Negative Encoding Length as a Subjective Interestingness Measure for Groups of Rules

Einoshin Suzuki

Department of Informatics, ISEE, Kyushu University
suzuki@i.kyushu-u.ac.jp

Abstract. We propose an interestingness measure for groups of classification rules which are mutually related based on the Minimum Description Length Principle. Unlike conventional methods, our interestingness measure is based on a theoretical background, has no parameter, is applicable to a group of any number of rules, and can exploit an initial hypothesis. We have integrated the interestingness measure with practical heuristic search and built a rule-group discovery method CLARDEM (Classification Rule Discovery method based on an Extended-Mdlp). Extensive experiments using both real and artificial data confirm that CLARDEM can discover the correct concept from a small noisy data set and an approximate initial concept with high "discovery accuracy".

1 Introduction

The most serious problem in rule discovery would be the interestingness problem: typically many rules are discovered but most of them are uninteresting [4,13]. Solutions for this problem can be classified into the objective approach [11,12,13], which uses only data as input, and the subjective approach [3,4], which uses also user-supplied information in addition to data. In both approaches, an interestingness measure [3,11,12,13], which is a function for estimating the degree of the interestingness of a rule, is actively studied.

Despite the numerous studies on interestingness measures, few of them have a theoretical background, are parameter-free, can discover a group of rules which are mutually related, and can exploit an initial hypothesis. We attribute the reasons to the subjective nature of interestingness and the high time complexity. [3,4,11,12] are exceptions for some of them but none satisfy these four conditions. Moreover, as far as we know, no study has ever made a systematic investigation on the discovered rules under noisy data and incorrect user-supplied information.

The Minimum Description Length Principle (MDLP) is a principle that the best hypothesis that can be inferred from data is the one that has the shortest "(code length of the hypothesis) + (code length of the data using the hypothesis)" [5,8,9,15]. The MDLP is based on a solid theoretical framework, has a clear interpretation, is robust to noise, and requires no parameter specification. In association rule discovery, the MDLP has been applied to the problem of discovering frequent itemsets [11]. However, the discovered patterns are still large in number and are unrelated. Moreover, the method belongs to the objective

T. Theeramunkong et al. (Eds.): PAKDD 2009, LNAI 5476, pp. 220–231, 2009.

approach thus a theoretical framework which can be integrated for exploiting user-supplied information is unknown.

We restrict our attention to the classification rule [4], which has a class label in its conclusion and has been well-studied due to its importance. It is not obvious how to apply the MDLP for classification to the classification-rule discovery problem since a classifier can be applied to any example unlike a typical group of rules. Moreover, the standard MDLP cannot exploit an initial hypothesis and the MDLP extended for this purpose [14] has problems such as a redundancy in its encoding method. In summary, the MDLP has problems to be used in developing a method with a theoretical background for discovering a group of rules which are mutually related by exploiting user-supplied information. To resolve these problems we formalize the discovery problem of interesting classification rules as an estimation problem of a partial decision list, extend the MDLP for classification so that it can exploit an initial hypothesis, invent an encoding method, and use the negative encoding length as our interestingness measure.

2 Preliminaries

2.1 MDL for Classification

A data set D consists of n examples $d_1, d_2, \ldots, d_n$. Each example d_i is described with m attributes $a_1, a_2, \ldots, a_m$ as an attribute value vector $(v_{i1}, v_{i2}, \ldots, v_{im})$ and belongs to one of M classes, of which labels are represented by $c_1, c_2, \ldots, c_M$. A classifier is a function which outputs a class label given an attribute value vector. We call the process of learning a classifier from D classification.

As a principle for preferring a classifier in classification, the MDLP states that the best classifier T_{MDL} is given as follows [5,8,9,15].

$$T_{\mathrm{MDL}} \equiv \arg\min_{T} \left(-\log P(T) - \log P(D|T)\right) \tag{1}$$

where $P(T)$ and $P(D|T)$ represent the probability that T occurs and the conditional probability that D occurs given T, respectively. Consider the problem of encoding T as a binary string. According to the coding theory [10], the length of the code string for T using an optimally efficient code is $-\log P(T)$. Similarly, $-\log P(D|T)$ may be regarded as the length of the code string for D encoded using T. In the MDLP for classification, these code lengths are calculated in a problem where the receiver has D except for the class labels. The sender first sends T then the class labels of examples in D using T.

It is straightforward to show T_{MDL} coincides with the maximum a posteriori hypothesis. The MDLP can be interpreted as assigning priors to theories based on a compact coding: $P(T)$ is defined by the encoding method for $-\log P(T)$.

