A Bayesian Scoring Technique for Mining Predictive and Non-Spurious Rules

Iyad Batal[1], Gregory Cooper[2], and Milos Hauskrecht[1]

[1] Department of Computer Science, University of Pittsburgh
[2] Department of Biomedical Informatics, University of Pittsburgh

Abstract. Rule mining is an important class of data mining methods for discovering interesting patterns in data. The success of a rule mining method heavily depends on the evaluation function that is used to assess the quality of the rules. In this work, we propose a new rule evaluation score - the Predictive and Non-Spurious Rules (PNSR) score. This score relies on Bayesian inference to evaluate the quality of the rules and considers the structure of the rules to filter out spurious rules. We present an efficient algorithm for finding rules with high PNSR scores. The experiments demonstrate that our method is able to cover and explain the data with a much smaller rule set than existing methods.

1 Introduction

The large amounts of data collected today provide us with an opportunity to better understand the behavior and structure of many natural and man-made systems. Rule mining is an important direction of machine learning and data mining research, which aims to elicit knowledge from data in terms of if-then rules that are intuitive and easy to understand by humans.

In this work, we study and apply rule mining to discover patterns in supervised learning tasks, where we have a specific target variable (outcome) and we want to find patterns (subpopulations of data instances) where the distribution of the target variable is statistically "most interesting". Examples of such patterns are: "subpopulation of patients who smoke and have a positive family history are at a significantly higher risk for lung cancer than the rest of the patients". This task has a high practical relevance in many domains of science or business. For example, finding a pattern that clearly and concisely defines a subpopulation of patients that respond better (or worse) to a certain treatment than the rest of the patients can speed up the validation process of this finding and its future utilization in patient-management.

In order to perform supervised rule discovery, we need to define a *search algorithm* to explore the space of potential rules and a *scoring function* to assess the interestingness of the rules. In this work, we use Frequent Pattern Mining (FPM) [1] to search for rules. The advantage of FPM is that it performs a more systematic search than heuristic rule induction approaches, such as greedy sequential covering [7–9]. However, its main disadvantage is that it often produces a large number of rules. Moreover, many of these rules are spurious because they can be naturally explained by other simpler (more general) rules. Therefore, it is

crucial to devise an effective scoring function that allows us to select important and non-redundant rules from a large pool of frequent patterns.

To achieve this goal, we introduce the *Predictive and Non-Spurious Rules* (*PNSR*) score. This score applies Bayesian inference to evaluate the quality of individual rules. In addition, it considers the structure of patterns to assure that every rule is not only predictive with respect to the general population, but also with respect to all of its simplifications (generalizations). We show that using our score to mine the top rules, we are able to cover and explain the data with much fewer rules compared with classical supervised rule discovery methods. Finally, we present an efficient algorithm that integrates rule evaluation with frequent pattern mining and applies pruning strategies to speed up the mining.

2 Supervised Descriptive Rule Discovery

In this work, we are interested in applying rule mining in the *supervised setting*, where we have a special variable of interest Y (the target variable) and we want to mine rules that can help us to uncover "interesting" dependencies between Y and the input variables (attributes).

The dominant paradigm for supervised rule induction is to apply a sequential covering method [7–9], which learns a set of rules by first learning a single rule, removing the positive instances it covers and then repeating the process over the remaining instances. However, this approach is not appropriate for knowledge discovery because the rules are induced from biased data (including only positive instances not covered by previous rules). Therefore, the rules are difficult to interpret and understand by the user.

In contrast to the sequential covering approach, our task is to find a set of *comprehensible* rules/patterns that are statistically interesting with respect to the entire data, e.g., the rules should have wide coverage and unusual distributional characteristics with respect to the target variable [18]. This task appeared in the literature under a variety of different names, such as contrast set mining [2], emerging pattern mining [11] and subgroup discovery [17, 18]. Later on, [23] provided a unifying framework of this work which is named *Supervised Descriptive Rule Discovery* (*SDRD*).

To apply *SDRD*, we need to define a search algorithm to explore the space of potential rules and a scoring function S (quality measure) to assess the interestingness of each rule (S maps each rule R_i to a real number $S(R_i) \in \mathbb{R}$ that reflects its importance). Our objective in this work is to design a function S such that the top rules do not only predict well the target class variable compared to the entire population, but are also non-spurious in that their prediction is better than all of their generalizations (simplifications).

2.1 Definitions

Let $D = \{x_i, y_i\}_{i=1}^{n}$ be our data, where each instance x_i is described by a fixed number of attributes and is associated with a class label $y_i \in dom(Y)$. We assume that all attributes have discrete values (numeric attributes must be discretized [13, 28]).

We call every attribute-value pair an *item* and a conjunction of items a *pattern*. A pattern that contains k items is called a *k-pattern*. For example, *Education = PhD* $\wedge$ *Marital-status = Single* is a *2-pattern*.

Pattern P is a *subpattern* of pattern P', denoted as $P \subset P'$, if every item in P is contained in P' and $P \neq P'$. In this case, P' is a *superpattern* of P. For example, P_1 :*Education=PhD* is a subpattern of P_2 :*Education=PhD* $\wedge$ *Marital-status=Single*. This subpattern (*more-general-than*) relation defines a *partial ordering* of patterns, i.e. a *lattice structure*, as shown in Figure 1.

