A Rough Sets Based Approach to Feature Selection

M. Zhang J. T. Yao
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
Email: [zhang2mi, jtyao]@cs.uregina.ca

Abstract— **Feature selection techniques aim at reducing the number of unnecessary features in classification rules. The features are measured by their necessity in heuristic feature selection techniques. Rough set theory has been used to define the necessity of features in literature. We propose a new rough set based feature selection approach called Parameterized Average Support Heuristic (PASH). The PASH considers the overall quality of the potential set of rules. It selects features causing high average support of rules over all decision classes. In addition, the PASH arms with parameters that are used to adjust the level of approximation.**

I. Introduction

Classification is a main problem in machine learning. It may be viewed as a supervised learning process. The rules learnt from this process will be used for prediction. Rules normally consist of a classifier and a group of attributes or features. The features will be used to classify unseen instances into different classes based on the value of the classifier. However, the time required to generate rules will increase dramatically with the number of features [1]. Moreover, if the number of training instances is relatively smaller than the number of features, it will degrade the accuracy of prediction [13]. Feature selection techniques aim at reducing the number of unnecessary, irrelevant, or unimportant features. It is common practice to use a measure to decide the importance and necessity of features.

Rough set theory is an extension of set theory for study of the intelligent systems characterized by insufficient and incomplete information [12]. An undefinable subset is approximately represented by two definable subsets, called lower and upper approximations. Rough set theory is a good candidate for classification applications [2]. Various efforts have been made to improve the efficiency and effectiveness of classification with rough sets [5], [15].

The concepts in rough set theory are used to define the necessity of features. The measures of necessity are calculated by the functions of lower and upper approximation. These measures are employed as heuristics to guide the feature selection process. There are at least two types of heuristics, namely significance oriented method and support oriented method, that have appeared in literature. The heuristic in [5] favors significant features, i.e., features causing the faster increase of the positive region. Zhong's heuristic [15] considers the positive region as well as the support of rules.

This paper proposes a new heuristic function called Parameterized Average Support Heuristic (PASH) based on parameterized lower approximation definition in rough sets. The main advantage of PASH are:

- It considers the overall quality of the set of potential rules. In other words, it takes into account the average support of rules for every decision class. As a result, PASH produces a set of rules with balanced support distribution over all decision classes.
- It considers the predictive instances that are excluded by the existing methods. Predictive instances are instances that may produce predictive rules which hold true with a high probability but are not necessarily always true.

The organization of this paper is as follows: Section II introduces fundamentals of rough set theory. Section III gives background information of feature selection. Section IV analyzes the limitation of existing methods and proposes a new heuristic function, PASH. An demonstrative example showing the advantages of the new heuristic is given in Section V. The paper ends with a conclusion section.

II. Fundamentals of Rough Set Theory

In rough set theory, an information table is defined as a tuple $T = (U, A)$ where U and A are two finite, non-empty sets, U the universe of primitive objects and A the set of attributes. Each attribute or feature $a \in A$ is associated with a set V_a of its value, called the domain of a. We may partition the attribute set A into two subsets C and D, called condition and decision attributes, respectively.

Let $P \subset A$ be a subset of attributes. The indiscernibility relation, denoted by $IND(P)$, is an equivalence relation defined as:

$$IND(P) = \{(x, y) \in U \times U : \forall a \in P, a(x) = a(y)\},$$

where $a(x)$ denotes the value of feature a of object x. If $(x, y) \in IND(P)$, x and y are said to be indiscernible with respect to P.

The family of all equivalence classes of $IND(P)$ (Partition of U determined by P) is denoted by $U/IND(P)$. Each element in $U/IND(P)$ is a set of indiscernible objects with respect to P. Equivalence classes $U/IND(C)$ and $U/IND(D)$ are called condition and decision classes.

For any concept $X \subseteq U$ and attribute subset $R \subseteq A$, X could be approximated by the R-lower approximation and R-upper approximation using the knowledge of R. The lower approximation of X is the set of objects of U that are surely in X, defined as:

$$R_*(X) = \bigcup\{E \in U/IND(R) : E \subseteq X\}.$$

The upper approximation of X is the set of objects of U that are possibly in X, defined as :

$$R^*(X) = \bigcup\{E \in U/IND(R) : E \bigcap X \neq \phi\}.$$

The boundary region is defined as:

$$BND_R(X) = R^*(X) - R_*(X).$$

If the boundary region is empty, that is, $R_*(X) = R^*(X)$, concept X is said to be R-definable. Otherwise X is a rough set with respect to R.

