Feature Projection Based Rule Classification*

Tolga Aydın and H.Altay Güvenir

Bilkent University
Department of Computer Engineering

Bilkent, 06533 Ankara, TURKEY
{atolga, guvenir}@cs.bilkent.edu.tr

Abstract. *Due to the increase in data mining research and applications, selection of interesting rules among a huge number of learned rules is an important task in data mining applications. In this paper, the metrics for the interestingness of a rule is investigated and an algorithm that can classify the learned rules according to their interestingness is developed. Classification algorithms were designed to maximize the number of correctly classified instances, given a set of unseen test cases. Furthermore, feature projection based classification algorithms were tested and shown to be successful in large number of real domains. So, in this work, a feature projection based classification algorithm (VFI, Voting Feature Intervals) is adapted to the rule interestingness problem, and FPRC (Feature Projection Based Rule Classification) algorithm is developed.*
Keywords: rule classification, interestingness, voting, feature projection

1. Introduction

Data mining is the efficient discovery of patterns, as opposed to data itself, in large databases [1]. Patterns in the data can be represented in many different forms, including classification rules, association rules, clusters, sequential patterns, time series, contingency tables, and others [2]. In this paper, we concentrate on the patterns represented by the classification rules. Quinlan's C4.5/C5.0 [3] is an industrial quality decision tree and classification rule induction technique that we used in our research.

A pattern is interesting if it is easily understood, unexpected, potentially useful and actionable, novel, or it validates some hypothesis that a user seeks to confirm [4]. Therefore, selection of interesting patterns seems to be an important research topic.

In this paper, we deal with classification rules that were obtained using C4.5 decision tree/rule induction algorithm on Gastric Carcinoma [5] data set. Gastric Carcinoma data set includes information about 285 patients having stomach cancer illness. These classification rules are classified as either *interesting* or *uninteresting* by the FPRC (Feature Projection Based Rule Classification) algorithm.

The classification rules belonging to the Gastric Carcinoma data set is described in the next section. The knowledge representation and the FPRC algorithm is explained in sections 3 and 4, respectively. Section 5 describes the experimental results of the FPRC algorithm. Finally, the last section concludes with some remarks and suggestions for future work.

2. Rules belonging to the Gastric Carcinoma Data Set

We used C4.5 decision tree/rule induction algorithm on Gastric Carcinoma data set and obtained 33 classification rules. However, it is also possible to use other

* This project is supported by TUBITAK (Scientific and Technical Research Council of Turkey) under Grant 101E044.

classification algorithms to obtain different and more classification rules. The only criteria for such a classification algorithm is to represent the knowledge in the form of classification rules so that it will be suitable to compare those rules with the ones obtained using C4.5.

The aim of the work presented in this paper is to select the interesting rules from the set of learned classification rules. This selection problem is modeled as a new classification problem and a *rule set* is produced from the given rules. There is a total of 33 instances in this rule set, where each rule is represented by an instance. Also each instance is represented by a vector whose components are the metric values having the potential to determine the interestingness of the corresponding instance and the interestingness class label of the instance. The interestingness class concept and the metrics used in the representation of the instances are explained in detail in the following section.

3. Knowledge Representation

FPRC algorithm makes use of VFI (Voting Feature Intervals) classification algorithm [6], which was previously developed by the Bilkent University Machine Learning Group. However, the VFI in FPRC does not use the data set (such as Gastric Carcinoma) as the original VFI algorithm does. Instead, it uses the instances (rule set) that were produced from the classification rules of the C4.5 rule induction algorithm. FPRC classifies the classification rules of the rule set according to their interestingness. Therefore, it performs a second level of data mining and makes learning of learning.

Each classification rule is in the form of $A_1, A_2, \ldots, A_n \rightarrow Class = c$ and produces just one rule instance. Here, A_i's and c denote the features and the class label of the data set, respectively. This rule instance is shown by a vector consisting of linear or nominal features and a class label, just as every instance of the data set. However, as opposed to the data set, features of a rule instance consist of rule metrics that were developed throughout our research studies. Each feature carries information about a specific metric of the corresponding rule. On the other hand, the class label of the rule instance denotes the interestingness of the associated rule. To distinguish from the *class* label of the instances of the data set, *class* label of the instances of the rule set is referred to as *interestingness class* label.