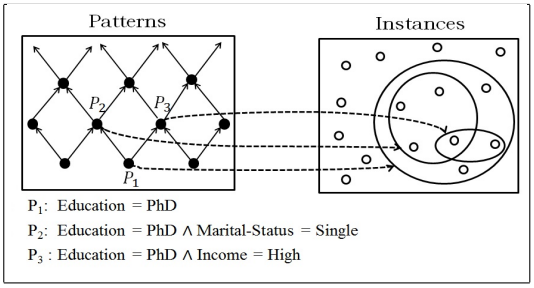

Fig. 1. The box on the left shows the set of all patterns and the box on the right shows the set of all instances. Each pattern is associated with a group of instances that satisfy the pattern. The patterns are organized in a lattice structure according to the subpattern-superpattern relation.

Instance x_i satisfies pattern P, denoted as $P \in x_i$, if every item in P is present in x_i. Every pattern P defines a *group* (subpopulation) of the instances that satisfy P: $G_P = \{(x_i, y_i) : x_i \in D \wedge P \in x_i\}$. If we denote the empty pattern by ϕ, G_ϕ represents the entire data D. Note that $P \subset P'$ (P is a subpattern of P') implies that $G_P \supseteq G_{P'}$ (see Figure 1).

The *support* of pattern P in dataset D, denoted as $sup(P, D)$, is the number of instances in D that satisfy P (the size of G_P). Given a user defined *minimum support* threshold σ, P is called a *frequent pattern* if $sup(P, D) \geq \sigma$.

A rule is defined as $P \Rightarrow y$, where P (the condition) is a pattern and $y \in dom(Y)$ (the consequent) is a class label. We say that $P \Rightarrow y$ is a *subrule* of $P' \Rightarrow y'$ if $P \subset P'$ and $y = y'$. The *coverage* of rule $P \Rightarrow y$ is the proportion of instances in the data that satisfy P. The *confidence* of rule $P \Rightarrow y$, denoted as $conf(P \Rightarrow y)$, is the proportion of instances from class y among all the instances that satisfy P, i.e., it is the maximum likelihood estimation of $Pr(Y = y|P)$.

2.2 Rule Evaluation

A straightforward approach to *SDRD* is to use a rule quality measure (cf [14]) to score each rule by contrasting it to the general population (the entire data) and report the top rules to the user. We will argue that this approach is ineffective and can lead to many spurious (redundant) rules. We start by illustrating the spurious rules problem using an example and then describe it more formally in Section 2.3.

Example 1. Assume our objective is to identify populations of patients who are at high risk of developing coronary heart disease (CHD). Assume our dataset contains 150 instances, 50 of them are CHD cases and the others are controls. That is, the CHD prior, i.e. $conf(\Phi \Rightarrow CHD)$, is $50/150 \approx 33.3\%$.

Now, our task is to evaluate the following 3 rules:

- R_1:Race=White $\Rightarrow$ CHD
  [#cases=29, #controls=61, conf=32.2%]
- R_2:Family history=Yes $\Rightarrow$ CHD
  [#cases=30, #controls=20, conf=60%]
- R_3:Family history=Yes $\wedge$ Race=White $\Rightarrow$ CHD
  [#cases=21, #controls=11, conf=65.6%]

For each rule, we show the number of CHD cases and the number of controls that the rule covers. We also show the confidence of the rule.

One of the commonly used approaches to filter out uninteresting rules is to apply the χ^2 *test* to assure that there is a significant positive correlation between the condition and the consequent of each rule [22, 2, 20, 5]. If we apply the χ^2 test on our three rules, the p-values we get for R_1, R_2, and R_3 are 0.724, 9.6×10^{-7}, and 1.2×10^{-5}, respectively. That is, both R_2 and R_3 are statistically (very) significant with respect to a significance level $\alpha = 0.05$. Moreover, these two rules will be considered interesting using most rule quality measures [14].

[3] proposed the *confidence improvement* constraint, which says that each rule in the result should have a higher confidence than all of its subrules:

$$conf(P \Rightarrow y) - \max_{S \subset P}\{conf(S \Rightarrow y)\} > 0$$

This filter have been used quite a lot in the rule mining literature [15, 26, 20, 19]. If we applied the confidence improvement constraint to our working example, both R_2 and R_3 will be retained.

As we can see, both χ^2 test and confidence improvement agree that R_3 is an interesting rule. However, this rule may seem predictive only because it contains a simpler predictive rule (R_2). So should we consider R_3 to be interesting (show it to the user) or spurious? We will revisit this question after introducing the PNSR score.

2.3 Spurious Rules

Spurious rules are formed by adding irrelevant items to the antecedent of a simpler predictive rule. Let us illustrate this using the simple Bayesian belief network in Figure 2. In this network, the value of the class variable Y only depends on the value of feature F_1 and is independent of the values of the other features: $Y \perp\!\!\!\perp F_i : i \in \{2, ..., n\}$. Assume that pattern $P : F_1 = 1$ is predictive of class $Y = y_1$, so that $Pr(y_1|P) > Pr(y_1)$. Clearly, P is the only important pattern for predicting y_1.

Now consider pattern P' that is a superpattern of P, $P' : F_1 = 1 \wedge F_{q_1} = v_{q_1} \wedge ... \wedge F_{q_k} = v_{q_k}$, where $F_{q_i} \in \{F_2, ..., F_n\}$ and v_{q_i} is *any possible value* of variable F_{q_i}. The network structure implies that $Pr(y_1 \,|\, P') = Pr(y_1 \,|\, P)$, hence $Pr(y_1 \,|\, P')$ is also larger than the prior $Pr(y_1)$.

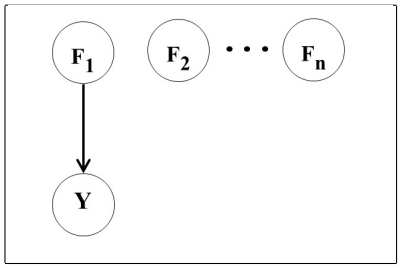

Fig. 2. Illustrating the problem of spurious rules.