The positive region of decision classes $U/IND(D)$ with respect to condition attributes C is denoted by $POS_c(D) = \bigcup R_*(X)$. It is a set of objects of U that can be classified with certainty to classes $U/IND(D)$ employing attributes of C. A subset $R \subseteq C$ is said to be a D-reduct of C if $POS_R(D) = POS_C(D)$ and there is no $R' \subset R$ such that $POS'_R(D) = POS_C(D)$. In other words, a reduct is the minimal set of attributes preserving the positive region. There may exist many reducts in an information table.

III. Feature Selection

Feature selection and feature extraction are two kinds of methods of dimensionality reduction for classification [7]. Feature extraction creates new features by irreversibly transforming the original features such that the created features contain most useful information for the target concept. In contrast, feature selection only removes the features that are unnecessary or unimportant to the target concept and the remaining features are kept intact. The process of feature extraction is more complicated. It is difficult to compare the effectiveness of the two methods as they are employed under different circumstances.

Features selection is a process to find the optimal subset of features that satisfy certain criteria. In this paper, we consider two parameters: the size of the selected feature subset, and the accuracy of the classifier induced using only the selected features. We have to define an evaluation measure that is able to reflect both of the parameters. In this context, feature selection problem can be viewed as a search problem. The optimal feature subset is the one that maximizes the value of evaluation measure.

A. Feature Selection Methods

The most intuitive method for feature selection is to enumerate all the candidate subsets and apply the evaluation measure to them. Unfortunately, the exhaustive search is infeasible under most circumstances as there are 2^n subsets for a feature set of size n. The exhaustive search could only be used in domain where n is relatively small. Large n will make the search intractable in many real world applications.

An alternative way is to use a random search method where the candidate feature subset is generated randomly [11]. Each time, the evaluation measure is applied to the generated feature subset to check whether it satisfies certain criteria. This process repeats until one subset that satisfies the given criteria is found. The process will also end when a predefined time period has elapsed or a predefined number of subsets have been tested.

The third and most commonly used method is called the heuristic search [9], [8], where a heuristic function is employed to guide the search. The search is performed towards the direction that maximizes the value of a heuristic function.

The exhaustive search is infeasible due to its high time complexity. The random and heuristic search reduce computational complexity by compromising performance. They are not complete search under most circumstances. In other words, they do not guarantee to produce an optimal result. Heuristic search is an important search method used by the feature selection community.

B. Characteristic of Features

The aim of feature selection is to remove unnecessary features to the target concept. Unnecessary features can be classified into irrelevant features and redundant features [3]. Irrelevant features are those that do not affect the target concept in any way. Redundant features do not add anything new to the target concept. Hall [4] argued that a good feature subset is one that contains features highly correlated with the class, yet uncorrelated with each other. If two features are functional dependent, one of them could be removed without the loss of predication accuracy.

A simple heuristic is to define a measure that evaluates the necessity of a feature. However, it is difficult to define a heuristic function on these qualitative descriptions of irrelevance and redundance. John *et al.* [6] defined strong relevance and weak relevance of a feature in terms of the probability of the occurrence of the target concept given this feature. Strong relevant features are indispensable in the sense that it cannot be removed without loss of prediction accuracy. Weak relevant features can sometimes contribute to prediction accuracy.

Strong relevance and weak relevance provide a good foundation upon which we can define the heuristic function. The set of strong relevant features is equivalent to relative CORE in the rough set theory. The relative reduct is a combination of all strong relevant features and some weak relevant features. In rough sets theory, a subset $R \subseteq C$ is said to be a D-reduct of C if $POS_R(D) = POS_C(D)$ and there is no $R' \subset R$ such that $POS'_R(D) = POS_C(D)$. In other words, reduct is the minimal set of attributes preserving the positive region. There may exist many reducts in a information table. The CORE is the set of attributes that are contained by all reducts, defined as: $CORE_D(C) = \bigcap RED_D(C)$ where $RED_D(C)$ is the D-reduct of C. In other words, the CORE is the set of attributes that cannot be removed without changing the positive region. This means that all attributes present in the CORE are indispensable.

