Application-Independent Feature Construction from Noisy Samples[*]

Dominique Gay[1], Nazha Selmaoui[1], and Jean-François Boulicaut[2]

[1] University of New Caledonia, ERIM EA3791, PPME EA3325,
BP R4, 98851 Nouméa, New Caledonia
{dominique.gay, nazha.selmaoui}@univ-nc.nc
[2] INSA-Lyon, LIRIS CNRS UMR5205
69621 Villeurbanne Cedex, France
jean-francois.boulicaut@insa-lyon.fr

Abstract. When training classifiers, presence of noise can severely harm their performance. In this paper, we focus on "non-class" attribute noise and we consider how a frequent fault-tolerant (FFT) pattern mining task can be used to support noise-tolerant classification. Our method is based on an application independent strategy for feature construction based on the so-called δ-free patterns. Our experiments on noisy training data shows accuracy improvement when using the computed features instead of the original ones.

1 Introduction

Real-world data may contain noise. When dealing with classification tasks, it has been shown that presence of noise in the data may have a negative impact on the performance of classifiers learned from the data [1]. One may differentiate two main types of noise: class noise when noise affects the class label, and attribute noise when noise affects all attributes but the class label. Many solutions have been proposed to tackle class noise, e.g., by noise elimination or noise correction (see [1] for a survey) and more recently by instance weighting [2]. On the other hand, others approaches aim at solving the problem of attribute noise by noise identification and modeling [3,4], and noise cleansing [1,5]. In this paper, we focus on Boolean attribute noise problem and we address the following important question: *how to learn accurate predictive models from attribute-noisy data sets?*

Instead of removing noisy instances or correcting noisy values, we propose a method to cope with attribute noise without changing or removing any attributes values in the training data. Our approach combines two recent advances in fault-tolerant itemset mining and feature construction. The goal of *fault-tolerant itemset* mining [6] is to support the discovery of relevant frequent itemsets in noisy binary data (see, e.g., [7] for a recent survey). Among others, an extension to (frequent) closed set mining towards fault-tolerance has been studied in [8] that enables a bounded number (δ) of errors per item/attribute. It is based on the

[*] This work is partly funded by French ANR contract MDCO-2007 Bingo2.

T. Theeramunkong et al. (Eds.): PAKDD 2009, LNAI 5476, pp. 965–972, 2009.

so-called δ-free sets, i.e., the approximate condensed representation of frequent itemsets introduced in [9]. Using such patterns is the core of our approach to robust *feature construction.* Following the proposals from, for instance, [10,11,12], we consider that attribute sets may be more relevant than single attributes for class discrimination. Then, pattern types based on the so-called closedness properties enable to avoid redundant features in an application-independent setting. In this paper, we investigate further the use of δ-freeness (and thus δ-closedness) when considering feature construction from noisy training data. From that perspective, it extends our previous work [11] which focussed on noise-free samples only. Our proposal can be summarized as follows. First, we mine non-redundant fault-tolerant patterns based on (δ)-free itemsets. Then, we process these patterns to compute new features that will enable to encode a new training set. Finally, classical classification algorithms can be applied. This proposal is yet another contribution to pattern-based classification. In pioneering studies, association rules were considered [13]. Since then, emerging patterns [14] (see [15] for a survey and [16] for a noise tolerant pattern-based approach), and more recently condensed representations of frequent itemsets have been studied [10,11,12]. The paper is organized as follows. Section 2 provides the needed definitions before the description of our proposal in Section 3. Section 4 provides an experimental validation on selected `UCI` (http://archive.ics.uci.edu/ml/) data. Section 5 concludes.