The problem is that if we evaluate the rules individually (without considering the nested structure of the patterns), we may falsely think that $P' \Rightarrow y_1$ is an important rule. However, this rule is totally redundant given its subrule $P \Rightarrow y_1$. Even by requiring complex rules to have a higher confidence than their simplifications (the confidence improvement) [3, 15, 26, 20, 19], the problem still exists and many spurious rules can easily satisfy this constraint due to noise in sampling. Clearly, having spurious rules in the results is undesirable because they overwhelm the user and prevent him/her from understanding the real causalities in the data.

3 Mining Predictive and Non-Spurious Rules

In this section, we present our approach for scoring/ranking rules. We start by defining a Bayesian score to evaluate the predictiveness of a rule with respect to a more general population. After that, we introduce the *PNSR-score* to address the problem of spurious rules. Lastly, we present an efficient mining algorithm that integrates rule evaluation with frequent pattern mining.

3.1 Classical Rule Quality Measures

A large number of rule quality measures have been proposed in the literature to evaluate the interestingness of individual rules. Examples of such measures include confidence, lift, weighted relative accuracy, J-measure, and others (cf [14]). Most of these measures trade-off two factors: 1) the *distributional unusualness* of the class variable in the rule compared to the general population and 2) the *coverage* of the rule, which reflects its generality [18, 23]. This trade-off is often achieved in an ad-hoc way, for instance by simply multiplying these two factors as in the weighted relative accuracy score [17] or in the J-measure [25]. Furthermore, most interestingness measures rely on point estimates of these quantities, often using the maximum likelihood estimation, and they do not capture the uncertainty of the estimation. In the following, we present a novel Bayesian score to evaluate the quality of a rule.

3.2 The Bayesian Score

Suppose we want to evaluate rule $P \Rightarrow y$ with respect to a group of instances G where $G_P \subseteq G$. Intuitively, we would like the rule to get a high score when there

is a strong evidence in the data to support the hypothesis that $Pr(Y=y|G_P) > Pr(Y=y|G)$. Our Bayesian score treats these probabilities as random variables as opposed to using their point estimation as in the classical measures [14].

Let us begin by defining M_e to be the model that conjectures that all instances in group G have the **same probability** for having class $Y=y$, even though we are uncertain what that probability is. Let us denote $Pr(Y=y|G)$ by θ. To represent our uncertainty about θ, we use a beta distribution with parameters α and β. Let N_{*1} be the number of instances in G with class $Y=y$ and let N_{*2} be the number of instances in G with class $Y \neq y$. The marginal likelihood for model M_e is as follows:

$$Pr(G|M_e) = \int_{\theta=0}^{1} \theta^{N_{*1}} \cdot (1-\theta)^{N_{*2}} \cdot beta(\theta;\alpha,\beta) d\theta$$

The above integral yields the following well known closed-form solution [16]:

$$Pr(G|M_e) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha+N_{*1}+\beta+N_{*2})} \cdot \frac{\Gamma(\alpha+N_{*1})}{\Gamma(\alpha)} \cdot \frac{\Gamma(\beta+N_{*2})}{\Gamma(\beta)} \tag{1}$$

where Γ is the gamma function.

Let us define M_h to be the model that conjectures that the probability of $Y=y$ in G_P, denoted by θ_1, is different from the probability of $Y=y$ in the instances of G not covered by P ($G \setminus G_P$), denoted by θ_2. Furthermore, M_h believes that θ_1 is **higher** than θ_2. To represent our uncertainty about θ_1, we use a beta distribution with parameters α_1 and β_1, and to represent our uncertainty about θ_2, we use a beta distribution with parameters α_2 and β_2. Let N_{11} and N_{12} be the number of instances in G_P with $Y=y$ and with $Y \neq y$, respectively. Let N_{21} and N_{22} be the number of instances outside G_P with $Y=y$ and with $Y \neq y$, respectively (see Figure 3). Note that $N_{*1} = N_{11} + N_{21}$ and $N_{*2} = N_{12} + N_{22}$.

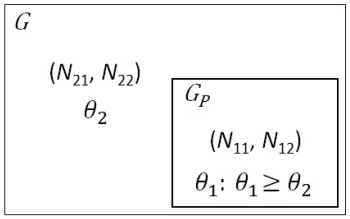

Fig. 3. A diagram illustrating model M_h.

The marginal likelihood for model M_h is defined as follows:

$$Pr(G|M_h) = \int_{\theta_1=0}^{1} \int_{\theta_2=0}^{\theta_1} {\theta_1}^{N_{11}} \cdot (1-\theta_1)^{N_{12}} \cdot {\theta_2}^{N_{21}} \cdot (1-\theta_2)^{N_{22}} \cdot \frac{beta(\theta_1;\alpha_1,\beta_1) \cdot beta(\theta_2;\alpha_2,\beta_2)}{k} d\theta_2 d\theta_1 \tag{2}$$

where k is a normalization constant for the parameter prior[3]. Note that this formula does not assume that the parameters are independent, but rather constrains θ_1 to be higher than θ_2.