2.2 Preliminaries for Encoding

Firstly we consider a problem of sending a binary string of length x which consists of y binary 1s and $(x - y)$ binary 0s. A common method first sends the number

y of binary 1s with code length $\log(x+1)$ then specifies the positions of binary 1s [8,15]. The required code length is denoted with $\Theta(x,y)$.

$$\Theta(x,y) \equiv \log(x+1) + \log \binom{x}{y}$$

For example, "1110010" is sent with a code length $\Theta(7,4) = 8.13$ bits. Note that we do not have to generate the binary message for our purpose.

If we know that $y > 0$, the number y of binary 1s can be sent with code length $\log x$. The required code length in this case is denoted with $\Theta_0(x,y)$.

$$\Theta_0(x,y) \equiv \log x + \log \binom{x}{y}$$

Likewise, we consider a problem of sending a string of length x described with M symbols where the ith symbol occurs x_i times. The sender first sends the numbers $x_1, x_2, \ldots, x_{M-1}$ then specifies the positions except that of the last symbol. We denote the required code length with $H(x,(x_1,x_2,\ldots,x_M),M)$.

$$H(x,(x_1,x_2,\ldots,x_M),M) \equiv \sum_{i=1}^{M-1} \log \left(x+1-\sum_{j=1}^{i-1} x_j \right) + \log \left(\frac{x!}{x_1!x_2!\ldots x_M!} \right) \tag{2}$$

For example, "AACBBAB" is sent in $H(7,(3,3,1),3) = 12.45$ bits.

Lastly we consider a problem of sending a positive integer x under the assumption that $x = y$ is most likely and the occurrence probability $P(i)$ of $x = i$ is given by $P(y)(1/2)^{|y-i|}$. This setting may be interpreted as the length for sending i is longer than that for sending y by $|y-i|$ bits. Since $P(1)+P(2)+\cdots = 1$, the length $-\log\left[P(y)(1/2)^{|y-x|}\right]$, which is required to send x and is denoted with $\Lambda(x,y)$, is given as follows.

$$\Lambda(x,y) \equiv \log\left[3-\left(\frac{1}{2}\right)^y\right] + |y-x| \tag{3}$$

2.3 Classification-Rule Discovery Problem

We call an assignment $a = v$ of a value v to an attribute a an atom. A literal is defined as either a single atom or a conjunction of multiple atoms. An example $(v_{i1}, v_{i2}, \ldots, v_{im})$ is said to satisfy a literal δ if every atom in δ is included in $\{a_1 = v_{i1}, a_2 = v_{i2}, \ldots, a_m = v_{im}\}$. We define a distribution rule r as $r \equiv \rho(r) \to (P_1, P_2, \ldots, P_M)$, where its premise $\rho(r)$ is a literal and its conclusion is a probabilistic distribution $P_1, P_2, \ldots, P_M$ over the classes $1, 2, \ldots, M$.

A partial decision list T, which may be interpreted as a decision list without the default class label, consists of μ distribution rules $r_1, r_2, \ldots, r_\mu$ i.e. $T \equiv r_1, r_2, \ldots, r_\mu$. For a partial decision list $r_1, r_2, \ldots, r_\mu$, a rule r_j is said to cover an example e iff. (if and only if) e does not satisfy $\rho(r_i)$ $(i = 1, 2, \ldots, j-1)$ but satisfies $\rho(r_j)$. We believe that a partial decision list is adequate as the

representation of a hypothesis as it represents a group of rules which are mutually related in a separate-and-conquer manner. The set of examples each of which is satisfied by a distribution rule in a partial decision list T is denoted with $D(T)$.

A null partial decision list B consists of ν distribution rules $b_1, b_2, \ldots, b_\nu$ without conclusions i.e. $B \equiv b_1, b_2, \ldots, b_\nu$. We believe that a null partial decision list is adequate as the representation of an initial hypothesis since it is easier to be obtained from domain experts or textbooks. A partial decision list (or a null partial decision list) which satisfies $\mu = 0$ (or $\nu = 0$) is denoted with $\emptyset$ and is called a null hypothesis.