2 Preliminaries

Let us consider a binary database $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$, where $\mathcal{T}$ is a set of transactions (or objects) described by a set $\mathcal{I}$ of Boolean items (or attributes) and $\mathcal{R} : \mathcal{T} \times \mathcal{I} \mapsto \{0,1\}$. When $\mathcal{R}(t,i) = 1$, we say that transaction t contains item i. An itemset $I \in \mathcal{I}$ is a set of items. The frequency of itemset $I \in \mathcal{I}$ is $freq(I,r) = |Objects(I,r)|$ where $Objects(I,r) = \{t \in \mathcal{T} \mid \forall i \in \mathcal{I} \quad \mathcal{R}(t,i) = 1\}$. I is said to be γ-frequent if $freq(I,r) \geq \gamma$.

Definition 1 (association rule, δ-strong rule, δ-free itemset). *An* association rule π *on* r *is an expression* $I \Rightarrow J$, *where* $I \subseteq \mathcal{I}$ *and* $J \subseteq \mathcal{I} \setminus I$. *The* frequency *of the rule* π *is* $freq(I \cup J, r)$ *and its* confidence *is* $conf(\pi, r) = freq(I \cup J, r)/freq(I, r)$. *Let* δ *be an integer. A* δ-strong rule *is an association rule of the form* $I \Rightarrow^{\delta} J$ *which is violated in at most* δ *objects, and where* $I \subseteq \mathcal{I}$ *and* $J \subseteq \mathcal{I} \setminus I$. *An itemset* $I \subseteq \mathcal{I}$ *is a* δ-free itemset *iff there is no* δ*-strong rule which holds between its proper subsets. When* $\delta = 0$, δ *is omitted, and we talk about* strong rules, *and* free itemsets.

First introduced in [9], δ-free itemsets and δ-strong rules have been designed as an approximate condensed representation for frequency queries. δ-freeness is a generalization of the key pattern concept [17] (case $\delta = 0$) and it can also be discussed in terms of equivalence classes.

Definition 2 (δ-closure, equivalence class). *Let* δ *be an integer. The* δ-closure *of an itemset* I *on* r *is* $cl_\delta : \mathcal{P}(\mathcal{I}) \rightarrow \mathcal{P}(\mathcal{I})$ *s.t.* $cl_\delta(I, r) = \{i \in \mathcal{I} \mid$

$freq(I,r) - freq(I \cup \{i\}, r) \leq \delta\}$. When $\delta = 0$, $cl_0(I,r) = \{i \in \mathcal{I} \mid freq(I,r) = freq(I \cup \{i\}, r)\}$ and it corresponds to the well-known closure operator. We can also group itemsets by δ-closure equivalence classes *(δ-CECs): two δ-free itemsets I and J are said δ-equivalent ($I \sim_{cl_\delta} J$) if $cl_\delta(I,r) = cl_\delta(J,r)$.*

Once again, when $\delta = 0$, we get the formalization of closure equivalence classes from [17]. We can also derive δ-strong rules from δ-CECs (i.e., from δ-free itemsets and their δ-closures). Indeed, we have a δ-strong association rule between a δ-free itemset and each element of its δ-closure. In [7,8], δ-free itemsets and their associated δ-closures are combined to define the so-called δ-bi-sets.

Definition 3 (frequent δ-bi-set). *A bi-set (T,I) such that $T \subseteq \mathcal{T}$ and $I \subseteq \mathcal{I}$ is a frequent δ-bi-set iff $I = I_1 \cup I_2$, I_1 is a γ-frequent δ-free itemset, $cl_\delta(I_1,r) = I$ and $Objects(I_1,r) = T$.*

δ-bi-sets appear as an extension of the so-called formal concepts or associated closed itemsets. Indeed, they appear as examples of maximal combinatorial rectangles of 1 values having at most δ 0 per column [8]. We now consider how we use δ-CECs to derive robust features.