Below we show the closed form solution we obtained by solving Equation 2. The derivation of the solution is omitted in this manuscript due to space limitation[4].

$$Pr(G|M_h) = \frac{1}{k} \cdot \frac{\Gamma(\alpha_1+\beta_1)}{\Gamma(\alpha_1)\Gamma(\beta_1)} \cdot \frac{\Gamma(\alpha_2+\beta_2)}{\Gamma(\alpha_2)\Gamma(\beta_2)} \cdot \sum_{j=a}^{a+b-1} \left(\frac{\Gamma(a)\Gamma(b)}{\Gamma(j+1)\Gamma(a+b-j)} \cdot \frac{\Gamma(c+j)\Gamma(a+b+d-j-1)}{\Gamma(a+b+c+d-1)} \right) \tag{3}$$

where $a = N_{21} + \alpha_2$, $b = N_{22} + \beta_2$, $c = N_{11} + \alpha_1$, $d = N_{12} + \beta_1$. We solve for k by applying Equation 3 (without the k term) with $a = \alpha_2$, $b = \beta_2$, $c = \alpha_1$ and $d = \beta_1$.

Equation 3 can be expressed in logarithmic form (to avoid computing very large numbers). Its computational complexity is $O(b) = O(N_{22} + \beta_2)$ (the number of terms in the summation). It turns out that we can redefine the solution of Equation 2 so that its computational complexity is $O(min(N_{11} + \alpha_1, N_{12} + \beta_1, N_{21} + \alpha_2, N_{22} + \beta_2))$. The modifications that achieve this complexity result are omitted due to space limitation[5].

Lastly, let M_l be the model that conjectures that θ_1 is **lower** than θ_2. The marginal likelihood for M_l is similar to Equation 2, but integrates θ_2 from 0 to 1 and constrains θ_1 to be integrated from 0 to θ_2 (forcing θ_1 to be smaller than θ_2). The solution for $P(G|M_l)$ can reuse the terms computed in Equation 3 and can be computed with complexity O(1).

Now that we computed the marginal likelihood for models M_e, M_h and M_l, we compute the posterior probability of M_h (the model of interest) using Bayes theorem:

$$Pr(M_h|G) = \frac{Pr(G|M_h)Pr(M_h)}{Pr(G|M_e)Pr(M_e) + Pr(G|M_h)Pr(M_h) + Pr(G|M_l)Pr(M_l)} \tag{4}$$

To be "non-informative", we might simply assume that all three models are equally likely a-priori: $Pr(M_e) = Pr(M_h) = Pr(M_l) = \frac{1}{3}$.

Equation 4 quantifies in a Bayesian way how likely (a posteriori) is the model which presumes $Pr(Y = y|G_P)$ is higher than $Pr(Y = y|G)$. Since this is the quantity we are interested in, we use $Pr(M_h|G)$ to score rule $P \Rightarrow y$ with respect to group G. We denote this Bayesian score by $\boldsymbol{BS(P \Rightarrow y, G)}$.

Example 2. Let us use the Bayesian score to evaluate rule R_2: Family history=Yes $\Rightarrow$ CHD in Example 1. We evaluate R_2 with respect to the entire dataset G_ϕ by computing $BS(R_2, G_\phi)$. Using the notations introduced earlier, $N_{*1} = 50$ and $N_{*2} = 100$ (the number of cases and controls in the dataset). Also, $N_{11} = 30$, $N_{12} = 20$, $N_{21} = N_{*1} - N_{11} = 20$ and $N_{22} = N_{*2} - N_{12} = 80$. Let us use uniform beta priors for all parameters: $\alpha = \beta = \alpha_1 = \beta_1 = \alpha_2 = \beta_2 = 1$. The likelihood

[3] $k = \frac{1}{2}$ if we use uniform priors on both parameters by setting $\alpha_1 = \beta_1 = \alpha_2 = \beta_2 = 1$.
[4] The derivation can be found on the author's website: `www.cs.pitt.edu/~iyad`.
[5] These modifications can found on the author's website: `www.cs.pitt.edu/~iyad`.

of M_e is 3.2×10^{-43}, the likelihood of M_h is 1.5×10^{-38} and the likelihood of M_l is 1×10^{-44}. Hence, $BS(R_2, G_\phi) = Pr(M_h|G_\phi) = 0.99998$. This implies that there is a strong evidence in the data to conclude that pattern Family history=yes makes CHD more likely.

3.3 The Predictive and Non-Spurious Rules Score

The Bayesian score proposed in the previous section provides a way to evaluate the predictiveness of a rule by contrasting it to a more general population than the population covered by the rule. One approach to supervised descriptive rule discovery is to score each rule R_i with respect to the entire data $BS(R_i, G_\phi)$ and report the top rules to the user. However, this approach does not overcome the spurious rules problem: if a rule $P \Rightarrow y$ achieves a very high score, many spurious rules $P' \Rightarrow y$: $P' \supset P$ are expected to have a high score as well (provided that P' have enough support in the data). As a result, the rules presented to the user would contain a lot of redundancies and fail to provide a good coverage of the data.

To overcome this problem, we propose the *Predictive and Non-Spurious Rules score*, denoted as *PNSR-score*, which we define as follows:

$$\textit{PNSR-score}(P \Rightarrow y) = \min_{S:S\subset P}\{BS(P \Rightarrow y, G_S)\}$$

If a rule R achieves a high *PNSR-score*, then there is a strong evidence in the data not only to conclude that R improves the prediction of its consequent with respect to the entire data, but also with respect to the data matching any of its subrules. That is, the rule's effect on the class distribution cannot be explained by any more general rule that covers a larger population. This implies that every item in the condition of the rule is an important contributor to its predictiveness (the rule is concise).