Partial classification [1] has been mainly studied in the context of decision making. As the objective of a data mining process is not usually restricted to prediction, neither a utility function nor a cost function is adequate for evaluating the goodness of our partial decision list T. In a domain where there is a ground truth i.e. for our case a "correct" partial decision list T_{true}, we define, as an evaluation index, the discovery accuracy $E(\mathcal{M})$ of a rule-group discovery method $\mathcal{M}$ as $E(\mathcal{M}) \equiv \frac{\Upsilon(T_{\mathcal{M}}=T_{\text{true}})}{\Upsilon}$, where Υ is the total number of different trials and $\Upsilon(T_M = T_{\text{true}})$ is the number of trials in each of which the hypothesis $T_{\mathcal{M}}$ returned by $\mathcal{M}$ is equivalent to T_{true}.

[Classification-rule Discovery Problem] Given a data set D and a null partial decision list B as an initial hypothesis, discover a partial decision list T. The goodness of a discovery method $\mathcal{M}$ is evaluated with its discovery accuracy $E(\mathcal{M})$ if a correct partial decision list T_{true} is known.

3 Our Method CLARDEM

3.1 Incorporating Background Knowledge

The MDLP for classification (1) cannot handle an initial hypothesis B thus cannot be applied to our discovery problem directly. We have extended the original MDLP for classification so that T is inferred from D and B. The best hypothesis T_{EMDL} chosen by our extended MDLP is stated as follows.

$$T_{\text{EMDL}} \equiv \arg\min_T \left(-\log P(T) - \log P(D|T) - \log P(B|T)\right)$$

A unique feature of our method is the term $-\log P(B|T)$, which allows us to consider B rigorously. We calculate the code length $L(T)$ in a problem setting where the receiver has D except for the class labels. The sender first sends T, then the class labels of examples in D using T, and B using T.

$$L(T) \equiv -\log P(T) - \log P(D|T) - \log P(B|T) \tag{4}$$

Note that the smaller $L(T)$ is the more interesting T is thus the negative code length $-L(T)$ can be considered as our interestingness measure.

We assume that B and D are independent because B is typically given by the user and not inferred from D. In this case, T_{EMDL} is shown to coincide with the maximum a posteriori hypothesis i.e. $T_{\text{EMDL}} = \arg\min_T(-\log P(T) - \log P(D, B|T)) = \arg\max_T P(T|D, B)$.

3.2 Encoding Method

Here we propose how to calculate (4). A hypothesis T is sent by first sending the number μ of distribution rules in T then the premise $\rho(r_i)$ of each r_i in T. The conclusion of r_i is sent as the class labels in $D(T)$ in the message of $-\log P(D|T)$. μ is sent with code length $\Lambda(\mu, 0)$ given by (3). $\rho(r_i)$ is sent by specifying the attributes in the premise and their values. Let $\kappa(a_j)$ and $|x|$ represent the number of possible values of an attribute a_j and the number of atoms in a literal x.

$$-\log P(T) = \Lambda(\mu, 0) + \sum_{i=1}^{\mu} \left[\Theta_0\left(m, |\rho(r_i)|\right) + \sum_{a_j \text{ in } \rho(r_i)} \log \kappa(a_j) \right]$$

The initial hypothesis B is sent by first sending the number ν of distribution rules without conclusions in B then each distribution rule b_i without conclusions using T. The former is sent with code length $\Lambda(\nu, \mu)$. We say that a conjunction x of atoms is more general than a conjunction y of atoms iff. each atom in x is found in y and y has at least one atom which does not exist in x, and denote with $x \sqsupset y$. For instance, $a1 = v1, a3 = v3 \sqsupset a1 = v1, a2 = v2, a3 = v3$, where $a1, a2, a3$ are attributes and $v1, v2, v3$ are their values. For the latter, we consider four distinctive cases: 1. $\rho(b_i) = \rho(r_i)$, 2. $\rho(b_i) \sqsupset \rho(r_i)$, 3. $\rho(r_i) \sqsupset \rho(b_i)$, and 4. other cases. The sender sends ν flags for indicating the corresponding case of (b_i, r_i) using (2). For the cases 2. and 3., the attributes in $\rho(r_i)$ are used to specify those in $\rho(b_i)$. Below s.t. represents "such that" and $\nu_1(B,T)$, $\nu_2(B,T)$, $\nu_3(B,T)$, $\nu_4(B,T)$ are the respective numbers of the four cases.