Interestingness Classes:

There are two possible interestingness class labels for each rule: “interesting” and “uninteresting”. While experimenting the FPRC algorithm with the rules of the Gastric Carcinoma data set, each rule was classified by the expert of the domain as either interesting or uninteresting.

Features:

Each of the rule metrics that were developed in the course of the research studies corresponds to a feature in the rule set. There are 7 metrics (features) used so far. Table 1 shows the names and the possible values taken by these features. Among these features, six take linear and just one takes nominal values. The rule set does not include any missing feature value.

Any rule learned by the C4.5 algorithm belongs to just one class. And this class value constitutes the value of the first feature of that rule. For instance, if a rule learned

from the Gastric Carcinoma data set has the class EarlyI, the value of the first feature of the instance associated to that rule becomes EarlyI.

Table 1. Features of the rule set.

Feature	Possible Values
Class	Early I, Early IIa, Early IIb, Early II c, Early III, BI, BII, BIII, BIV
ClassFrequency	[0, 100]
RuleSize	> 0
Confidence	[0, 100]
CoverageRatio	[0, 100]
Support	[0, 100]
Completeness	(0, 100]

Below one of the rules produced by C4.5 algorithm for the Gastric Carcinoma data set is given:

Rule 9:
If (sigarettes_per_day > 30) **and**
(deep_ulc = 1) **and**
(infiltrated_ulcer = 0) **and**
(base_infiltrated = 0)
then Class = EarlyIII

This rule was labeled as interesting by the domain expert. And the associated instance of the rule is constructed as follows:

Class = EarlyIII
ClassFrequency = 2.1%
RuleSize: 4
Confidence = 100%
CoverageRatio = 1%
Support = 1%
Completeness = 50%

Among the features shown above, *ClassFrequency* gives the percentage of the instances of the data set that have EarlyIII, the value of the first feature of Rule 9, as the class label. *RuleSize* feature shows the number of conditions in the antecedent part of the rule. For a rule shown as $A \rightarrow B$:

n : total number of the instances of the data set
$|A|$: number of instances satisfied by the rule antecedent in the data set
$|B|$: number of instances having B as the class label in the data set
$|A\&B|$: number of instances satisfied both by the rule antecedent and consequent in the data set

According to these information, other feature values of a rule instance are determined as follows:

Confidence = |*A*&*B*| / |*A*|
CoverageRatio= |*A*| / *n*
Support = |*A*&*B*| / *n*
Completeness = |*A*&*B*| / |*B*|

4. FPRC Algorithm

FPRC (Feature Projection Based Rule Classification) is the first rule interestingness classification algorithm developed by the Bilkent University Machine Learning Group. FPRC classifies the rules according to their interestingness. The rules need not necessarily be produced by the C4.5 algorithm. Any other algorithm capable of producing classification rules can be used to prepare the input rules of the FPRC algorithm. In the following subsections, feature projection based concept description representation technique and the execution of FPRC is explained. Finally, in the experimental results section, FPRC's test results using the Gastric Carcinoma data set are given.

VFI used in the course of FPRC shows the learned concept description as a set of feature intervals. An interval can either be a range or point interval. A range interval corresponds to a set of continuous values, where as a point interval corresponds to just one value for the feature on which they exist. In a range interval, beginning and end values of the interval are kept along with the vote vector that this interval distributes among the interestingness classes "interesting" and "uninteresting". For a point interval, the scenario is the same except that the beginning and the end values of the interval are the same.

4.1 Learning Concept Description by the VFI Algorithm

Learning concept description by the VFI algorithm is shown in Figure 1. The procedure *find_end_points*(*TrainingSet*, *f*, *c*) finds the lowest and the highest values for linear feature (metric feature) *f* from the instances of interestingness class *c* and also finds each observed value for nominal feature *f* from the instances in *TrainingSet*. For each linear feature at most four end points are found, since there are two interestingness classes. Then the list of end points is sorted and each consecutive pair of points constitutes a range interval. For nominal features, each observed value found constitutes a point interval of a single value.