3 Deriving Features from δ-closure Equivalence Classes

In order to manage classification tasks, we are interested in relevant δ-strong association rules contained in δ-CECs. Figure 1(a) shows a typical case of an interesting δ-CEC: δ-free itemsets X and Y do not contain a class attribute and their (equal) δ-closure (X, Y, Z, c_i) contains a class attribute c_i. Indeed, we may derive two potentially interesting δ-strong rules : $X \rightarrow c_i$ and $Y \rightarrow c_i$. According to the formalization from [18], $\pi : X \rightarrow c_i$ is a δ-strong characterization rule (δ-SCR) if c_i is a class attribute and body X is minimal. X is minimal if there is no other frequent rule $\pi' : Y \rightarrow c_i$ s.t. $Y \subseteq X$ and $conf(\pi', r) \geq 1 - \frac{\delta}{\gamma}$. Moreover, when $\delta \in [0; \lfloor \gamma/2 \rfloor[$, the set of δ-SCRs does not contain included or equal body conflicts.

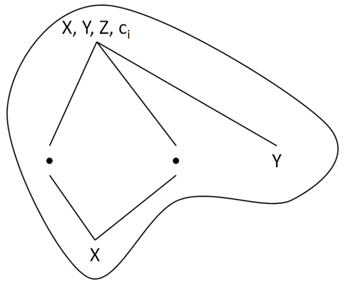
Defining δ-SCR only based on the confidence measure is not sufficient for prediction. Therefore, we propose to exploit a Growth rate measure Gr which has been already proved useful in such a context [14]. The Growth rate of $\pi : X \rightarrow c_i$ is defined as a ratio of relative frequencies as follows:

$$Gr(\pi, r) = \frac{freq_r(X, r_{c_i})}{freq_r(X, r \setminus r_{c_i})}$$

where r_{c_i} is the database restricted to objects of Class c_i. In [19], Gr is set in the general framework of the so-called δ-dependent measures. Such measures depend on the rule antecedent frequency (γ) and the rule number of exceptions (δ) following two principles: *(i)*, when γ is fixed, $Gr(\pi, r)$ increases with $freq(\pi, r)$ and *(ii)*, when δ is fixed, $Gr(\pi, r)$ increases with γ.

This leads us to lower bounds for several interestingness measures (including Growth rate) w.r.t. γ and δ values (see [19] for details). In Figure 1(b), contingency table for $\pi : X \rightarrow c_i$ a δ-strong rule concluding on a class attribute c_i

(a) Interesting case of a δ-closure equivalence class for class c_i.



(b) Contingency table forδ-strong rule $X \rightarrow^{\delta} c_i$.

$X \rightarrow c_i$	c_i	$\bar{c_i}$	Σ
X	$\gamma - \delta$	δ	γ
$\bar{X}$	$\cdot$	$\cdot$	$\cdot$
Σ	$\lvert r_{c_i}\rvert$	$\lvert r \setminus r_{c_i}\rvert$	$\lvert r\rvert$

Fig. 1. Information contained in δ-closure equivalence classes

shows that, by construction, we have a lower bound $(\gamma - \delta)$ for $freq(X, r_{c_i})$, an upper bound δ for $freq(X, r \setminus r_{c_i})$ and other deductible bounds for dotted cells. Moreover, we can deduce a lower bound for Gr and $conf$ measure. Indeed,

$$Gr(\pi, r) \geq \frac{\gamma - \delta}{\delta} \cdot \frac{|r \setminus r_{c_i}|}{|r_{c_i}|} \quad \text{and} \quad conf(\pi, r) \geq 1 - \delta/\gamma$$

Then, with few deduction steps, we get:

$$Gr(\pi, r) \geq 1 \quad \Longrightarrow \quad \delta < \gamma \cdot \frac{|r \setminus r_{c_j}|}{|r|} \tag{1}$$

$$conf(\pi, r) \geq 1/2 \quad \Longrightarrow \quad \delta < \gamma/2 \tag{2}$$

where c_j is the majority class. Thus, for a given frequency threshold γ, γ-frequent δ-free itemsets whose δ-closures contain a class attribute s.t. δ satisfies Equations (1) and (2) are emerging patterns. In the rest of the paper, γ and δ values are constrained w.r.t. Equations (1) and (2).