C. Basic Issues of Heuristic Feature Selection

The feature selection process is a search process where the whole search space covers all 2^n subsets of the n features, and with each state specifying a candidate subset. A partial order could be imposed on this search space, making each child having exactly one more feature than its parents. The structure of this space determines the basic issues of the heuristic feature selection process [10].

The first step is to decide from which state in the search space that the search starts. We may adopt forward selection that starts with an empty feature set and successively adds features. Another approach is to employ backward elimination that starts with all features and successively removes unnecessary ones. It is also possible to start from somewhere in the middle of the search space, that is, start with a subset that contains some indispensable features and search outwards from this point. In rough set based feature selection approaches, the CORE can be used as the starting point.

The second issue is how the search is carried out. The simplest way is the greedy method which traverses the search space without backtrack. At each step, only one feature is added or removed. Once a feature is added, it can not be removed in later steps. Likewise, once a feature is removed, it can not be added. Another method, known as stepwise selection or elimination, allows adding (removing) a feature that was removed (added) in the previous step.

In our rough sets based feature selection, we adopt the forward selection approach since all the features in CORE cannot be removed. We successively add features until the stop criterion is satisfied. We use a measure, or heuristic function, to evaluate alternative feature subsets. The measure decides the next candidate subset. Filter and wrapper are two classes of commonly used measures [6]. The filter method is independent of the induction algorithm that will use the selected features as it relies only on the characteristics of the features. The wrapper method uses the induction algorithm as the evaluation measure. The rough sets based heuristic functions discussed in this paper belong to the filter measure.

The last basic issue of heuristic search is the stop criteria. A stop criterion is used to halt the search process. In the rough sets based method, the size of the positive region could be used as stop criteria. In particular, the algorithm stops when the positive region of the selected features reaches the original positive region, *i.e.*, $POS_R(D) = POS_C(D)$.

IV. Rough Set based feature selection

This section focuses on rough set based heuristic functions. These heuristic functions are used to decide which attribute is relevant to the target concept. The concepts in the rough set theory can manifest the property of strong and weak relevance as defined in [6]. For example, the relative *reduct* is a combination of all strong relevant features and some weak relevant features. The set of strong relevant features is equivalent to relative *CORE*, which includes attributes contained by all reducts. The rough set concepts could be employed to define the heuristic functions as in [5] and [15]. We analyze some existing heuristic functions in this section. A new rough set based heuristic function, which remedies some limitations of previous functions, is proposed.

A. Significance Oriented Methods

The significance of features was used as the heuristic in one of the pioneer research on feature selection with rough sets [5]. Each time the most significant feature from the unselected features is added to generate the next candidate feature subset. Significance of a feature a, denoted as $SIG(a)$, is the increase of dependency between condition attributes and decision attribute as a result of the addition of a. Therefore, the heuristic is to select with higher preference features causing the dependency to increase faster. The dependency between condition attributes and decision attribute is defined as

$$g(R, D) = card(POS_R(D))/card(U),$$

where $card(POS_R(D))$ is the cardinality of the positive region and $card(U)$ the cardinality of the universe. The dependency $g(R, D)$ reflects the importance of R in classifying the objects into the classes of $U/IND(D)$. The formal heuristic function is defined as follows:

$$SIG(a) = g(R + a, D) - g(R, D),$$

where R is the set of currently selected features and D is the decision attribute.

This heuristic function is simple and with low time complexity. However, this method only considers the dependency of the selected features. The other important information is ignored. As the ultimate goal of feature selection is to reduce the number of features used to generate classification rules, we have to consider the quality of the potential rules. The quality of the rules can be evaluated by two parameters: 1) the number of instances covered by the potential rules, that is, the size of consistent instances; and 2) the number of instances covered by each rule, called support of each rule.

Significance oriented methods only consider the first parameter. It attempts to increase faster the size of consistent instances but ignoring the second parameter(the support of individual rules). However, rules with very low support are usually of little use. For example, the patients' identification number may be picked as the only feature needed in medical diagnosis since every patient has a unique identification number [15].