Following this, *count_instances*(*f*, *i*, *c*) procedure is used to find the number of training instances falling into the i^{th} interval of the feature *f* and having *c* as the interestingness class label. When a training instance of interestingness class *c* falls on the boundary of two consecutive intervals of linear feature f, *interval_interestingness_class_count* of both intervals are incremented by 0.5. Otherwise, *interval_interestingness_class_count* of the associated interval is incremented by 1. For a nominal feature, the increment amount will always be 1 since no range interval exists for nominal type features.

It is better to explain the concept description by an example. In Figure 2, there is a rule set including 7 training instances and 2 linear features. Four of the rules have been found as interesting where as the remaining rules have been labeled as uninteresting. Each of the feature projection contains 4 range intervals. For instance, when we consider the feature projection with respect to the 2nd feature, 3 training instances fall on the boundary of the 1st and the 2nd interval. Two of these training instances have previously been labeled as interesting and one as uninteresting. Since there are no other training instances that fall on the 1st range interval:

interval_interestingness_class_count[2, *1*, *interesting*] = 0.5 + 0.5 = 1,
interval_interestingness_class_count[2, *1*, *uninteresting*] = 0.5.

VFI$_{train}$(***TrainingSet***):
begin
  for each metric feature *f*
    /* there are 2 interestingness classes, 1: interesting, 0: uninteresting */
    for each interestingness class *c*
      EndPoints[*f*] = *EndPoints*[*f*] ∪ find_end_points(*TrainingSet*, *f*, *c*)
    sort (*EndPoints*[*f*])
    /* each pair of consecutive points in *EndPoints*[*f*] form a metric feature interval */
    for each interval *i* /* on metric feature *f* */
      for each interestingness class *c*
        /* count the number of instances of interestingness class *c* falling into interval *i* */
        interval_interestingness_class_count[*f*, *i*, *c*] = count_instances(*f*, *i*, *c*)
  return (Intervals)
end.

Figure1. Learning concept description by the VFI algorithm.

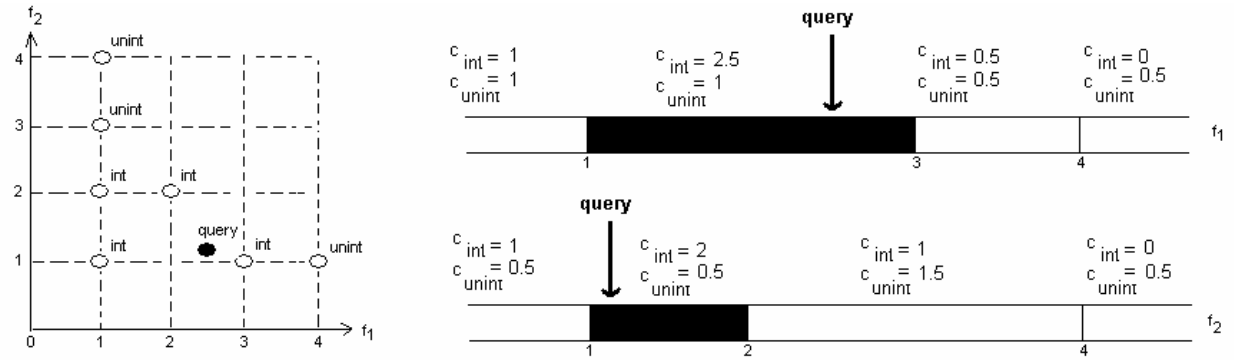

Figure 2. Example rule set and the concept description learned by the VFI algorithm.