Our feature construction process may now be summarized with Algorithm 1. Procedure `FeaturesExtraction` (Line 1) mines all γ-frequent δ-free itemsets I' that are bodies of δ-strong characterization rules in r. This step is performed efficiently using straightforward extension of `AC-like`[1]. `AC-like` is an implementation of a levelwise algorithm presented in [9] that benefits from anti-monotonic properties of δ-freeness and γ-frequency to compute γ-frequent δ-free itemsets. Moreover, since we are interested in minimal itemsets whose δ-closures contain a class attribute, there is no need to check supersets of itemsets selected in a precedent level.Then, each I' becomes a new descriptor for r', and (Line 1) the value of I' for a transaction t is the proportion of items in I' that are verified by t in r. *Items* is the dual operator for *Objects*. Now, $\mathcal{R}' \mapsto [0;1]$ and

[1] http://liris.cnrs.fr/~turing

Algorithm 1: Building new data set with noise-tolerant features

```
input  : A binary database $r = \{\mathcal{T}, \mathcal{I}, \mathcal{R}\}$,
         two integers $\gamma$ and $\delta$ for frequency and errors thresholds
output: A numeric database $r' = \{\mathcal{T}, \mathcal{I}', \mathcal{R}'\}$ made of noise-tolerant features as descriptors
begin
    $\mathcal{I}' \longleftarrow$ FeaturesExtraction($r, \gamma, \delta$);
    for $t \in \mathcal{T}$ do
        for $I' \in \mathcal{I}'$ do
            $\mathcal{R}'(t, I') \longleftarrow \frac{|I' \cap \text{Items}(t,r)|}{|I'|}$;
    $r' \longleftarrow \{\mathcal{T}, \mathcal{I}', \mathcal{R}'\}$;
end
```

$\mathcal{R}'(t, I') \in \{0, \frac{1}{p}, \ldots, \frac{p-1}{p}, 1\}$ with $p = |I'|$). Finally, r' is the new database made of noise-tolerant features, ready for a classification learning step.

4 Experimental Validation

To assess the noise-tolerance of our process, we ran our feature construction process (NTFC) on several UCI data sets and several noisy versions of them. We want to learn accurate predictive models despite of noisy samples. Therefore, in our experiments we deal with attribute-noisy training sets and clean test sets. We add random noise at different levels only on attributes and only in training sets. For a data set d and a $x\%$ noise level ($x \in \{10, 20, 30, 40, 50\}$), each attribute value has $x\%$ chances to be changed (within its range values) in each transaction of the training set[2]. When dealing with continuous attributes, we first discretize the training data: each attribute is split into several intervals using entropy-based splitting method [20]. Then, we add $x\%$ noise and perform a simple binarization. Finally, we run C4.5 decision tree [21] and Naive Bayes classifier (NB) on each noisy version of the data set and on each data set enhanced by new descriptors generated with Algorithm 1. All pre-processing steps (adding noise, discretization, binarization) and accuracy results for NB and C4.5 classifiers are obtained with 10-folds stratified cross-validation – using Weka platform [22].

A strategy for setting δ: Automatically setting frequency threshold is still an open question (see [23,24] for preliminary results in that direction). Given a frequency threshold γ, how can we determine relevant δ values? Evolution of δ-dependent interestingness measure (such as growth rate) w.r.t. δ is well known. Decreasing δ implies higher values for Gr but such interesting patterns could be rare, especially in noisy data sets. When increasing δ, extracted patterns tend to match with noisy patterns in training data, but higher δ values tend to be less relevant (with low Gr values). In Figure 2, we plotted training accuracy results w.r.t. δ values for tic-tac-toe data set. As expected, accuracy on training data increases with δ until stabilization (or slowing down). δ values around stabilization are interesting since lower values lead to less accurate models and

[2] It does not mean that $x\%$ of values have been modified.

