.20% with respect to the standard classifier. Hence, tuning to adapt to the dataset characteristics yields on average a significant accuracy improvement.

For the Chi-square measure, which provides the best average accuracy value, we also report in Table 5 the detailed accuracy values for all the 35 UCI datasets. The reported results show that the Chi-square measure already provides an accuracy improvement on many UCI datasets by setting the filtering threshold to the fixed value 0.455 for all datasets. If the filtering threshold is separately tuned for each dataset, the benefits of exploiting the Chi-square measure in I-prune become more evident.

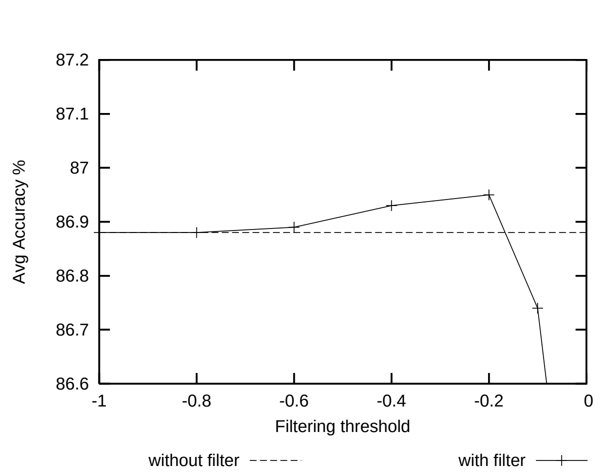
To validate the statistical significance of the accuracy improvement given by I-prune, we applied the paired t-test at level p=0.05 on all datasets. We compared I-prune with the chi-square item filtering measure against the standard classifier. In Table 5 the symbol $\bullet$ is reported near I-prune accuracy value when the improvement with respect to the standard classifier is statistically significant. Similarly, the symbol $\circ$ is reported when the decrease is statistically



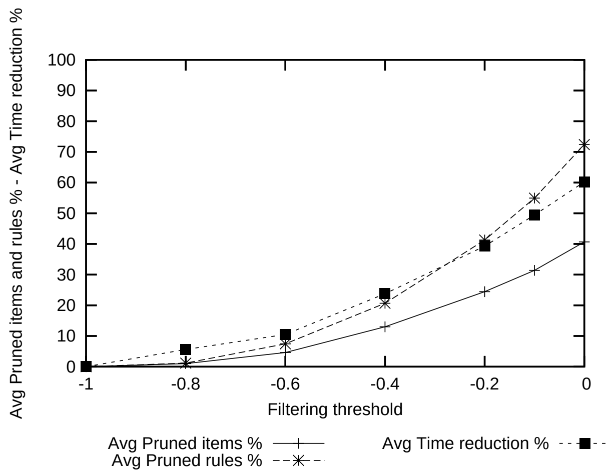
(a) Average accuracy when varying the Chi-square threshold



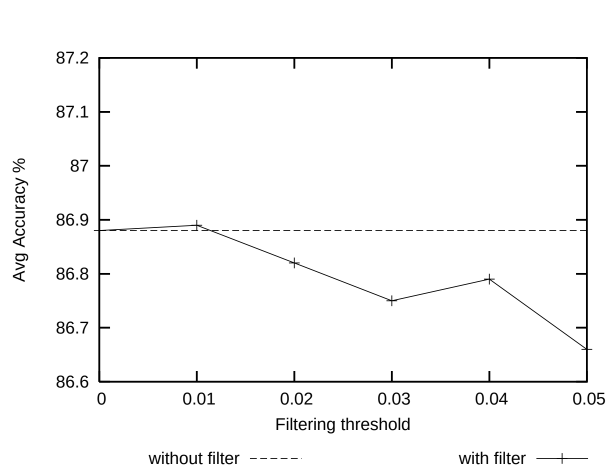
(b) Average pruned items, pruned rules and time reduction when varying the Chi-square threshold