B. Support Oriented Methods

Zhong, *et al.* [15] proposed a heuristic function that considers both parameters. The heuristic selects feature a such that, by adding a to the current set, the size of consistent instances increase faster and the support of the most significant rule is larger than by adding any other features. The most significant rule is the one with the largest support. This function is a product of two factors, defined as follows,

$$\begin{aligned} F(R,a) &= Card(POS_{R+\{a\}}(D)) \times \\ &\quad MAXSize(POS_{R+\{a\}}(D)/IND(R+\{a\})). \end{aligned}$$

The first factor, $Card(POS_{R+\{a\}}(D))$, indicates the size of consistent instances. The second factor, $MAXSize(POS_{R+\{a\}}(D)/IND(R + \{a\}))$, denotes the maximal size out of indiscernibility classes included in the positive region, i.e., the support of the most significant rule. In the remaining part of the paper, we refer to this heuristic as Maximum Support Heuristic.

The limitation of Maximum Support Heuristic is that it selects with high preference features causing the highest support of the most significant rule rather than the highest overall quality of the potential rules. In other words, it only considers a local optimum instead of a global optimum of the potential rules. The training instances may belong to many classes. Maximum Support Heuristic favors one of the classes. It will produce a set of rules with a biased support distribution. Moreover, sometimes Maximum Support Heuristic fails to make a choice between two sets of features when they cause the same size of positive region and support of the most significant rule.

C. Average Support Heuristic

Based on the above discussion, we propose a new heuristic function, called Average Support Heuristic. The Average Support Heuristic considers the overall quality of the potential set of rules rather than the support of the most significant rule. The overall quality of the potential set of rules, denoted by Q, is the average support of the most significant rules for every decision classes. Unlike the Maximum Support Heuristic, Average Support Heuristic considers all the decision classes. It selects with high preference features causing the highest average support of rules over all decision classes.

The overall quality of the potential set of rules $Q(R, a)$ is defined as follows:

$$Q(R,a) = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} S(R,a,d_i), \qquad (1)$$

where

$$S(R,a,d_i) = MAXSize(POS_{R+\{a\}}(D = d_i)/IND(R+\{a\}))$$

is the support of the most significant rule for decision class $\{D = d_i\}$ and D is the decision attribute. The domain of D is $\{d_1, d_2, \ldots, d_n\}$.

Average Support Heuristic function is defined as the product of $Card(POS_{R+\{a\}}(D))$ and $Q(R, a)$:

$$F(R,a) = Card(POS_{R+\{a\}}(D)) \times Q(R,a). \qquad (2)$$

It is important to note that Average Support Heuristic has the same order of magnitude in time complexity as Maximum Support Heuristic. Both of them could be computed by one scan of the decision classes.

TABLE I

PART OF AN INFORMATION TABLE

	Size	S_1	S_2	S_3	D
E_3	40	0	1	2	2
E_{10}	5	2	2	2	1
E_{11}	100	2	2	2	2

D. Parameterized Average Support Heuristic

The heuristic functions discussed above only consider the positive region in the traditional rough sets model. These functions ignore the information provided by inconsistent instances, or the boundary region. However, this information becomes important to the target concept when the number of inconsistent instances increases.

Table I shows part of a information table of the demonstrative example in Section V, where S_1, S_2 and S_3 are condition attributes and D is decision attribute. In the traditional rough set model, E_{10} and E_{11} will never be included in the positive region. The information contained in E_{10} and E_{11} will never be considered in the feature selection process with Average Support Heuristic. However, the potential rule "$S_1 = 2 \bigwedge S_2 = 2 \bigwedge S_3 = 2 \Longrightarrow D = 2$" obtained from E_{11} has support = 100 and it holds true with probability of 95.2%. It is unsafe to say that this rule is less useful than the rule "$S_1 = 0 \bigwedge S_2 = 1 \bigwedge S_3 = 2 \Longrightarrow D = 2$" from E_3 (E_3 belongs to the positive region) with support = 40.

Since the heuristic functions are defined on the positive region which is the union of lower approximations, we may redefine the lower approximation. We need to broaden the concept of lower approximation and make it to include predictive instances that are excluded by the traditional lower approximation. Predictive instances refer to instances that may produce predictive rules, which hold true with high probability but are not necessarily 100%. In this section, we propose a new definition of lower approximation, based on which we improve the Average Support Heuristic to Parameterized Average Support Heuristic (PASH). PASH also uses a product of two factors: $Card(POS_{R+\{a\}}(D)) \times Q(R, a)$, where $Card(POS_{R+\{a\}}(D))$ is cardinality of the positive region and $Q(R, a)$ the overall quality of potential rules. However, they have been modified in the new heuristic function.