Table 1. Accuracy results : original v.s. `NTFC` processed data

Dataset	`C4.5`	`NTFC & C4.5` (Avg)	Max	`NB`	`NTFC & NB` (Avg)	Max
breast-w (f:20-40/5)	95.57	**95.62**	**95.85**	97.28	*96.59*	*96.85*
10%	95.14	*94.59*	**95.14**	96.99	**97.22**	**97.56**
20%	93.28	*92.31*	*92.99*	96.99	**97.05**	**97.14**
30%	89.56	**91.00**	**91.85**	96.85	**97.14**	**97.28**
40%	87.70	**87.73**	**88.41**	96.42	*94.99*	**97.28**
50%	87.56	*86.30*	**87.56**	96.28	**96.51**	**97.21**
colic (f:25-50/5)	85.04	*84.31*	**85.85**	79.90	**82.73**	**85.31**
10%	83.14	*82.95*	**85.29**	78.81	**82.59**	**84.77**
20%	82.04	*80.88*	**83.12**	78.81	**81.28**	**85.31**
30%	80.95	*80.78*	**82.85**	79.09	**81.37**	**83.96**
40%	80.93	**81.72**	**84.21**	78.00	**82.32**	**84.22**
50%	81.53	**82.04**	**83.66**	73.92	**80.56**	**82.31**
heart-c (f:25-45/5)	80.47	**82.48**	**84.13**	81.79	**83.20**	**84.12**
10%	78.83	**80.17**	**81.18**	82.78	**83.39**	**84.78**
20%	75.85	**79.46**	**82.18**	83.12	**83.26**	**84.46**
30%	77.19	**78.40**	**80.21**	84.45	*83.93*	**84.82**
40%	77.50	*76.40*	**78.52**	84.45	*82.63*	**84.48**
50%	72.28	**75.53**	**77.52**	79.52	**79.93**	**88.20**
heart-h (f:10-22/2)	75.55	**79.60**	**81.64**	84.07	*82.50*	*83.03*
10%	77.62	**78.83**	**80.56**	83.73	*80.70*	*81.31*
20%	76.60	**78.24**	**80.29**	83.39	*82.30*	*83.00*
30%	75.56	**79.08**	**80.00**	84.05	*83.10*	**84.34**
40%	74.17	**79.19**	**80.96**	83.37	*80.51*	*82*
50%	73.83	**79.93**	**81.68**	79.95	**80.17**	**82.34**
heart-s (f:22-32/2)	81.85	**83.33**	**86.67**	81.48	**82.10**	**84.45**
10%	80.74	**81.17**	**84.45**	81.48	*81.17*	**85.92**
20%	78.52	**79.75**	**80.74**	82.59	*81.60*	**84.07**
30%	74.07	**74.07**	**76.30**	81.11	**81.92**	**84.07**
40%	72.96	*70.15*	**73.33**	81.85	*71.70*	*72.59*
50%	59.26	**64.07**	**65.55**	55.55	**61.18**	**65.18**
iris (f:7-30)	93.33	**93.33**	**93.33**	92.67	**94.00**	**94.00**
10%	92.67	**93.00**	**94.67**	92.67	**92.67**	**94.00**
20%	90.00	*89.22*	**93.33**	92.67	**93.11**	**94.67**
30%	90.00	*86.50*	*89.33*	92.67	*91.33*	*92.00*
40%	86.00	*84.67*	*84.67*	92.67	*88.00*	*88.33*
50%	84.00	*77.33*	*77.33*	94.00	*77.33*	*77.33*
tic-tac-toe (f:25-40/5)	93.21	**99.40**	**100.00**	68.47	**79.72**	**81.73**
10%	92.06	**97.65**	**99.17**	68.57	**75.99**	**78.81**
20%	88.95	**96.26**	**97.18**	70.14	**73.81**	**74.42**
30%	82.78	**93.00**	**95.62**	71.92	**73.56**	**74.53**
40%	77.98	**89.43**	**91.55**	74.11	*73.56*	**74.83**
50%	71.39	**83.82**	**85.60**	70.57	**71.16**	**71.39**
wine (f:9-18)	91.08	**91.53**	**93.76**	98.89	*92.92*	*93.86*
10%	91.63	**91.87**	**94.38**	96.6	*93.24*	*94.41*
20%	91.01	**91.74**	**92.71**	96.04	*94.42*	*95*
30%	84.37	**88.15**	**91.04**	94.38	*91.73*	**94.97**
40%	82.02	**88.28**	**88.82**	93.27	*91.42*	**93.79**
50%	82.12	**85.11**	**87.58**	88.10	**89.29**	**90.42**