Example 3. Let us go back to Example 1 and compute the *PNSR-score* for rule R_3. If we evaluate R_3 with respect the entire dataset, $BS(R_3, G_\phi) = 0.9997$. If we evaluate R_3 with respect to subrule R_1, $BS(R_3, G_{R_1}) = 0.999992$. Finally, if we evaluate R_3 with respect to subrule R_2, $BS(R_3, G_{R_2}) = 0.47$. We can see that R_3 is considered very predictive when compared to the entire dataset or to subrule R_1, but is not predictive when compared to subrule R_2. Therefore, we do not consider R_3 an important rule because it is equivocal whether it predicts CHD as being more likely than does R_2.

Example 4. Let us consider again the simple Bayesian network in Figure 2. Assume we have 10 binary features (F_1 to F_{10}) and the CPTs are defined as follows: $Pr(F_i = 1) = 0.4 : i \in \{1, ..., 10\}$, $Pr(Y = y_1|F_1 = 1) = 0.9$ and $Pr(Y = y_1|F_1 = 0) = 0.5$. Let the data D be 500 instances that are randomly generated from this network and let us use D to mine rules that are predictive of class y_1[6]. As we discussed earlier, the only important rule for predicting y_1 is $F_1 = 1 \Rightarrow y_1$ and all other rules are just spurious.

[6] The prior of y_1 in this network is $Pr(Y = y_1) = 0.66$.

We use frequent pattern mining to explore all patterns that occur in more than 10% of the data. Doing so, we obtain 1,257 frequent patterns (potential rules). If we apply the χ^2 test with significance level $\alpha = 0.05$, we get 284 rules that positively predicts y_1 and are statistically significant. Even if we apply the False Discovery Rate (FDR) technique [4] to correct for multiple hypothesis testing, we get 245 positive significant rules! If we use our Bayesian score to evaluate each rule (individually) with respect to the entire dataset and report rules with $BS(R_i, G_\phi) \geq 0.95$, we get 222 rules[7]. Note that this approach still suffers from the spurious rules problem. Let us now apply the confidence improvement constraint to filter out "non-productive" rules [3, 15, 26, 20, 19]. By doing so, we get 451 rules! This clearly demonstrates that the confidence improvement constraint is ineffective for removing spurious rules. Lastly, let us use our proposed *PNSR-score* and report rules with *PNSR-score*$(R_i) \geq 0.95$. Doing so, we obtain only a single rule $F_1 = 1 \Rightarrow y_1$ (the only important rule) and effectively filter out all other spurious rules.

3.4 The Mining Algorithm

In this section, we present the algorithm for mining predictive and non-spurious rules. The algorithm utilizes frequent pattern mining to explore the space of potential rules and applies the *PNSR-score* to evaluate the rules.

To search for rules, we partition the data according to the class labels $y \in dom(Y)$ and mine frequent patterns for each class separately (using a local minimum support σ_y that is related to the number of instances from class y). The reason for doing this as opposed to mining frequent patterns from the entire data is that when the data is unbalanced, exploring only patterns that are globally frequent may result in missing many important rules for the rare classes.

The mining algorithm takes as input 1) the data instances from class y: $D_y = \{(x_i, y_i) : y_i = y\}$, 2) the data instances that do not belong to class y: $D_{\neg y} = \{(x_i, y_i) : y_i \neq y\}$, 3) the local minimum support threshold σ_y and 4) a user specified significance parameter g. The algorithm explores the space of frequent patterns and outputs the rules with *PNSR-score* higher than g.

A straightforward way to obtain the result is to apply the commonly used two-phase approach as in [6, 26, 27, 17, 12, 10, 20], which generates all frequent patterns in the first phase and evaluates them in the second phase (a post-processing phase). That is, we need to perform the following two steps:

1. Phase I: Mine all frequent patterns: $FP = \{P_1, ..., P_m : sup(P_i) \geq \sigma_y\}$
2. Phase II: For each pattern $P_i \in FP$, output rule $P_i \Rightarrow y$ if *PNSR-score*$(P_i \Rightarrow y) \geq g$.

In contrast to this two-phase approach, our algorithm integrates rule evaluation with frequent pattern mining, which allows us to apply additional pruning techniques that are not applicable in the two-phase approach.

The mining algorithm explores the lattice of frequent patterns level by level from the bottom-up starting from the empty pattern. That is, the algorithm

[7] The 0.95 threshold is chosen so that it is comparable to the commonly used frequentist 0.05 significance level.

first explores frequent *1-patterns*, then frequent *2-patterns*, and so on. When the algorithm visits a frequent pattern P (a node in the lattice), it computes the *PNSR-score* of rule $P \Rightarrow y$ and adds it to result if $\textit{PNSR-score}(P \Rightarrow y) \geq g$.

Lossless pruning: We now illustrate how to utilize the *PNSR-score* to prune portions of the search space that are guaranteed not to contain any result.

We say that pattern P is **pruned** if we do not explore any of its superpatterns ($P' \supset P$). This means that we exclude the entire sublattice with bottom P from the lattice of patterns we have to explore.

Frequent pattern mining relies only on the *support* of the patterns to prune infrequent patterns according to the following anti-monotone property: if a pattern is not frequent, all of its superpatterns are guaranteed not to be frequent.

By integrating rule evaluation with frequent pattern mining, we can apply an additional pruning technique. The idea is to prune pattern P if we guarantee that none of its superpatterns will be in the result:

$$\textit{Prune } P \text{ if } \forall P' \supset P : \textit{PNSR-score}(P' \Rightarrow y) < g$$

However, since patterns are explored in a level-wise fashion, we do not know the class distribution in the superpatterns of P. But we know that for any $P' \supset P$: $G_{P'} \subseteq G_P$, and hence $sup(P', D_y) \leq sup(P, D_y) \wedge sup(P', D_{\neg y}) \leq sup(P, D_{\neg y})$.