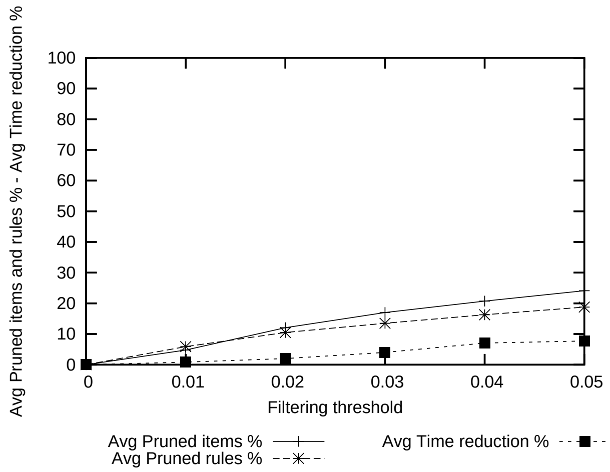
(c) Average accuracy when varying the Yule's Y threshold



(d) Average pruned items, pruned rules and time reduction when varying the Yule's Y threshold



(e) Average accuracy when varying the Jaccard threshold



(f) Average pruned items, pruned rules and time reduction when varying the Jaccard threshold

Figure 1: Average accuracy, training time reduction, percentage of pruned items, and percentage of pruned rules when varying the filtering threshold (minsup=1%, minconf=50%)

Interestingness measure			Avg accuracy improvement value range
Group	Name	Value range	
	Without filter	-	-
1	Chi-square	[0,+∞)	[0,3.841]
	Conviction	[0.5,+∞)	[0.5,0.83]
	Yule's Q	[-1,1]	[-1,-0.55]
2	Yule's Y	[-1,1]	[-1,-0.2]
	Odds ratio	[0,+∞)	[0,0.6]
	Kappa	[-1,1]	[-1,-0.1]
	Lift/Interest	[0,+∞)	[0,0.6]
	ϕ-coefficient	[-1,1]	[-1,-0.04]
	Cosine	[0,1]	[0,0.1]
	Piatetsky-Shapiro's	[-0.25,0.25]	[-0.25,-0.04]
	Collective strength	[0,+∞)	-
3	Jaccard	[0,1]	[0,0.01]
	Confidence	[0,1]	-

Table 6: Average accuracy improvement value range for each interestingness measure on the UCI datasets

significant. The statistical test confirms that I-prune allows increasing or preserving model accuracy.

Effect of the filtering threshold on accuracy. For each considered interestingness measure, column (3) of Table 6 reports the range of values characterizing the measure domain, while the value range yielding a positive impact on the average accuracy of the generated models is given in Column (4) of Table 6.

For one representative measure in each group, we also report the behavior of average accuracy when varying the filtering threshold. In particular, the selected measures are: (i) Chi-square, (ii) Yule's Y, and (iii) Jaccard. The Chi-square measure (see Figure 1(a)) provides a slight accuracy improvement with respect to the standard classifier for a wide range of filtering threshold values. The average accuracy rapidly increases for very low values of the filtering threshold and, after the value chi-square=0.455, slowly decreases. Hence, the Chi-square measure is characterized by a wide range of appropriate filtering threshold values.

Differently, for the Yule's Y measure reported in Figure 1(c)), average accuracy is close to the standard classifier for low values of the filtering threshold and it rapidly decreases for threshold values higher than -0.2. Hence, threshold value selection may significantly affect the accuracy of the generated classifier. The Jaccard measure, reported in Figure 1(e), has a rather weak impact on the average accuracy value when low filtering values are used. When the threshold value increases, average accuracy decreases.

Effect of the support threshold on accuracy. To analyze the effect of the support threshold, we considered the Chi-square measure, which is the best interestingness measure. We performed the analysis for three reference values of the filtering threshold: 0.455 (highest average accuracy value with support threshold 1%), 3.841 (same average accuracy of the standard classifier), and 2.148 (middle point of the previous interval).