Some research on non-traditional lower approximation could be found in literature. Decision-theoretic rough set model [14] and variable precision rough set model [16] are two examples. The new lower approximations are based on the following assumption: class X has prior probability $P(X)$, and two lower and upper limit certainty threshold parameters l and u such that $0 \leq l < P(X) < u \leq 1$. The lower approximation of X is defined as

$$R_*(X) = \bigcup\{E_i \in U/IND(R) : P(X|E_i) > u\},$$

where $P(X|E_i)$ denotes the probability of X given E_i. This definition is broader than the traditional one. However, prior probability of X required by this model is usually unknown in the real world application. Moreover, one pair of (l, u) confines this model to information tables with only a binary-valued decision attribute. What we need is a definition of lower approximation that is applicable to multi-valued decision attribute.

Suppose that we have an information table T, in which the domain of decision attribute D, denoted by V_D, contains n values, such that $V_D = \{d_1, d_2, \ldots, d_n\}$. Assume each value of the decision attribute has equal prior probability, i.e., $P(D = d_1) = P(D = d_2) = \cdots P(D = d_n)$. This assumption is reasonable when the prior probabilities are unknown. In this case, we define the lower approximation of class $\{D = d_i\}$ as follows:

$$\begin{aligned} &R_*(D = d_i) = \bigcup\{E_j \in U/IND(R) : \\ &P(D = d_i|E_j) > P(D \neq d_i|E_j)\}, \end{aligned} \tag{3}$$

where $P(D \neq d_i|E_j) = \sum_{k=1,k\neq i}^{n} P(D = d_k|E_j)$. The lower approximation of class $\{D = d_i\}$ is the set of such objects E_j in U that, given E_j, the probability of $D = d_i$ is greater than the probability of $D \neq d_i$. In other words, E_j is predictive of concept $D = d_i$ from $D \neq d_i$.

Since $P(D \neq d_i|E_j) = 1 - P(D = d_i|E_j)$, we can rewrite (3) to (4):

$$\begin{aligned} &R_*(D = d_i) = \\ &\bigcup\{E_j \in U/IND(R) : P(D = d_i|E_j) > 0.5\}, \end{aligned} \tag{4}$$

where $P(D = d_i|E_j)$ could be estimated by taking the ratio $Card(D = d_i \bigcap E_j)/Card(E_j)$.

When the decision attribute has few number of values, in the extreme case, the decision attribute is binary, that is, $|V_D| = 2$, (3) may be too broad and degrade the performance. We can introduce a parameter $k(k \geq 1)$ to (3) as follows:

$$\begin{aligned} &R_*(D = d_i) = \bigcup\{E_j \in U/IND(R) : \\ &P(D = d_i|E_j) > k \times P(D \neq d_i|E_j)\}. \end{aligned} \tag{5}$$

Equation (5) reflects that, given E_j, the concept $D = d_i$ is k times more probable than the concept $D \neq d_i$.

By replacing $P(D \neq d_i|E_j)$ with $1 - P(D = d_i|E_j)$, (5) becomes

$$\begin{aligned} &R_*(D = d_i) = \\ &\bigcup\{E_j \in U/IND(R) : P(D = d_i|E_j) > \tfrac{k}{k+1}\}. \end{aligned} \tag{6}$$

As $k \geq 1 \Longrightarrow \frac{k}{k+1} \geq 0.5$, we can simplify (6) as:

$$\begin{aligned} &R_*(D = d_i) = \bigcup\{E_j \in U/IND(R) : \\ &P(D = d_i|E_j) > t(t \geq 0.5)\}. \end{aligned} \tag{7}$$

Clearly, Equation (4) is a special case of (7). Equation (7) guarantees that each object $E \in U$ is contained in at most one lower approximation, that is,

$$R_*(D = d_i) \bigcap R_*(D = d_j) = \phi, (i \neq j).$$

TABLE II

AN EXAMPLE INFORMATION TABLE

	Size	S_1	S_2	S_3	D
E_1	150	2	0	1	1
E_2	150	0	1	0	2
E_3	40	0	1	2	2
E_4	50	2	1	0	1
E_5	50	0	1	3	1
E_6	170	0	0	2	1
E_7	300	0	2	1	1
E_8	10	1	1	0	2
E_9	250	3	1	1	1
E_{10}	5	2	2	2	1
E_{11}	100	2	2	2	2