4.2 Classification in the VFI Algorithm

The classification phase of the VFI algorithm is given in Figure 3. The process starts by initializing the votes of each interestingness class to zero. For each feature *f*, the interval on feature dimension *f* into which *query*$_f$ falls is searched, where *query*$_f$ is the *f* value of the query rule. If *f* is a nominal feature and *query*$_f$ was not observed in the learned concept description before, this feature will not participate in determining the interestingness class of the query rule. If such a situation does not exist in a nominal feature *f* or if *f* is a linear feature, then the interval *i* into which *query*$_f$ falls is found. For each interestingness class *c*, feature *f* gives a vote equal to

$$feature\_vote[f, c] = \frac{interval\_interestingness\_class\_count[f, i, c]}{interestingness\_class\_count[c]}$$

In the above formula, *interval_interestingness_class_count* is divided by the number of training rule instances having interestingness class *c*. The reason is to increase the value of the vote given to the minor interestingness class. If a query rule falls on the boundary of two intervals on a feature projection f, the votes given by the two neighbor intervals for the query rule are averaged.

Following this, the individual vote of feature *f* for interestingness class *c*, *feature_vote*[*f*, *c*], is normalized to have the sum of votes of feature *f* equal to 1. This ensures that all features have the same voting power. Finally, each feature declares local vote distribution among the two interestingness classes. These local votes are summed up to get a total vote vector and to predict the interestingness class of the query rule as the one having the highest total vote. Classification phase can better be explained by the example given in Figure 2. The query rule falls on the 2nd range interval in both of the feature projections.

feature_vote[*1*, *interesting*] = 2.5 / 4 = 0.625
feature_vote[*1*, *uninteresting*] = 1 / 3 = 0.333
normalized_feature_vote[*1*, *interesting*] = 0.625 / (0.625 + 0.333) = 0.65
normalized_feature_vote[*1*, *uninteresting*] = 0.333 / (0.625 + 0.333) = 0.35

feature_vote[2, interesting] = 2 / 4 = 0.5
feature_vote[*2*, *uninteresting*] = 0.5 / 3 = 0.166
normalized_feature_vote[*2*, *interesting*] = 0.5 / (0.5 + 0.166) = 0.75
normalized_feature_vote[*2*, *uninteresting*] = 0.166 / (0.5+ 0.166) = 0.25

Therefore, total votes given two both interestingness classes are as follows:

vote[*interesting*] = 0.65 + 0.75 = 1.4
vote[*uninteresting*] = 0.35 + 0.25 = 0.6

Query rule is labeled as interesting since vote[*interesting*] (1.4) is greater than the vote[*uninteresting*] (0.6). Certainty factor (C_f) of the classification is:

1.4 / (1.4 + 0.6) = 70%.

VFI$_{query}$(*Intervals*, *e*): /* *e*: query rule to be classified */
begin
  for each interestingness class *c*
    vote[*c*] = 0
  for each metric feature *f*
    for each interestingness class *c*
      feature_vote[*f*, *c*] = 0 /* vote of metric feature *f* for interestingness class *c* */
    If *f* is a nominal feature and e_f value is not observed in the training data
      do nothing
    else
      i = *find_interval*(*f*, e_f)
      for each interestingness class *c*

$$feature\_vote[f,c] = \frac{interval\_i\ nteresting\ ness\_class\ \_count\ [f,i,c]}{interestin\ gness\_clas\ s\_count\ [c]}$$

      normalize_metric_feature_votes(*f*) /* such that $\sum_c feature\_vo\ te[f,c] = 1$ */

  for each interestingness class *c*
    vote[*c*] = *vote*[*c*] + *feature_vote*[*f*, *c*]

  return interestingness class *c* with highest *vote*[*c*]
end.

Figure 3. Classification by the VFI algorithm.

4.3 Execution of the FPRC Algorithm

Figure 4 shows the execution of the FPRC algorithm. The set of classification rules produced by the C4.5 algorithm is denoted as R. In the beginning of the program, a random subset of these rules is taken as warm-up rules ($R_{warm\text{-}up}$) and they are classified by the domain expert to learn the initial concept description.