As depicted in Figure 2, albeit with an offset, average accuracy shows the same behavior

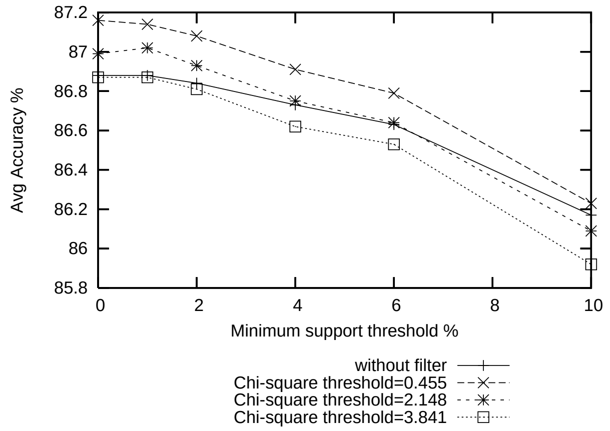


Figure 2: Effect of the minimum support threshold on average accuracy

for all three filtering threshold values. Hence, minimum support and filtering threshold may be tuned separately, thus significantly simplifying the tuning process. Furthermore, we observe that tuning the Chi-square threshold yields better accuracy results with respect to simply tuning the support threshold. Hence, correlation based filtering (e.g., Chi-square based) yields more accurate classifiers than frequency based filtering (e.g., support based).

5.2 Model generation

We analyzed the time reduction (in percentage) for classifier generation yielded by I-prune with respect to the standard classifier generation. To evaluate the pruning selectivity of our approach, we have also analyzed the corresponding average percentage of pruned items and pruned rules per class, i.e., items and rules pruned by I-prune during the rule mining step (before the rule selection performed by the post-processing step). All the experiments have been performed with minimum support threshold 1% and minimum confidence threshold 50%.

Column (4) in Table 7 reports the model generation time reduction for each measure, with filtering threshold set to the value that allows achieving the best average accuracy value (Column (3)). Most interestingness measures yield a significant reduction of the model generation time (from 15% to 45%) without negatively affecting the average accuracy. Only measures in group 3, which also have a negligible impact on accuracy, are characterized by a very low time reduction. Since measures in group 3 yield neither accuracy improvements, nor time reduction, they should not be considered for item pruning. Independently of the dataset, on average the model generation time reduction is related to the percentage of pruned items and rules. Table 7 also reports the average percentage of pruned items per class (Column (5)) and the average percentage of pruned rules per class (Column (6)) for each measure with filtering threshold set to the value in Column (3).

Detailed results on all UCI datasets are shown in Table 8 for the Chi-square measure, which provides the highest classification accuracy. Column (2) reports the model generation time of the standard classifier (i.e., the algorithm without item filtering), while Column (5) reports the model generation time reduction when the Chi-square measure, with the filtering threshold set to 0.455, is used. I-prune provides a significant reduction of the model generation time on almost all datasets. Only on few datasets (e.g., TicTacToe) the model generation time reduction is low. We also separately reported the execution time of the two steps of the model generation

Interestingness measure Group	Name	Threshold value	Avg time reduction%	Avg pruned items%	Avg pruned rules%
1	Chi-square	0.455	31.68	15.35	65.66
	Conviction	0.8	20.76	18.19	15.44
	Yule's Q	-0.6	27.79	16.11	27.37
2	Yule's Y	-0.2	39.00	24.46	41.29
	Odds ratio	0.6	46.02	29.59	51.76
	Kappa	-0.1	37.33	22.27	38.48
	Lift/Interest	0.6	17.06	18.14	18.63
	ϕ-coefficient	-0.04	29.10	15.76	28.17
	Cosine	0.1	6.01	16.90	15.55
	Piatetsky-Shapiro's	-0.04	28.89	15.76	28.17
	Collective strength	0.5	15.72	7.84	15.00
3	Jaccard	0.01	0.38	4.75	5.87
	Confidence	0.02	2.07	3.76	7.03