$$\begin{aligned}
&-\log P(B|T) \\
&= \Lambda(\nu, \mu) + H(\nu, (\nu_1(B,T), \nu_2(B,T), \nu_3(B,T), \nu_4(B,T)), 4) \\
&\quad + \sum_{i \text{ s.t. } \rho(b_i) = \rho(r_i)} 0 + \sum_{i \text{ s.t. } \rho(b_i) \sqsupset \rho(r_i)} \Theta_0\left(|\rho(r_i)|, |\rho(r_i)| - |\rho(b_i)|\right) \\
&\quad + \sum_{i \text{ s.t. } \rho(r_i) \sqsupset \rho(b_i)} \left[\Theta_0\left(m - |\rho(r_i)|, |\rho(b_i)| - |\rho(r_i)|\right) + \sum_{\substack{a_j \text{ in } \rho(b_i) \\ \text{but not in } \rho(r_i)}} \log \kappa(a_j) \right] \\
&\quad + \sum_{i \text{ for othercases}} \left[\Theta_0\left(m, |\rho(b_i)|\right) + \sum_{a_j \text{ in } \rho(b_i)} \log \kappa(a_j) \right]
\end{aligned}$$

The class labels in D is sent using T: they are decomposed into those covered by each r_i and those in $D \setminus D(T)$. For the former, we use (2) with a small modification to avoid inconveniences[1]. Let $n(T,i), n_j(T,i)$ be the number of examples covered by the i-th rule in T and the number of examples of class j covered by the i-th rule in T, respectively. Let $j_{\mathrm{NTH}(T,i,d)}$ be the d-th most

[1] For instance, $H(8,(4,2,1,1),4) \neq H(8,(1,1,2,4),4)$ and $H(8,(3,3,1,1),4) < H(8,(1,1,2,4),4)$.

numerous class for its number $n_{j_{\mathrm{NTH}(T,i,d)}}(T,i)$ of examples covered by the i-th rule in T so $n_{j_{\mathrm{NTH}(T,i,1)}}(T,i) \geq n_{j_{\mathrm{NTH}(T,i,2)}}(T,i) \geq \cdots \geq n_{j_{\mathrm{NTH}(T,i,M)}}(T,i)$. We assume that the message which specifies the new order of the class labels $c_{j_{\mathrm{NTH}(T,i,1)}}, c_{j_{\mathrm{NTH}(T,i,2)}}, \ldots, c_{j_{\mathrm{NTH}(T,i,M)}}$ has a fixed size and omit counting its code length for simplicity. For the latter, we assign the code length $-\log M$, which is the longest code length for an event with M possible states, to each class label. This assignment represents the indifference of a partial decision list to its uncovered examples. It helps us avoid obtaining a counterintuitive hypothesis of which rules try to "get rid of" examples to have a $D \setminus D(T)$ which is nearly homogeneous with the majority class. We omit the reason due to lack of space.

$$-\log P(D|T) = \sum_{i=1}^{\mu} H(n(T,i), (n_{j_{\mathrm{NTH}(T,i,1)}}(T,i), \ldots, n_{j_{\mathrm{NTH}(T,i,M)}}(T,i)), M) + \sum_{e \notin D(T)} \log M$$

3.3 Desirable Properties

Studying (4) for two similar hypotheses T and T' reveals that (4) exhibits attractive properties. This fact is important because it differentiates (4) from many empirical interestingness measures which are designed to exhibit attractive properties. Due to space constraint, we just show the following without proof.

Theorem 1. *Let $\mu(T_0)$ be the number of distribution rules in a hypothesis T_0. Let two distinct hypotheses T and T' satisfy*

$$-\log P(T) - \log P(B|T) = -\log P(T') - \log P(B|T')$$
$$\mu(T) = \mu(T').$$

If T is more accurate than T' and covers the same number of examples for each rule, i.e.

$$\forall i \; n_{j_{\mathrm{NTH}(T,i,1)}}(T,i) > n_{j_{\mathrm{NTH}(T',i,1)}}(T',i)$$
$$\forall i \forall d \neq 1 \; n_{j_{\mathrm{NTH}(T,i,d)}}(T,i) \leq n_{j_{\mathrm{NTH}(T',i,d)}}(T',i)$$
$$\forall i \; n(T,i) = n(T',i)$$

then T is judged better with our interestingness measure i.e. $L(T) < L(T')$.

3.4 Practical Heuristic Search

Since an exhaustive search for all possible partial decision lists is prohibitive due to its time-inefficiency, CLARDEM applies three heuristic search methods then outputs the partial decision list with the minimum code length. The first two methods are hill climbing from B and $\emptyset$ where a step is an addition/deletion of a rule/atom, where an added rule has a single atom in its premise.