The number of warm-up rules is determined by the two parameters. Warm-up rule count should not be less than a predefined threshold (*Warm_up_Rule_Count*) and an excessive imbalance between the number of interesting warm-up rules and uninteresting warm-up rules should be prevented (*Min_Occurrence_Rate*). For instance, for a *Min_Occurrence_Rate* of 0.25, the ratio of the warm-up rules classified as interesting (or uninteresting) to the whole warm-up rules should not be less than 25%. These two parameters are required to ensure that the initial concept description is sufficiently powerful to make reliable predictions. These two parameters are not used after the warm-up period.

In the following stage of the FPRC algorithm, the rules in the set $R_q = R - R_{warm\text{-}up}$ are classified by the concept description learned by the VFI algorithm. If the certainty factor (C_f) of the classification is greater than or equal to the minimum certainty factor ($MinC_t$), VFI is found successful and the query rule is placed into the set of rules that were successfuly classified by the VFI algorithm, (R_s). The certainty factor of a query rule that was predicted to be an (interesting)uninteresting rule is the ratio of the vote given to the (interesting)uninteresting class to the sum of the votes given to both interestingness classes. If the certainty factor (C_f) of the classification is smaller than the minimum certainty factor ($MinC_t$), the concept description constructed by the VFI is said to be not powerful enough to make a reliable prediction for the query rule. Therefore, this query rule, just like the warm-up rules, is presented to the expert for classification. The certainty factor of any classification performed by the domain expert is 100%. This rule's metric values and interestingness class label are then packaged into a vector and a corresponding rule instance is constructed. The instance is placed into the training rule set (R_t), and the concept description is reconstructed to have a more powerful prediction model. The training rule set is $R_t = R_w \cup R_e$, where R_e is the set of query rules that were classified with low certainty factor by the VFI algorithm and therefore, then presented to the domain expert for classification.

As a consequence, all the rules in the set R_q are classified either by the concept description learned by the VFI algorithm or the domain expert according to their interestingness class. As the number of rules in R_t increases, VFI begins to construct more powerful and reliable concept descriptions. And as a result, most of the rules in R_q begin to be classified with high certainty factors by the VFI algorithm, rather than the domain expert. When all the rules in R_q are classified, the rules in R_s are reclassified by the most recent version of the concept description. This is required, because any rule classified with a high certainty factor by a weak concept description may now be classified with a low certainty factor by the most recent and powerful version of the concept description. These type of rules are taken from R_s and replaced in R_q. The process restarts as explained in Figure 4, and continues until R_q gets empty.

FPRC ends up with sorting the rules in R_s with respect to their certainty factor and presenting the sorted rules to the user.

```
FPRC ( R, Warm_up_Rule_Count, Min_Occurrence_Rate, MinCt)
begin
    Rwarm-up ← Random_Rule_Select (Warm_up_Rule_Count, Min_Occurrence_Rate)
    Rt ← Rwarm-up
    C ← VFItrain (Rt)        // Initialize the interestingness concept description
    Rq ← R – Rwarm-up        // Rq : the set of rules not used in the warming-up period
    Rs ← ∅

    While Rq is not empty
        for each rule r ∈ Rq
            label r as VFIquery (C, r), let Cf be the certainty factor
            if Cf ≥ MinCt
                insert r into Rs
            else // Cf < MinCt
                ask the expert to classify r
                set Cf of the classification of r as 1
                insert r into Rt
                C ← VFItrain (Rt)
            remove r from Rq
        for each rule r ∈ Rs
            label r as VFIquery (C, r), let Cf be the certainty factor
            if Cf < MinCt
                remove r from Rs
                insert r into Rq

    sort rules in Rs with respect to Cf
end.
```

Figure 4. FPRC algorithm.