Table 7: Average model generation time reduction, and pruned items and rules for each interestingness measure on the UCI datasets

process: (a) rule mining and (b) rule selection. Column (3) and Column (4) of Table 8 report respectively the rule mining and the rule selection time of the standard classifier. Column (6) and Column (7) report the time reductions obtained by applying I-prune (Chi-square threshold set to 0.455). I-prune significantly reduces both execution times. Since during the first step a large percentage of rules is pruned, fewer rules have to be considered by the second step. Hence, on average, I-prune has a higher impact on the execution time of the rule selection step.

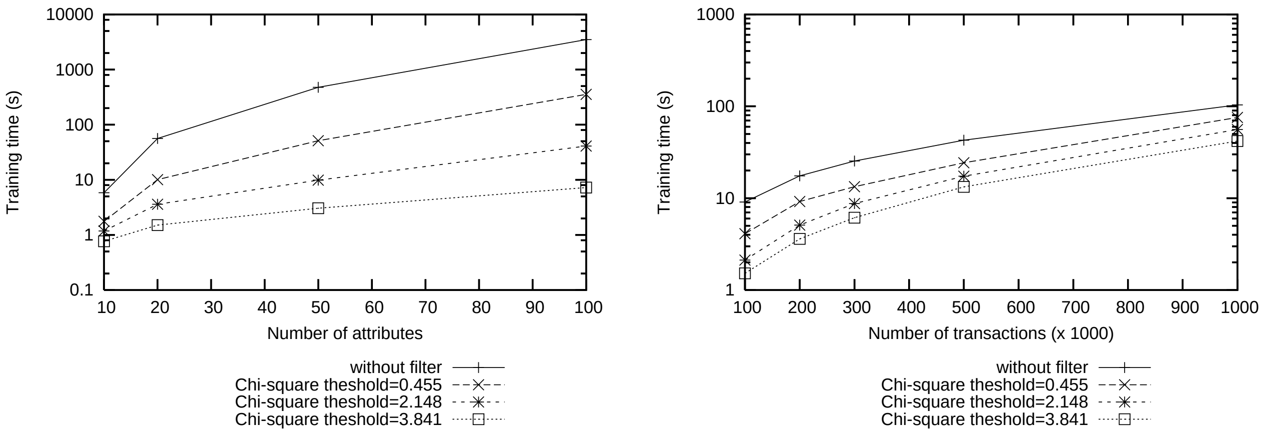
We applied the paired t-test at level p=0.05 on all datasets to validate the statistical significance of the time reduction yielded by I-prune. In Table 8 the symbol • is reported near I-prune time reduction value when it is statistically significant with respect to the standard classifier. The time reduction is statistically significant on 32 of datasets. Only for 3 datasets, characterized by a low model generation time and a low percentage of pruned items, the difference is not statistically significant.

In Table 8, the average percentage of pruned items per class (Column (8)) and the average percentage of pruned rules per class (Column (9)) are also reported. For the Chi-square measure the percentage of pruned rules (i.e., rules which are pruned during the rule mining step) is significantly higher than the percentage of pruned items. The Chi-square measure prunes items that are highly frequent and strongly correlated to each other. Hence, the number of pruned rules is high even when few items are pruned.

Figures 1(b), 1(d), and 1(f) report the average time reduction when varying the filtering threshold value for the three group representative measures Chi-square, Yule's Y, and Jaccard which were discussed in Section 5.1. The Chi-square measure is characterized by a significant time reduction (around 30%) also for low threshold values. Differently, for the Yule's Y measure, the time reduction is negligible for low threshold values and grows only for rather high threshold values. However, since high filtering threshold values may have a negative impact on average accuracy (see Figure 1(c)), the appropriate threshold should be chosen by means of a careful tuning process. The Jaccard measure has a negligible effect on time reduction in the whole value