Separate-and-conquer is frequently used for learning a rule-based classifier (e.g. [7]). Here we use a modified version which never returns a hypothesis with a longer code length. It is a double-loop algorithm which searches rules with sequential covering in its outer loop. It searches conjunctions of atoms as premises of the rule with greedy search which checks up to conjunctions of m atoms. Below we show its pseudo-code, where $r_\mu(T'')$ represents the μ-th rule of T''.

```
algorithm Separate-and-conquer
  T = ∅, min = ∞, μ = 1, T' = T
  do // outer loop
    SacInnerLoop(μ, min, T, T', f), μ = μ + 1
  while(f == TRUE) // outer loop
  output T
procedure SacInnerLoop(μ, min, T, T', f)
  f = FALSE, T'' = T', ρ(r_μ(T'')) = TRUE
  for π = 1,...,m // decide the π-th atom in the μ-th rule
    min' = ∞
    foreach attribute a_i
      If a_i does not exist in ρ(r_μ(T''))
        foreach value v_ij of a_i
          ρ(r_μ(T'')) = ρ(r_μ(T'')) ∧ (a_i = v_ij)
          If L(T'') < min' // update the best hypothesis T' with μ rules
            min' = L(T''), T' = T''
          If L(T'') < min // update the best hypothesis T
            min = L(T''), T = T'', f = TRUE
          Delete ∧(a_i = v_ij) from ρ(r_μ(T''))
    T'' = T'
```

In the hill climbing method from B, an addition of a rule, which has a single atom in its premise, at each step takes $O(mn\kappa_{\mathrm{MAX}})$, where κ_{MAX} represents the maximum number of values that an attribute can take. A deletion of a rule at each step takes $O(n\mu_{\mathrm{MAX}})$, where μ_{MAX} represents the maximum number of rules in a hypothesis during the search. An addition of an atom at each step takes $O(mn\kappa_{\mathrm{MAX}}\mu_{\mathrm{MAX}})$. A deletion of an atom at each step takes $O(n\pi_{\mathrm{MAX}}\mu_{\mathrm{MAX}})$, where π_{MAX} represents the maximum number of atoms in a premise during the search. We assume that the number of search steps is $O(|\mu - \nu|)$, $O(\mu) = O(\nu) = O(\mu_{\mathrm{MAX}})$, and $O(\pi_{\mathrm{MAX}}) = O(1)$. Thus the time complexity is given by $O(mn\kappa_{\mathrm{MAX}}\mu^2)$. The same result holds even if the starting point is $\emptyset$. For our Separate-and-conquer, the time complexity is given by $O(m^2 n\kappa_{\mathrm{MAX}}\mu^2)$.

4 Experiments

4.1 Application to Benchmark Data Sets

We use for comparison MDL, which a method based on the MDLP. It employs $L'(T) \equiv -\log P(T) - \log P(D|T)$ as its coding length for T and is equivalent

Table 1. Characteristics of data sets and their initial hypotheses, where rec., prec., κ_{M}, π_{M} represent recall, precision, κ_{MAX}, and π_{MAX}, respectively

	data set				initial hypothesis					data set				initial hypothesis			
name	n	m	κ_{M}	M	ν	π_{M}	rec.	prec.	name	n	m	κ_{M}	M	ν	π_{M}	rec.	prec.
golf	14	5	3	3	4	2	85.7	100.0	ttt	958	10	3	2	18	5	99.0	100.0
spon.	76	46	12	12	13	4	96.1	97.3	car	1728	7	4	4	87	5	99.5	96.9
p.-op.	90	9	5	3	2	2	26.7	83.3	kr-kp	3196	37	7	2	14	12	89.8	99.7
vote	435	17	3	2	5	2	98.4	97.4	mush.	8124	22	12	2	9	3	100.0	99.8
soyb.	683	36	19	19	36	8	97.2	98.0	nurse.	12960	9	5	5	352	7	95.6	99.5

Table 2. Performance on benchmark data sets, where the best method represents the heuristic search method that returned the best result. HC1, HC2, S, and # nodes represent hill climbing from a null hypothesis, hill climbing from the initial hypothesis, the separate-and-conquer method, and the number of the searched nodes, respectively