In the case that the prior probabilities of decision classes are known, (7) is too simple to be effective. Assume that the information table obtained from the training data can reflect the distribution of decision classes. The prior probability of class $(D = d_i)$ could be estimated by $P(D = d_i) = \frac{Card(D=d_1)}{Card(U)}$. We can modify (7) to (8):

$$\begin{aligned} &R_*(D = d_i) = \bigcup\{E_j \in U/IND(R) : \\ &\tfrac{P(D=d_i|E_j)}{P(D=d_i)} = MAX\{\tfrac{P(D=d_k|E_j)}{P(D=d_k)}, 1 \leq k \leq n\} \\ &\text{and } P(D = d_i|E_j) > t(t \geq 0.5)\}. \end{aligned} \tag{8}$$

Equation (8) ensures that the lower approximation of class $\{D = d_i\}$ contains such objects $E_j \in U$ that, given E_j, the probability of class $\{D = d_i\}$ increases faster than any other class+-. Equation (8) also guarantees $R_*(D = d_i) \bigcap R_*(D = d_j) = \phi, (i \neq j)$. Equation (7) is a special case of (8).

The newly proposed Average Support Heuristic has been improved to PASH by defining a new lower approximation. There are two cases to be considered when using PASH:

- When the prior probabilities of decision classes are unknown, we assume they have equal prior probability and use (7).
- When the prior probabilities of decision classes are known, we use (8).

Average Support Heuristic and Parameterized Average Support Heuristic can be viewed as extensions to Maximum Support Heuristic.

V. DEMONSTRATIVE EXAMPLE

In this section, we use a demonstrative example in Table II to show the advantages of PASH. Suppose that Table II is an information table for medical diagnosis, where S_1, S_2, S_3 are symptoms and D disease prediction. $\{D = 1\}$ and $\{D = 2\}$ predict disease one and disease two, respectively.

We first apply Maximum Support Heuristic to the information table. In the first step, $R = \phi$, we have $F(R, S_3) > F(R, S_1)$ and $F(R, S_3) > F(R, S_2)$. Hence, symptom S_3 is the first feature to be selected. In the second step, $R = S_3$, we have $F(R, S_1) = F(R, S_2)$ which means it fails to make choice between symptoms $\{S_1, S_3\}$ and $\{S_2, S_3\}$. Symptom S_1 and S_2 are regarded as equally important and can be randomly selected. However, we find that $\{S_1, S_3\}$ is more useful than $\{S_2, S_3\}$ for the following reason. Since the information table is used to predict two diseases $\{D = 1\}$ and $\{D = 2\}$, we have to consider the support of rules for both diseases. Table III and Table IV show the positive region of $\{S_1, S_3\}$ and $\{S_2, S_3\}$, respectively. In Table III, the support of the most significant rule for $\{D = 2\}$ is 150 and the support of the most significant rule for $\{D = 1\}$ is 300. In Table IV, the support of most significant rule for $\{D = 1\}$ is also 300, but the support of the most significant rule for $\{D = 2\}$ is only 40. In other words, the classification rule for decision class $\{D = 2\}$ obtained from Table IV does not have enough supporting instances. Thus, $\{S_1, S_3\}$ is preferred over $\{S_2, S_3\}$. However, Maximum Support Heuristic cannot tell the difference between $\{S_1, S_3\}$ and $\{S_2, S_3\}$. Moreover, Maximum Support Heuristic is based on traditional rough set model and does not consider E_{11}.

TABLE III

POSITIVE REGION OF $\{S_1,S_3\}$

	Size	S_1	S_3	D
E_1	150	2	1	1
E_2	150	0	0	2
E_4	50	2	0	1
E_5	50	0	3	1
E_7	300	0	1	1
E_8	10	1	0	2
E_9	250	3	1	1

TABLE IV

POSITIVE REGION OF $\{ S_2, S_3\}$

	Size	S_2	S_3	D
E_1	150	0	1	1
E_3	40	1	2	2
E_5	50	1	3	1
E_6	170	0	2	1
E_7	300	2	1	1
E_9	250	1	1	1