higher values bring nothing more. Let δ_{opt} denote these values. These interesting values depends on the amount of noise in data. Since in real case, noise level is not known a priori, a reasonable way to reach these δ_{opt} values is (1) to increase δ starting from 0, and (2) to check evolution of training data accuracy for stabilization.

Accuracy results: Accuracy results comparison are shown in Tab. 1. For each data set, we used different frequency thresholds indicated in parenthesis: (f:x-y/z) means that absolute frequency thresholds vary from x to y with step z (when regular). `NTFC` accuracy results are reported in two columns: *Avg* for average accuracy overall tested γ values for δ_{opt} values and *Max* for maximal accuracy overall γ and δ values.

First, we remark that there often exists a combination of γ and δ values (col. Max) for which `NTFC & C4.5` (44/48) and `NTFC & NB` (36/48) achieve better accuracy results than on original data. Second, even if less impressive, our simple

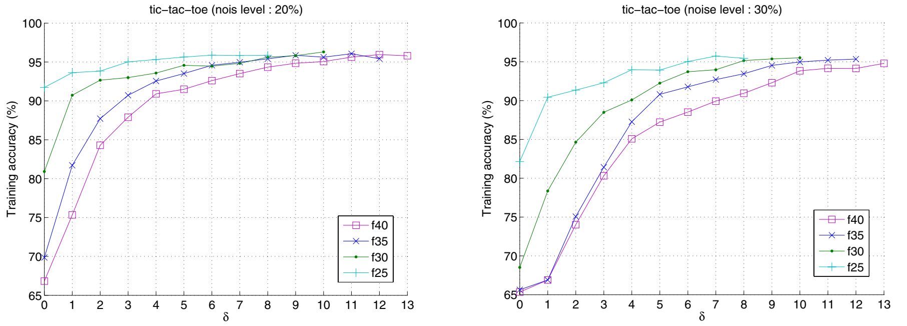


Fig. 2. Evolution of `NTFC & C4.5` accuracy on training database w.r.t. δ for various frequency thresholds and noise levels on `tic-tac-toe` data set

strategy for setting δ (col. Avg) also allows us to catch better accuracy results on noisy data sets enhanced by `NTFC` (36/48 for `C4.5` and 27/48 for `NB`).

5 Conclusion

In this paper, we have proposed an original process (`NTFC`) combining recent advances in fault-tolerant pattern mining and feature construction to handle classification tasks in attribute-noisy environments. We enhance data description with new features based on relevant fault-tolerant patterns, say γ-frequent δ-free itemsets and their δ-closures. Experimental results show that classification tools such that `NB` and `C4.5` achieve higher accuracy results with an enhanced description of data (i.e., using noise-tolerant features) than with original data. δ values have shown to be related to the amount of noise in the data. Finally, we have given a strategy to automatically set δ values that are relevant w.r.t. the noise level in data samples – thus supporting a parameter tuning dedicated to our process.