	Discovered hypothesis					Search		
name	method	μ	π_{MAX}	recall	precision	best method	# nodes	time
golf	CLARDEM	4	2	85.7	100.0	HC2 S	180	0.01s
	MDL	0	-	0.0	-	HC1 HC2	186	0.01s
	e-Jmeasure	5	2	100.0	100.0	HC1 HC2 S	508	0.10s
sponge	CLARDEM	13	4	96.1	97.3	HC2	16626	1.38s
	MDL	5	3	100.0	69.7	HC2	37803	5.73s
	e-Jmeasure	16	4	100.0	97.4	HC2	49008	5.01s
post-operative	CLARDEM	3	2	97.8	73.9	HC2	705	0.03s
	MDL	1	1	92.2	72.3	HC1 HC2 S	629	0.03s
	e-Jmeasure	21	3	100.0	86.7	HC2	15680	0.71s
vote	CLARDEM	5	2	98.4	97.4	HC2	1776	0.09s
	MDL	2	1	88.7	98.7	HC1 HC2 S	2463	0.14s
	e-Jmeasure	9	3	100.0	98.4	HC1	13774	0.79s
soybean	CLARDEM	37	8	100.0	96.5	HC2	33156	6.26s
	MDL	18	5	100.0	84.2	HC2	160838	36.71s
	e-Jmeasure	37	9	100.0	96.2	HC2	75551	10.38s
tic-tac-toe	CLARDEM	18	5	99.0	100.0	HC2	4046	0.57s
	MDL	11	3	100.0	100.0	HC2	8200	1.00s
	e-Jmeasure	19	5	100.0	100.0	HC2	8550	1.01s
car	CLARDEM	87	5	99.8	97.9	HC2	7766	4.83s
	MDL	25	4	100.0	96.5	HC2	61026	29.14s
	e-Jmeasure	87	5	100.0	97.6	HC2	24487	11.92s
kr-vs-kp	CLARDEM	15	12	100.0	99.7	HC2	61412	26.92s
	MDL	10	12	100.0	99.7	HC2	86656	46.88s
	e-Jmeasure	16	12	100.0	99.9	HC2	193132	31.90s
mushroom	CLARDEM	9	3	100.0	99.8	HC2	20538	23.05s
	MDL	9	3	100.0	99.8	HC2	18964	21.01s
	e-Jmeasure	9	3	100.0	99.8	HC2	19083	4.42s
nursery	CLARDEM	354	7	100.0	99.5	HC2	70822	14m
	MDL	113	7	100.0	98.2	HC2	1200555	191m
	e-Jmeasure	351	7	100.0	99.7	HC2	395409	73m

Table 3. Discovered hypotheses from the vote data set

CLARDEM	e-Jmeasure
`physician = n`	`physician = y, synfuels = n,`
`-> (245, 2)/247`	`immigration = y -> (0, 76)/76`
`missile = y, synfuels = y`	`physician = n -> (245, 2)/247`
`-> (6, 1)/7`	`education = n, salvador = n -> (5, 1)/6`
`adoption = y, synfuels = y`	`missile = ? -> (0, 3)/3`
`-> (6, 2)/8`	`education = ?, adoption = ? -> (2, 0)/2`
`physician = y, synfuels = n`	`water = ? -> (0, 8)/8`
`-> (3, 135)/138`	`adoption = n -> (4, 70)/74`
`physician = y, missile = n`	`satellite = n -> (11, 0)/11`
`-> (3, 25)/28`	`adoption = y -> (0, 8)/8`

to our method for other points. We also use e-Jmeasure, which is an extension of the J-measure [12] to evaluate the goodness of T with the amount $\Gamma(T)$ of information compressed by T, where $\Gamma(T) \equiv \sum_{i=1}^{\mu} \sum_{j=1}^{M} n_j(T,i)\Big(-\log P(c_j) + \log \frac{n_j(T,i)}{n(T,i)}\Big)$. We exclude ad-hoc methods such as those based on frequent itemsets because such a method requires parameters such as support and confidence thresholds, and lacks of a theoretical background and a clear interpretation.

We first apply the three methods to ten benchmark data sets from [2] to investigate their tendencies except discovery accuracies as there is no ground truth. An initial hypothesis is generated by deleting the default class label of the decision list obtained with C4.5rules [6]. We show the characteristics of the data sets and the initial hypotheses in Table 1.

The results of the experiments and the names of the data sets are shown in Table 2. We see that the number μ of the distribution rules in the output hypothesis often increases in the order of MDL, CLARDEM, and e-Jmeasure. Theses results make sense as MDL has a preference bias for $\emptyset$, CLARDEM for the initial hypothesis, and e-Jmeasure for hypotheses which compress a large amount of information. These reasons explain that recall and precision often improve in this order, though their differences are often small.

In terms of search, we see that the method chosen as best most frequently is the hill climbing from the initial hypothesis (HC2). We attribute the reason to the excellence of C4.5rules [6]. As CLARDEM has a preference bias for the initial hypothesis, HC2 is always chosen as the best method. For computation time, CLARDEM is the fastest among the three methods for most of the cases. This result may be explained by the fact that the discovered hypotheses are often most similar to the initial hypotheses. As MDL has a preference bias for an empty hypothesis, the similarity is often the least hence it was the slowest. The number of the searched nodes gives a rough estimate of the computation time for the same data sets (e.g. nursery) thus it will be used as an index.