5. Experimental Results

A total of 33 rules that were produced from the Gastric Carcinoma data set were classified by the FPRC algorithm. The parameters used in the course of the experiment phase and their associated values are as follows:

Warm_up_Rule_Count: 6
Min_Occurrence_Rate: 30%

Minimum Certainty Factor (*MinC*$_t$): 51% - 60%

For a set of minimum certainty factor (*MinC*$_t$) values, expert participation in the classification process and classification accuracy values were recorded. Furthermore, metric features of the rules were given weights to help the expert to give more importance to the metrics that he may have thought to be more important in the rule classification process (FPRC_weighted). In our experiment with the Gastric Carcinoma data set, experts were more interested with the rules having some specific class labels. Furthermore, in a classification rule of the form $A \rightarrow B$, they found the rules having less number of conditions in the antecedent part *A* as interesting and others as uninteresting. As a consequence, *Class* and *RuleSize* features were given 1 and the remaining features were given 0 as the feature weights in FPRC_weighted algorithm.

Table 2 shows that, in general, FPRC_weighted algorithm not only requires less expert participation but also gives high accuracy values when compared with FPRC. Both of the algorithms achieve high accuracy values even for low $MinC_t$ values. However, FPRC_weighted is shown to achieve these high accuracy values for lower $MinC_t$ values.

Table2. Results. Expert participation is the ratio of the rules classified by the expert (warm-up rules, and other rules classified by the expert later) to the whole rules.

Algorithm	$MinC_t$	Expert Participation	Accuracy
FPRC	51%	18%	52%
FPRC	53%	24%	48%
FPRC	55%	54%	87%
FPRC	57%	63%	100%
FPRC	60%	69%	100%
FPRC_weighted	51%	21%	65%
FPRC_weighted	53%	24%	96%
FPRC_weighted	55%	27%	96%
FPRC_weighted	57%	27%	96%
FPRC_weighted	60%	27%	96%

Expert participation monotonically increases when the minimum certainty factor increases. Because, learning a more powerful concept description is realized by increasing the number of training rule instances. And the number of training rule instances can only be increased by the expert participation. An increase in expert participation not only results in a more powerful concept description, but also leads to more accurate rule classifications, in general.

We were able to obtain accuracy values since the number of rules, 33, were small enough for the expert to classify completely. However, in general, especially in data mining applications, the number of rules may be so large that the expert may not classify all the rules one by one. This is not a problem, because, the development of FPRC algorithm is to classify the rules automatically, while holding the expert participation in a low ratio. Otherwise, if the expert is able to classify all of the rules himself, then there will not be a need for rule classification algorithms such as FPRC.

Finally, each of the metric features that were developed in our research studies constructs distinguishing feature intervals. For example, the expert generally finds rules that have very low and very high *Support* values as uninteresting, where as he finds rules having moderate *Support* values as interesting. Another example can be given for the *RuleSize* feature. The expert may find rules having low *RuleSize* values as interesting, where as he may find rules having high *RuleSize* values as uninteresting. However, to obtain distinguishing feature intervals, expert should give high weights to those features, and low weights to the remaining ones.

6. Conclusion and Future Work

FPRC and FPRC_weighted rule classification algorithms were shown to be successful in the Gastric Carcinoma data set. As a future work, other classification algorithms rather than VFI can be used in the FPRC algorithm. Also, more complex rule metrics can be constructed to use in rule classification process.

References:

[1] Fayyad, U., Shapiro, G., and Smyth, P. "**From data mining to knowledge discovery in databases**" *AI Magazine 17(3)*, 1996, pp.37-54.

[2] Hilderman, R.J., and Hamilton, H.J. "**Knowledge discovery and interestingness measures: a survey**" *Technical Report*, Department of Computer Science, University of Regina, 1999.

[3] Quinlan, J.R. "**C4.5: program for machine learning**" Morgan Kaufmann, 1992.

[4] Ikizler, N., and Guvenir, H.A. "**Mining interesting rules in bank loans data**" *Proceedings of the 10th Turkish Symposium on Artificial Intelligence and Neural Networks*, 2001, pp.238-246.

[5] Güvenir, H.A., Emeksiz, N. and Örmeci, N. "**Diagnosis of gastric carcinoma tumors by classification on feature projections**" *Technical Report*, BU-CE-0206.

[6] Güvenir, H.A., and Demiröz, G. "**Classification by voting feature intervals**" *Proceedings of 9th European Conference on Machine Learning*, 1997, pp.85-92.