We now define the *optimal superpattern* of P with respect to class y, denoted as P^*, to be a hypothetical pattern that covers all instances from y and none of the instances from the other classes:

$$sup(P^*, D_y) = sup(P, D_y) \wedge sup(P^*, D_{\neg y}) = 0$$

P^* is the best possible superpattern for predicting y that P can generate. Therefore, $\textit{PNSR-score}(P^* \Rightarrow y)$ is an upper bound on the *PNSR-score* for the superpattern of P. Now, we safely prune P when $\textit{PNSR-score}(P^* \Rightarrow y) < g$.

4 Experimental Evaluation

The experiments compare the performance of different rule quality measures for the problem of supervised descriptive rule discovery. In particular, we compare the following measures:

1. *GR*: Rules are ranked using the Growth Rate measure, which was used in [11] in the context of emerging pattern mining.

$$GR(P \Rightarrow y) = \frac{sup(P, D_y)/|D_y|}{sup(P, D_{\neg y})/|D_{\neg y}|}$$

   where D_y and $D_{\neg y}$ represent the instances from class y and not from class y, respectively.
2. *J-measure*: Rules are ranked using the J-measure [25], a popular information theoretic measure that scores the rules by their information content.

$$\textit{J-measure}(P \Rightarrow y) = \frac{sup(P, D)}{|D|} \times \sum_{z \in \{y, \neg y\}} conf(P \Rightarrow z) \cdot log_2 \left(\frac{conf(P \Rightarrow z)}{conf(\Phi \Rightarrow z)} \right)$$

3. *WRAcc*: Rules are ranked using the Weighted Relative Accuracy, which was used in [17] in the context of subgroup discovery[8].

$$WRAcc(P \Rightarrow y) = \frac{sup(P, D)}{|D|} \times (conf(P \Rightarrow y) - conf(\Phi \Rightarrow y))$$

   Note that this measure is compatible (provides the same rule ranking) with the support difference heuristic used in [2] for contrast set mining (see [23]).
4. *BS*: Rules are ranked using our proposed Bayesian score. However, this method scores each rule individually with respect to the entire data and do not filter out spurious rules.
5. *Conf-imp*: Only rules that satisfy the confidence improvement constraint are retained [3, 15, 26, 20, 19] and they are ranked according to their confidence.
6. *PNSR*: Only rules R_i that have a *PNSR-score*$(R_i) \geq 0.95$ are retained[9] and they are ranked according to the Bayesian score.

Note that the *GR* measure does not consider the coverage of the rule when assessing its interestingness. For example, *GR* favors a rule that covers 8% of the instances of in one class and 1% of the instances in the other classes over a rule that covers 70% of the instances of in one class and 10% of the instances in the other classes (as $\frac{8}{1} > \frac{70}{10}$). As a result, *GR* often chooses rules that are very specific (with low coverage) and do not generalize well. To overcome this, the *J-measure* and *WRAcc* explicitly incorporate the rule coverage $\frac{Sup(P,D)}{|D|}$ in their evaluation functions to favor high coverage rules over low coverage rules. This is done by multiplying the rule coverage with a factor that quantifies the distributional surprise (unusualness) of the class variable in the rule (the cross entropy for *J-measure* and the relative accuracy for *WRAcc*). However, it is not clear whether simply multiplying these two factors leads to the optimal trade-off. On the other hand, *BS* achieves this trade-off automatically by modeling the uncertainty of the estimation (the more data we have, the more certain we are about the estimated probabilities).

Note that the first four methods (*GR*, *J-measure*, *WRAcc* and *BS*) evaluate each rule individually with respect to the entire data and do not consider the nested structure of rules. On the other hand, *conf-imp* and *PNSR* evaluate each rule with respect to all of its subrules. *Conf-imp* simply requires each rule have a higher confidence than its subrules, while *PNSR* requires each rule to show a substantial evidence that it improves the prediction over its subrules, which is evaluated using our proposed *PNSR-score*.

For all methods, we use frequent pattern mining to explore the space of potential rules and we set the local minimum support (σ_y) to 10% the number of instance in the class. For *BS* and *PNSR*, we use uniform beta priors (uninformative priors) for all parameters.

[8] The algorithm by [17] uses weighted sequential covering and modifies the WRAcc measure to handel example weights.

[9] The 0.95 threshold is chosen so that it is comparable to the commonly used frequentist 0.05 significance level.

4.1 Datasets

We evaluate the performance of the different rule quality measures on 15 public datasets from the UCI Machine Learning repository. We discretize the numeric attributes into intervals using Fayyad and Irani discretization [13]. Table 1 shows the main characteristics of the datasets.

dataset	# features	# instances	# classes
Lymphography	18	142	2
Parkinson	22	195	2
Heart	13	270	2
Hepatitis	19	155	2
Diabetes	8	768	2
Breast cancer	9	286	2
Nursery	8	12,630	3
Red wine	11	1,599	3
Mammographic	5	961	2
Tic tac toe	9	958	2
Ionosphere	34	351	2
Kr vs kp	36	3,196	2
Pen digits	16	10,992	10
Zoo	16	74	3
WDBC	30	569	2

Table 1. UCI Datasets characteristics

4.2 Quality of Top-K Rules

For a set of rules to be practically useful, the rules should be accurate to predict the class label of unseen data instances (high precision) and the rule set should provide a good coverage of the data (high recall).