Due to lack of space we just show examples of the discovered hypotheses from the vote data set. MDL discovered a simple one with two rules, where class 1 and class 2 correspond to democrat and republican, respectively.

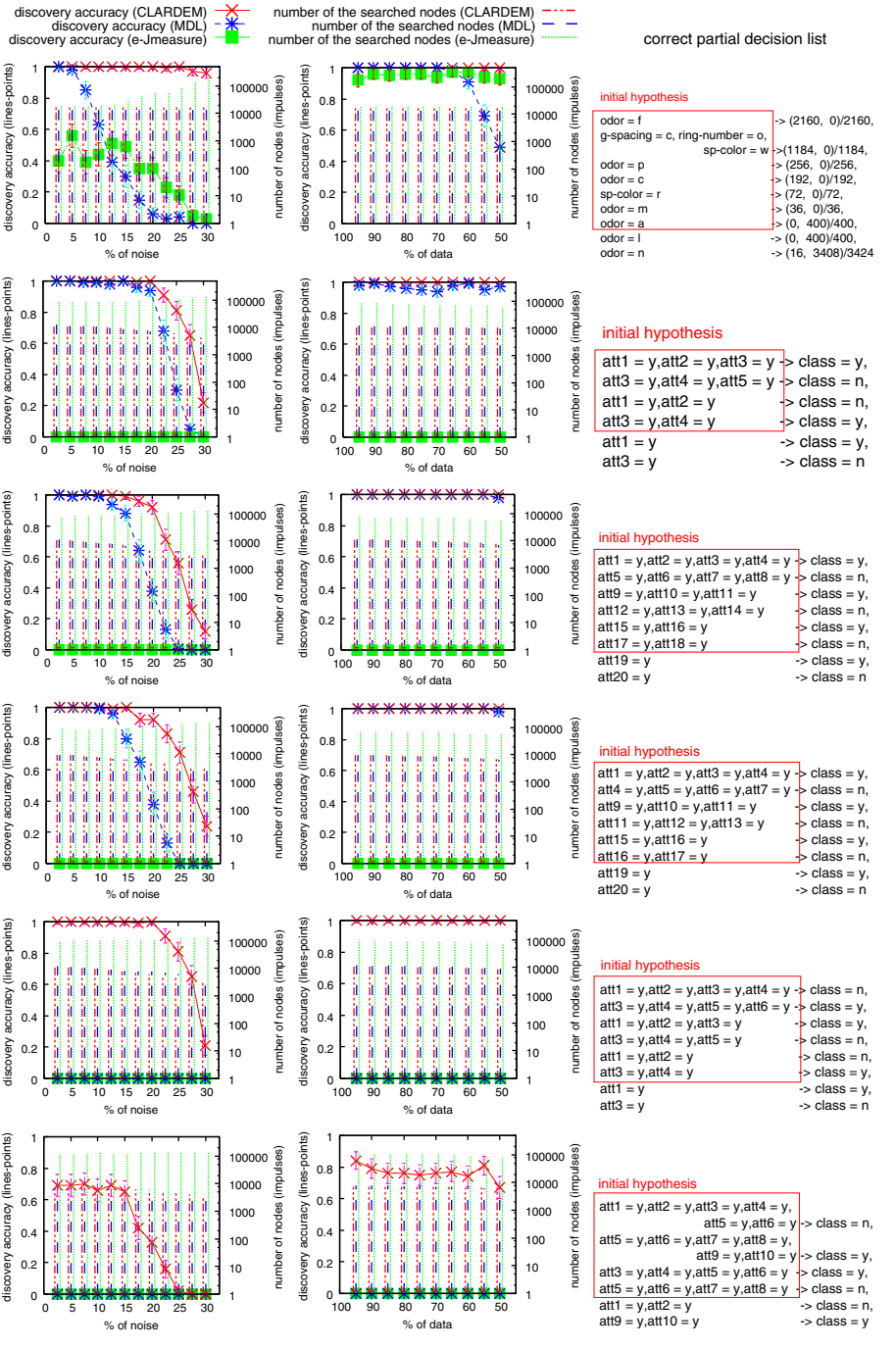

Fig. 1. Results of experiments for robustness with the mushroom data set and five artificial data sets, where class = y and class = n represent (1,0) and (0, 1), respectively

```
physician = n -> (245,  2)/247, synfuels = n -> (3,  136)/139
```

CLARDEM discovered the initial hypothesis and e-Jmeasure a complex one, which are shown in Table 3. Their preference biases explain these results.