We next apply PASH with $t = 90\%$ which ensures that E_{11} is included in positive region. In both situations (use (7) or (8)), we have

$$Q(S_3, S_1) = (|E_7| + |E_2|)/2 \quad = (300 + 150)/2 = 225,$$

$$\text{and } Q(S_3, S_2) = (|E_7| + |E_{11}|)/2 = (300 + 100)/2 = 200,$$

$$\text{so that,} \qquad Q(S_3, S_1) > Q(S_3, S_2).$$

In other words, the overall quality of the potential set of rules obtained from $\{S_1, S_3\}$ is better than those from $\{S_2, S_3\}$. By using (2),

$$F(S_3, S_1) = Card(POS_{\{S_3,S_1\}}(d)) \times Q(S_3, S_1) = 1060 \times 225,$$

$$F(S_3, S_2) = Card(POS_{\{S_3,S_2\}}(d)) \times Q(S_3, S_2) = 1060 \times 200,$$

$$\text{so that,} \qquad F(S_3, S_1) > F(S_3, S_2).$$

PASH chooses symptoms $\{S_3, S_1\}$ since $\{S_3, S_1\}$ produces higher average support of the rules for both $\{D = 1\}$ and $\{D = 2\}$.

VI. Conclusion

This paper proposes two new rough set based feature selection heuristics, Average Support Heuristic and Parameterized Average Support Heuristic (PASH). Unlike the existing methods, PASH is based on a special lower approximation which is defined to include all predictive instances. Predictive instances may produce predictive rules, which hold true with high probability (higher than a user specified threshold) but are not necessarily one hundred percent true. However, the traditional model excludes the predictive instances that may produce not-100%-true rules.

The main advantage of PASH is that 1) it considers the overall quality of the potential rules, thus produce a set of rules with balanced support distribution over all decision classes; 2) it arms with a parameter to adjust the level of approximation and keeps the predictive rules that are ignored by the existing methods.

References

[1] R. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 1961.

[2] J. S. Deogun, V. V. Raghavan, and H. Sever, "Rough set based classification methods and extended decision tables," *Proc. of The Int. Workshop on Rough Sets and Soft Computing*, pp302-309, 1994.

[3] M. Dash, H. Liu, "Feature Selection for Classification,". *Intelligence Data Analysis*, 1, 131-156, 1997.

[4] M. A. Hall, *Correlation-based Feature Selection for Machine Learning*, PhD thesis, Waikato University, New Zealand, 1999.

[5] X. Hu, *Knowledge discovery in databases: an attribute-oriented rough set approach*, PhD thesis, University of Regina, Canada, 1995.

[6] G. H. John, R. Kohavi and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," *Proceedings of the Eleventh International Conference on Machine Learning*, pp121-129, 1994.

[7] J. Kittler, "Feature selection and extraction," In Young and Fu (Eds.), *Handbook of pattern recognition and image processing,* pp203-217, New York: Academic Press, 1986.

[8] I. Kononenko, "Estimating attributes: analysis and extension of relief," *Proceedings of European Conference on Machine Learning,* pp171-182, 1994.

[9] K. Kira, L. Rendell, "A practical approach to feature selection," *Proceedings of the Ninth International Conference on Machine Learning,* pp249–256, 1992

[10] P. Langley, "Selection of Relevant Feature in Machine Learning," *proceedings of the AAAI fall symposium on Relevance,* pp140-144, 1994.

[11] H. Liu, R. Setiono, "A Probabilistic Approach to Feature Selection-A Filter Solution," *Proceedings of the 13th International Conference on Machine Learning*, pp319-327, 1996.

[12] Z. Pawlak, *Rough Sets-theoretical Aspects of Reasoning about Data,* Kluwer Academic, Dordrecht, 1991.

[13] G.V. Trunk, "A Problem of Dimensionality: A Simple Example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(3), 306-307, 1979.

[14] Y.Y. Yao and S.K.M. Wong, "A decision theoretic framework for approximating concepts," *International Journal of Man-machine Studies*", 37(6), 793-809, 1992.

[15] N. Zhong, J.Z. Dong and S. Ohsuga, "Using Rough Sets with Heuristics for feature Selection," *Journal of Intelligent Information Systems*, 16, 199-214, 2001.

[16] W. Ziarko, "Variable precision rough set model," *Journal of Computer and System Sciencs*, 46, 39-59, 1993.