In this section, we compare the different rule evaluation measures according to the quality of the top rules. In particular, for each of the compared evaluation measures, we mine the top k rules from the training data and use them to classify the testing data. The classification is done according to the highest confidence rule [21]:

$$Prediction(x) = \arg\max_{y_i}\{conf(P \Rightarrow y_i)\colon P \in x\}$$

The classification performance is evaluated using the F1 score [24], which is the harmonic mean of the precision and recall. All results are reported using a 10-fold cross-validation scheme, where we use the same train/test splits for all compared methods.

Figure 4 shows the classification performance for the different number of top rules. We can see that *GR* is the worst performing method for most datasets. The reason is that rules with the highest *GR* scores are usually very specific (low coverage) and may easily overfit the training data. The other measures (*J-measure*, *WRAcc* and *BS*) perform better than *GR* because they favor high-coverage rules over low-coverage rules, which results in rules that generalize

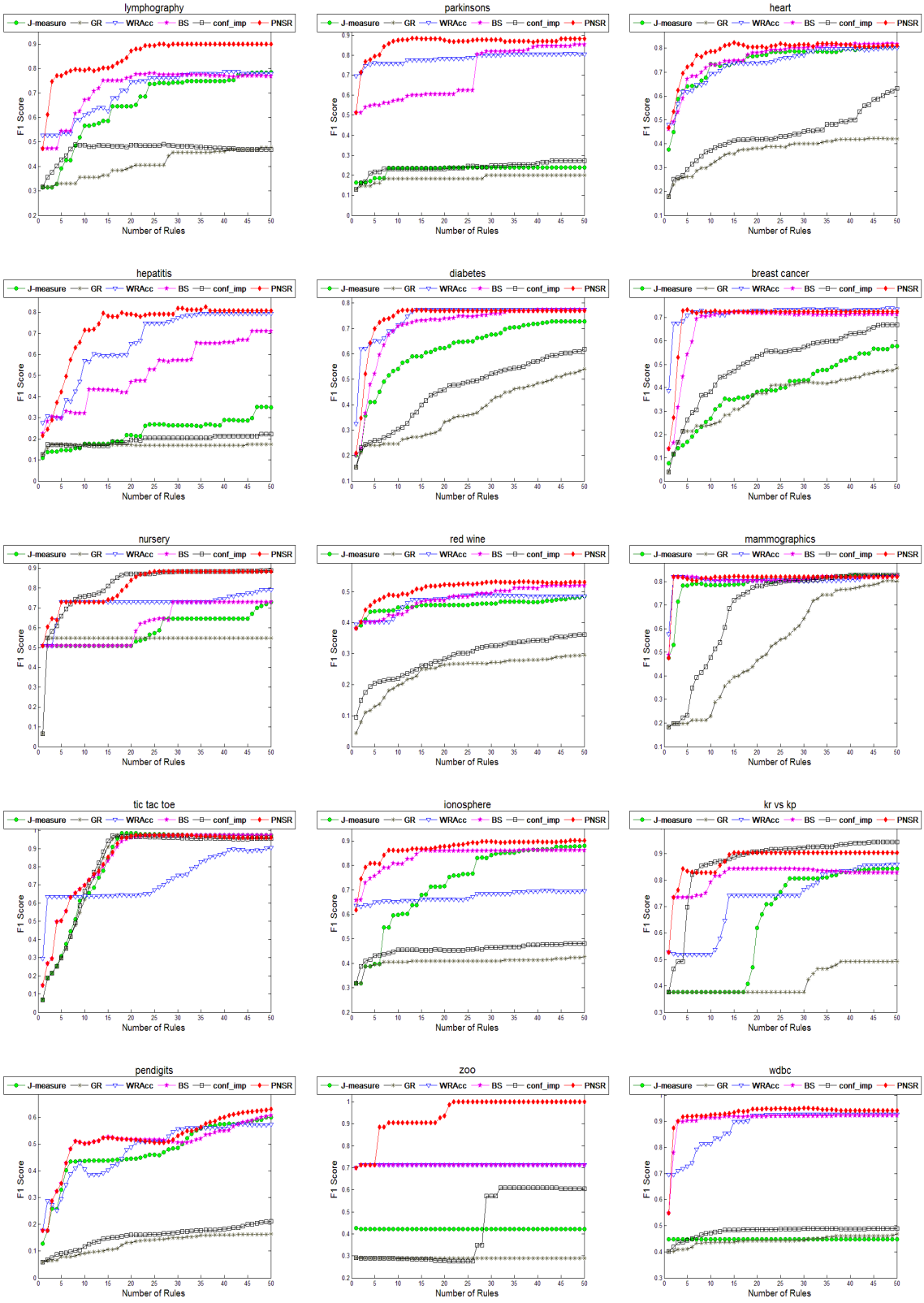

Fig. 4. Comparison of the performance of several rule evaluation measures. The X-axis is the number of the top rules and the Y-axis is the F1 score of the rule set.

better on the testing data. However, because these measures do not consider the relations among the rules, the top rules contain many spurious rules (rules describing the same underlying pattern and are small variations of each other). As a result, they fail to provide a good coverage of the data (see for example the *lymphography* and the *zoo* datasets). Finally, we can see that for most datasets, *PNSR* achieves the best performances with the smallest number of rules.

4.3 Mining Efficiency

In this section, we study the efficiency of our mining algorithm. In particular, we compare the running time of the following methods:

1. *FPM*: Frequent patterns mining, where we partition the data according to the class label and mine frequent patterns for each class (see Section 3.4). We apply the algorithm by [29], which mines frequent patterns using the vertical data format.
2. *PNSR-Naive*: The naive two-phase implementation for mining predictive and non-spurious rules, which applies *FPM* to mine all frequent patterns and then computes the *PNSR-score* of the patterns.
3. *PNSR*: Our mining algorithm, which integrates rule evaluation with frequent pattern mining and applies the lossless pruning technique described in Section 3.4 to prune the search space.