References

1. Zhu, X., Wu, X.: Class noise vs. attribute noise : A quantitative study. Artificial Intelligence Revue 22, 177–210 (2004)
2. Rebbapragada, U., Brodley, C.E.: Class noise mitigation through instance weighting. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS, vol. 4701, pp. 708–715. Springer, Heidelberg (2007)
3. Kubica, J., Moore, A.W.: Probabilistic noise identification and data cleaning. In: Proceedings ICDM 2003, pp. 131–138. IEEE Computer Society, Los Alamitos (2003)
4. Zhang, Y., Wu, X.: Noise modeling with associative corruption rules. In: Proceedings ICDM 2007, pp. 733–738. IEEE Computer Society, Los Alamitos (2007)

5. Yang, Y., Wu, X., Zhu, X.: Dealing with predictive-but-unpredictable attributes in noisy data sources. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS, vol. 3202, pp. 471–483. Springer, Heidelberg (2004)
6. Yang, C., Fayyad, U.M., Bradley, P.S.: Efficient discovery of error-tolerant frequent itemsets in high dimensions. In: Proceedings KDD 2001, pp. 194–203. ACM Press, New York (2001)
7. Besson, J., Pensa, R.G., Robardet, C., Boulicaut, J.F.: Constraint-based mining of fault-tolerant patterns from boolean data. In: Bonchi, F., Boulicaut, J.-F. (eds.) KDID 2005. LNCS, vol. 3933, pp. 55–71. Springer, Heidelberg (2006)
8. Pensa, R.G., Robardet, C., Boulicaut, J.F.: Supporting bi-cluster interpretation in 0/1 data by means of local patterns. Intelligent Data Analysis 10, 457–472 (2006)
9. Boulicaut, J.F., Bykowski, A., Rigotti, C.: Approximation of frequency queries by means of free-sets. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS, vol. 1910, pp. 75–85. Springer, Heidelberg (2000)
10. Selmaoui, N., Leschi, C., Gay, D., Boulicaut, J.F.: Feature construction and delta-free sets in 0/1 samples. In: Todorovski, L., Lavrač, N., Jantke, K.P. (eds.) DS 2006. LNCS, vol. 4265, pp. 363–367. Springer, Heidelberg (2006)
11. Gay, D., Selmaoui, N., Boulicaut, J.F.: Feature construction based on closedness properties is not that simple. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS, vol. 5012, pp. 112–123. Springer, Heidelberg (2008)
12. Cheng, H., Yan, X., Han, J., Hsu, C.W.: Discriminative frequent pattern analysis for effective classification. In: Proceedings ICDE 2007, pp. 716–725. IEEE Computer Society, Los Alamitos (2007)
13. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings KDD 1998, pp. 80–86. AAAI Press, Menlo Park (1998)
14. Dong, G., Li, J.: Efficient mining of emerging patterns: discovering trends and differences. In: Proceedings KDD 1999, pp. 43–52. ACM Press, New York (1999)
15. Ramamohanarao, K., Fan, H.: Patterns based classifiers. World Wide Web 10, 71–83 (2007)
16. Fan, H., Ramamohanarao, K.: Noise tolerant classification by chi emerging patterns. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS, vol. 3056, pp. 201–206. Springer, Heidelberg (2004)
17. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining frequent patterns with counting inference. SIGKDD Explorations 2, 66–75 (2000)
18. Crémilleux, B., Boulicaut, J.F.: Simplest rules characterizing classes generated by delta-free sets. In: Proceedings ES 2002, pp. 33–46. Springer, Heidelberg (2002)
19. Hébert, C., Crémilleux, B.: Optimized rule mining through a unified framework for interestingness measures. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2006. LNCS, vol. 4081, pp. 238–247. Springer, Heidelberg (2006)
20. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continous-valued attributes for classification learning. In: Proceedings IJCAI 1993, pp. 1022–1027. Morgan Kaufmann, San Francisco (1993)
21. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann, San Francisco (1993)
22. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
23. Cerf, L., Gay, D., Selmaoui, N., Boulicaut, J.F.: A parameter free associative classifier. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 293–304. Springer, Heidelberg (2008)
24. Zhang, S., Wu, X., Zhang, C., Lu, J.: Computing the minimum-support for mining frequent patterns. Knowledge and Information Systems 15, 233–257 (2008)