Dataset	Without filter model generation time (s)			Chi-square measure, threshold value = 0.455				
				time reduction%			pruned	pruned
	total	rule mining	rule selection	total	rule mining	rule selection	items%	rules%
Anneal	0.3	0.2	0.1	30.6 •	5.0	81.8	14.7	65.1
Australia	4.1	3.6	0.5	54.9 •	61.1	10.2	17.7	59.8
Auto	3.4	2.7	0.7	47.5 •	46.9	49.9	22.9	92.2
Breast-w	0.3	0.2	0.1	6.1	7.5	3.3	6.5	50.6
Census	53.5	1.7	51.8	4.0 •	1.9	4.1	3.3	50.6
Cleve	0.7	0.6	0.1	43.9 •	41.3	59.5	13.5	87.1
Crx	4.0	2.5	1.5	55.7 •	54.9	75.1	16.5	69.6
Diabetes	0.1	0.08	0.02	9.4	10.8	4.0	12.4	75.0
Flare	0.1	0.08	0.02	16.3 •	12.5	31.5	27.9	85.0
German	8.6	6.7	1.9	71.3 •	76.2	53.9	25.0	85.0
Glass	0.1	0.09	0.01	14.5 •	5.9	92.0	22.7	82.4
Heart	0.2	0.1	0.1	17.8 •	10.0	25.6	18.2	93.8
Hepatitis	2.1	1.8	0.3	46.9 •	45.0	58.2	8.6	74.4
Horse	5.0	4.2	0.8	39.7 •	40.5	35.5	14.5	68.8
Hypo	15.6	5.9	9.7	60.5 •	66.2	57.0	18.7	85.3
Ionosphere	77.4	44.0	33.4	9.1 •	6.6	12.4	3.9	50.2
Iris	0.1	0.09	0.01	5.1 •	5.6	1.0	6.8	10.1
Labor	0.1	0.09	0.01	24.5 •	18.6	78.0	19.4	38.1
Lymph	1.4	1.1	0.3	50.2 •	39.9	88.0	24.2	77.8
Mushroom	4.6	2.9	1.7	23.0 •	26.0	17.9	2.5	71.4
Nursery	1.2	0.6	0.6	36.3 •	17.0	55.6	16.8	27.8
Pendigits	89.0	19.3	69.7	1.6 •	7.2	0.1	6.7	61.6
Pima	0.1	0.07	0.03	12.2 •	17.1	0.7	11.8	75.0
Segment	64.3	16.5	47.8	55.4 •	35.6	62.2	9.0	65.9
Shuttle-s	4.2	2.0	2.2	7.1 •	14.5	0.4	13.2	36.4
Sick	625.2	263.0	362.2	79.2 •	85.4	74.7	12.2	99.6
Sonar	76.7	64.2	12.5	64.9 •	69.0	43.8	48.2	99.9
Soybean s.	0.5	0.4	0.1	70.9 •	68.9	79.0	47.0	99.9
TicTacToe	0.5	0.4	0.1	2.0	1.3	4.7	1.1	0.1
Vehicle	17.3	11.2	6.1	32.8 •	34.8	29.1	8.0	47.2
Voting	7.1	5.0	2.1	34.0 •	37.2	26.5	5.6	52.7
Waveform	43.0	19.2	23.8	1.0 •	1.6	0.5	8.2	77.5
Wine	0.5	0.3	0.2	29.9 •	19.5	45.5	8.3	39.9
Yeast	0.3	0.27	0.03	21.1 •	22.2	11.0	29.1	80.7
Zoo	0.4	0.39	0.01	29.5 •	30.0	10.0	12.2	61.6
Average				*31.68*	*29.80*	*36.70*	*15.35*	*65.66*

Table 8: Model generation time reduction, and pruned items and rules for the Chi-square interestingness measure on the UCI datasets. • is reported near the time reduction value when it is statistically significant



(a) Training time when varying the number of attributes (50000 transactions)

(b) Training time when varying the number of transactions (9 attributes)

Figure 3: Scalability

range reported in Figure 1(f).