The running time is measured on a Dell Precision T1600 machine with an Intel Xeon 3GHz CPU and 16GB of RAM. As before, we set the local minimum support (σ_y) to 10% the number of instance in the class. Table 2 shows the execution time (in seconds) of the compared methods on the UCI datasets.

dataset	*FPM*	*PNSR-Naive*	*PNSR*
Lymphography	328	410	153
Parkinson	9,865	11,229	800
Heart	45	69	37
Hepatitis	1,113	1,284	391
Diabetes	3	5	5
Breast cancer	3	5	4
Nursery	2	9	9
Red wine	28	52	50
Mammographic	1	1	1
Tic tac toe	3	4	4
Ionosphere	16,899	19,765	1,077
Kr vs kp	1,784	2,566	2,383
Pen digits	71	144	138
Zoo	185	244	23
WDBC	2,348	4,320	282

Table 2. The execution time (in seconds) of frequent pattern mining (*FPM*), two-phase PNSR mining (*PNSR-Naive*) and our mining algorithm (*PNSR*).

The results show that on seven of the fifteen datasets (*lymphography*, *Parkinson*, *Heart*, *Hepatitis*, *Ionosphere*, *Zoo* and *WDBC*), *PNSR* is more efficient than

FPM, which is the cost of the first phase of any two-phase method [6, 26, 27, 17, 12, 10, 20]. For some of these datasets, *PNSR* drastically improves the efficiency. For example, on the *Parkinson*, *Ionosphere* datasets, *PNSR* is more than an order of magnitude faster than *FPM*. This shows that utilizing the predictiveness of patterns to prune the search space can greatly help improving the mining efficiency.

5 Conclusion

In this paper, we study the problem of supervised descriptive rule discovery and propose a new rule evaluation score, the Predictive and Non-Spurious Rules (PNSR) score. This score relies on Bayesian inference to measure the quality of the rules. It also considers the structure of the patterns to ensure that each rule in the result offers a significant predictive advantage over all of its generalizations. We present an algorithm for mining rules with high PNSR scores, which efficiently integrates rule evaluation with frequent pattern mining. The experimental evaluation shows that our method is able to explain and cover the data with fewer rules than existing methods, which is beneficial for knowledge discovery.

6 Acknowledgments

This work was supported by grants 1R01GM088224-01 and 1R01LM010019-01A1 from the NIH. Its content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, 1994.
2. S. D. Bay and M. J. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5:213–246, 2001.
3. R. J. Bayardo. Constraint-based rule mining in large, dense databases. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 1999.
4. Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 57(1):289–300, 1995.
5. S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: generalizing association rules to correlations. In *Proceedings of the international conference on Management of data (SIGMOD)*, 1997.
6. H. Cheng, X. Yan, J. Han, and C. wei Hsu. Discriminative frequent pattern analysis for effective classification. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2007.
7. P. Clark and T. Niblett. The cn2 induction algorithm. *Machine Learning*, 1989.
8. W. Cohen. Fast effective rule induction. In *Proceedings of International Conference on Machine Learning (ICML)*, 1995.
9. W. Cohen and Y. Singer. A simple, fast, and effective rule learner. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1999.

10. M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering*, 17:1036–1050, 2005.
11. G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proceedings of the international conference on Knowledge discovery and data mining (SIGKDD)*, 1999.
12. T. P. Exarchos, M. G. Tsipouras, C. Papaloukas, and D. I. Fotiadis. A two-stage methodology for sequence classification based on sequential pattern mining and optimization. *Data and Knowledge Engineering*, 66:467–487, 2008.
13. U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1993.
14. L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys*, 38, 2006.
15. H. Grosskreutz, M. Boley, and M. Krause-Traudes. Subgroup discovery for election analysis: a case study in descriptive data mining. In *Proceedings of the international conference on Discovery science*, 2010.
16. D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 1995.
17. B. Kavsek and N. Lavrač. APRIORI-SD: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, 20(7):543–583, 2006.
18. N. Lavrač and D. Gamberger. Relevancy in constraint-based subgroup discovery. In *Constraint-Based Mining and Inductive Databases*, 2005.
19. J. Li, H. Shen, and R. W. Topor. Mining optimal class association rule set. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2001.
20. W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proceedings of the International Conference on Data Mining (ICDM)*, 2001.
21. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Knowledge Discovery and Data Mining*, pages 80–86, 1998.
22. S. Nijssen, T. Guns, and L. De Raedt. Correlated itemset mining in roc space: a constraint programming approach. In *Proceedings of the international conference on Knowledge discovery and data mining (SIGKDD)*, 2009.
23. P. K. Novak, N. Lavrač, and G. I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research (JMLR)*, 10:377–403, 2009.
24. F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 2002.
25. P. Smyth and R. M. Goodman. An information theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and Data Engineering*, 1992.
26. G. I. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.
27. D. Xin, H. Cheng, X. Yan, and J. Han. Extracting redundancy-aware top-k patterns. In *Proceedings of the international conference on Knowledge discovery and data mining (SIGKDD)*, 2006.
28. Y. Yang, G. I. Webb, and X. Wu. Discretization methods. In *The Data Mining and Knowledge Discovery Handbook*, pages 113–130. Springer, 2005.
29. M. J. Zaki. Scalable algorithms for association mining. *IEEE Transaction on Knowledge and Data Engineering (TKDE)*, 12:372–390, 2000.