4.2 Robustness of the Three Methods

We report the robustness of the methods to noisy data sets and incorrect initial hypotheses, where each result is an average performance on 100 data sets. For Mushroom, $T_{\rm true}$ is assumed to be the hypothesis generated by C4.5rules minus the default class label. Artificial data sets of $n = 1000$, $M = \kappa = 2$, $m = 32$ with 5, 10, ..., 30 % of random noise in the class labels are generated using hand-coded concepts. We have also generated small data sets with $n = 950, 900, \ldots, 500$ without noise. Class labels of uncovered examples are set randomly.

We consider problems of completing approximate initial hypotheses, which fits the nature of the partial decision list. The results of the experiment with correct concepts and the incorrect initial hypotheses are shown in Figure 1, where we also show $\pm$ 1.5*(standard deviations) for discovery accuracies. We see that CLARDEM is almost always the best method due to its capability of exploiting the initial hypothesis even if it is approximate. MDL is often the second method while e-Jmeasure is almost always the worst. We think e-Jmeasure always shows discovery accuracy 0 % for artificial data sets because it tries to compress the "random" parts not covered by $T_{\rm true}$. Anyway CLARDEM is also the best method for mushroom, which has no random part. CLARDEM shows high discovery accuracies even if the initial hypothesis is complex and contains strongly related rules. The numbers of the searched nodes show that CLARDEM and MDL are often one order of magnitude faster than e-Jmeasure.

5 Conclusions

Compression and learning are known to be highly related with each other [5]. The MDLP [5,9] is considered to be among the most successful works along this philosophy due to its performance and theoretical foundation. This paper has presented the first attempt to apply the MDLP and hence the philosophy of data compression to the discovery problem for a group of classification rules.

There are many evidences that the MDLP for classification is robust against noise [8,15]. Our method inherits this nice property and in addition can borrow strength from an initial hypothesis, which are shown through extensive experiments. Our method is adequate for discovering groups of rules even from a small amount of noisy data and an approximate initial hypothesis.

Acknowledgments

This work was partially supported by the grant-in-aid for scientific research on fundamental research (B) 18300047 from the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

1. Baram, Y.: Partial Classification: The Benefit of Deferred Decision. IEEE Trans. Pattern Analysis and Machine Intelligence 20(8), 769–776 (1998)
2. Blake, C., Merz, C.J., Keogh, E.: UCI Repository of Machine Learning Databases, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
3. Jaroszewicz, S., Simovici, D.A.: Interestingness of Frequent Itemsets Using Bayesian Networks as Background Knowledge. In: Proc. Tenth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD), pp. 178–186 (2004)
4. Padmanabhan, B., Tuzhilin, A.: Small is Beautiful: Discovering the Minimal Set of Unexpected Patterns. In: Proc. KDD, pp. 54–63 (2000)
5. Grünwald, P.D.: The Minimum Description Length Principle. MIT Press, Cambridge (2007)
6. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
7. Quinlan, J.R.: Learning Logical Definitions from Relations. Machine Learning 5(3), 239–266 (1990)
8. Quinlan, J.R., Rivest, R.L.: Inferring Decision Trees Using the Minimum Description Length Principle. Information and Computation 80(3), 227–248 (1989)
9. Rissanen, J.: Stochastic Complexity in Statistical Inquiry. World Scientific, Singapore (1989)
10. Shannon, C.: A Mathematical Theory of Communication. Bell System Technical Journal 27, 379–423, 623–656 (1948)
11. Siebes, A., Vreeken, J., van Leeuwen, M.: Item Sets that Compress. In: 2006 SIAM Conference on Data Mining (SDM), pp. 393–404 (2006)
12. Smyth, P., Goodman, R.M.: An Information Theoretic Approach to Rule Induction from Databases. IEEE TKDE 4(4), 301–316 (1992)
13. Tan, P.-N., Kumar, V., Srivastava, J.: Selecting the Right Interestingness Measure for Association Patterns. In: Proc. KDD, pp. 32–41 (2002)
14. Tangkitvanich, S., Shimura, M.: Learning from an Approximate Theory and Noisy Examples. In: Proc. AAAI, pp. 466–471 (1993)
15. Wallace, C.S., Patrick, J.D.: Coding Decision Trees. Machine Learning 11(1), 7–22 (1993)