The average percentage of pruned items and rules per class are also reported in Figures 1(b), 1(d), and 1(f). Independently of the filtering threshold, the behavior of the Chi-square measure outlined by Table 8 is confirmed. In particular, the percentage of pruned rules is significantly higher than the percentage of pruned items, while for the Jaccard measure it is almost equal. This different behavior highlights that each measure prunes different items and hence different rules. The Chi-square measure prunes frequent items strongly correlated to each other. Thus, it prunes few items, but a large number of itemsets. The Jaccard measure, instead, prunes items with low frequency or weakly correlated. Hence, a reduced number of itemsets is pruned.

We finally analyzed the impact of I-prune on the size of the final model, generated by the rule selection (post-processing) phase. On average, the final model size is independent of the considered interestingness measure, and also of the considered threshold value. It is strongly reduced only when an excessive pruning is performed.

5.3 Scalability

We analyzed the scalability of I-prune by varying (a) the number of transactions (rows) and (b) the number of attributes in synthetic datasets. The datasets were generated by means of the IBM data generator[2] with classification function 2. For each dataset, two class labels are defined and each transaction is labeled by applying the classification function on the first nine attributes of the dataset. All other attributes are not correlated with the class attribute and are exploited to analyze the scalability of the classifier when varying the number of attributes. We performed the scalability test by using the most effective interestingness measure, i.e., Chi-square, with the three reference threshold values (0.455, 2.148, and 3.841) presented in Section 5.1. Training time is represented in log-scale to enhance readability.

Figure 3(a) plots the model generation time when varying the number of attributes from 10 to 100 on datasets with 50,000 data rows. Item pruning reduces the model generation time between

[2]http://www.almaden.ibm.com/software/quest/Resources

10 and 100 times with respect to the standard classifier. Higher pruning thresholds, albeit slightly reducing accuracy, yield a better scalability with respect to the number of attributes.

Figure 3(b) shows the model generation time when varying the cardinality from 100,000 to 1,000,000 rows on datasets with nine attributes. In this case, the number of new items in the dataset grows significantly more slowly than the number of rows. Hence, item pruning becomes slowly less effective in reducing the execution time.

6 Conclusions

I-prune is an item pruning approach that discards uninteresting items for associative classification purposes. Interestingness is evaluated by means of several quality measures, among which the Chi-square has been experimentally shown to be the most effective.

Experiments on real and synthetic datasets highlight the effectiveness of I-prune in significantly reducing model creation time, while increasing or preserving the accuracy of the generated model. The experimental evaluation showed that tuning the Chi-square threshold yields higher accuracy with respect to simply tuning the support threshold. Hence, correlation based item filtering also yields more accurate classifiers than frequency (i.e., support) based filtering.

To evaluate the effectiveness of our approach, we integrated I-prune in an FP-growth based associative classifier. However, the proposed item filtering approach is general and can be easily and efficiently integrated in any associative classifier.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *20th International Conference on Very Large Data Bases (VLDB'94)*, pages 487–499, September 1994.

[2] M.-L. Antonie, O. Zaiane, and R. Holte. Learning to use a learned model: A two-stage approach to classification. In *IEEE International Conference on Data Mining (ICDM'06)*, pages 33–42, Dec. 2006.

[3] E. Baralis, S. Chiusano, and P. Garza. A lazy approach to associative classification. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):156–171, 2008.

[4] E. Baralis and P. Garza. A lazy approach to pruning classification rules. In *IEEE International Conference on Data Mining (ICDM'02)*, pages 35–42, December 2002.

[5] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.

[6] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing associations rules to correlations. In *ACM SIGMOD International Conference on Management of Data (SIGMOD'97)*, pages 265–276, May 1997.

[7] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *IEEE International Conference on Data Engineering (ICDE'07)*, pages 716–725, April 2007.

[8] H. Cheng, X. Yan, J. Han, and P. Yu. Direct discriminative pattern mining for effective classification. In *IEEE International Conference on Data Engineering (ICDE'08)*, pages 169–178, April 2008.

[9] F. Coenen and P. Leng. The effect of threshold values on association rule based classification accuracy. *Data & Knowledge Engineering*, 60(2):345–360, 2007.

[10] F. Coenen, P. Leng, and L. Zung. Considering re-occurring features in associative classifiers. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05)*, volume 3518 of *Lecture Notes in Computer Science*, pages 216–225. Springer, 2005.

[11] W. W. Cohen. Fast effective rule induction. In *International Conference on Machine Learning (ICML'95)*, pages 115–123, July 1995.

[12] G. Dong, X. Zhang, L. Wong, and J. Li. CAEP: Classification by aggregating emerging patterns. In *International Conference on Discovery Science*, pages 30–42, December 1999.

[13] W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. Yu, and O. Verscheure. Direct mining of discriminative and essential frequent patterns via model-based search tree. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*, pages 230–238, New York, NY, USA, 2008. ACM.

[14] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[15] M. A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1437–1447, 2003.

[16] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *ACM SIGMOD International Conference on Management of Data (SIGMOD'00)*, pages 1–12, May 2000.

[17] J. Li, G. Dong, and K. Ramamohanarao. Making use of the most expressive jumping emerging patterns for classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'02)*, pages 220–232, April 2000.

[18] J. Li, G. Dong, K. Ramamohanarao, and L. Wong. DeEPs: A new instance-based lazy discovery and classification system. *Machine Learning*, 54(2):99–124, 2004.

[19] W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In *IEEE International Conference on Data Mining (ICDM'01)*, pages 369–376, November 2001.

[20] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *International Conference on Knowledge Discovery and Data Mining (KDD'98)*, pages 80–86, August 1998.

[21] J. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, 1993.

[22] R. Rak, W. Stach, O. R. Zaïane, and M.-L. Antonie. Considering re-occurring features in associative classifiers. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05)*, volume 3518 of *Lecture Notes in Computer Science*, pages 240–248. Springer, 2005.

[23] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, pages 32–41, July 2002.

[24] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[25] F. Thabtah, P. Cowling, and Y. Peng. Mmac: a new multi-class, multi-label associative classification approach. In *IEEE International Conference on Data Mining (ICDM'04)*, pages 217–224, Nov. 2004.

[26] F. Thabtah, P. Cowling, and Y. Peng. Mcar: multi-class classification based on association rule. In *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'05)*, pages 33–, 2005.

[27] A. Veloso, W. Meira, and M. Zaki. Lazy associative classification. In *IEEE International Conference on Data Mining (ICDM'06)*, pages 645–654, Dec. 2006.

[28] A. Veloso, W. Meira, Jr., M. Cristo, M. Gonçalves, and M. Zaki. Multi-evidence, multi-criteria, lazy associative document classification. In *International Conference on Information and Knowledge Management (CIKM'06)*, pages 218–227, 2006.

[29] J. Wang and G. Karypis. Harmony: Efficiently mining the best rules for classification. In *SIAM International Conference on Data Mining (SDM'05)*, pages 205–216, 2005.

[30] J. Wang and G. Karypis. On mining instance-centric classification rules. *IEEE Transactions on Knowledge and Data Engineering*, 18(11):1497–1511, 2006.

[31] Y. J. Wang, Q. Xin, and F. Coenen. Mining efficiently significant classification association rules. In *Data Mining: Foundations and Practice*, volume 118 of *Studies in Computational Intelligence*, pages 443–467. Springer, 2008.

[32] O. R. Zaïane and M.-L. Antonie. On pruning and tuning rules for associative classifiers. In *Knowledge-Based Intelligent Information & Engineering Systems (KES'05)*, volume 3683 of *Lecture Notes in Computer Science*, pages 966–973. Springer, 2005.