# Rough Set Feature Selection Methods for Case-Based Categorization of Text Documents

Kalyan Moy Gupta[1,2], Philip G. Moore[1,2], David W. Aha[1], and Sankar K. Pal[3]

[1] ITT Industries,
2560 Huntington Ave, Alexandria, VA, USA
[2] Naval Research Laboratory,
4555 Overlook Ave., SW, Washington, DC, USA
[3] Indian Statistical Institute,
203 Barrackpore Trunk Road, Kolkata, India
{firstname.lastname}@nrl.navy.mil, sankar@isical.aci.in

**Abstract.** Textual case bases can contain thousands of features in the form of tokens or words, which can inhibit classification performance. Recent developments in rough set theory and its applications to feature selection offer promising approaches for selecting and reducing the number of features. We adapt two rough set feature selection methods for use on n-ary class text categorization problems. We also introduce a new method for selecting features that computes the union of features selected from randomly-partitioned training subsets. Our comparative evaluation of our method with a conventional method on the Reuters-21578 data set shows that it can dramatically decrease training time without compromising classification accuracy. Also, we found that randomized training set partitions dramatically reduce training time.

## 1 Introduction

Textual Case-Based Reasoning (TCBR) is a methodology that retrieves and reuses decisions from stored documents (i.e., *cases*) to solve new problems. Some TCBR methods can be used for supervised learning tasks. For example, they have been used to assign one or more topics to a document (Wiratunga, 2004). A problematic issue in TCBR, and text categorization in general, is the high dimensionality of the feature space. Most current approaches consider the unique terms and phrases that occur in the set of documents as features. These frequently number in the tens of thousands, which is prohibitively high for most learning algorithms. A variety of feature selection techniques can be used to address this issue. Conventional filter feature selection approaches for textual data have predominantly used statistical and entropy approaches such as document frequency, information gain, and mutual information (Yang & Pederson, 1997). Previously, rough set feature selection techniques have been applied to structured data with much success (Pal & Shiu, 2004), and may be a promising alternative for textual data. In this paper, we extend and evaluate two rough set feature selection methods on n-ary class text categorization problems. We also investigate and demonstrate the effectiveness of feature selection methods using randomly partitioned training sets to reduce training time.

S.K. Pal et al. (Eds.): PReMI 2005, LNCS 3776, pp. 792 – 798, 2005.

We organize this paper as follows. In Section 2, we briefly review feature selection techniques that have been applied with TCBR methods. In Section 3, we adapt two rough set feature selection techniques to the n-ary class problem and present a feature selection methodology using randomized training set partitions. We present their evaluation in Section 4 and Section 5 concludes the paper.

## 2 Related Work

Feature selection is imperative for categorization with massive textual databases. A variety of filter metrics (e.g., document frequency, information gain, mutual information) can be effectively used to select features (Yang & Pederson, 1997). These techniques, while effective, have two shortcomings. First, they have high average computational complexity. For example, the information gain measure has a complexity of O(*NVM*) where *N* is the number of documents, *V* is the vocabulary size, and *M* is the number of topics. Second, the user must specify a threshold (i.e., indicate when to stop selecting features).

Rough set methods have recently been used to select features from textual data (Chouchoulas & Chen, 2001; Li *et al.*, 2005), and have the advantage of not requiring a threshold setting. However, because finding optimal feature subsets using rough set methods is computationally intractable (i.e., NP hard), heuristic approaches (e.g., QuickReduct) have been developed. Still, the computational complexity of heuristic techniques is still problematic for massive data. For example, its worst case computational complexity is $O(RVN^2)$ where *R* is the size of reduct. Li *et al*. (2005) address this issue by developing an approximate reduct computation method that has a complexity of O(*VN*).

Chouchoulas and Chen (2001) explored the applicability of QuickReduct to a strict categorization problem (e.g., placing an email in exactly one folder). However, they did not explore its application to n-ary classification problems in large datasets. Li *et al*. (2005) applied their approximate reduct computation method to small sets of n-ary classification problems, but also did not examine performance issues pertaining to massive data sets. Alternative heuristics for rough sets include Johnson's algorithm and the Best Reduct Method (Popova, 2004). In this paper, we adapt QuickReduct and Johnson's algorithm to n-ary categorization problems and examine the impact of randomized partitions on their training time and classification performance.

## 3 N-Ary Classification with Rough Set Feature Selection

Assigning more than one class label to an object is an *n-ary* classification task. For example, assigning multiple keywords or topics to a document is an n-ary classification task. Rough set feature selection techniques can be applied to reduce the dimensionality of datasets used for n-ary classification.

Rough set approaches view a decision system as a table, where attributes are columns and objects are rows (Pawlak, 1991). For a set of conditional attributes **C**, decision attributes **D**, and universe of objects **U**, let *f(x,q)* denote an attribute value, where

$x \in \mathbf{U}$ and $q \in \mathbf{A}$ (where $\mathbf{A}=\mathbf{C} \cup \mathbf{D}$). Then $f(x,q)$ defines an equivalence relation $R_q$ over $\mathbf{U}$ that can be used to partition $\mathbf{U}$ into its disjoint subsets as follows:

$$R_q = \{x: x \in \mathbf{U} \wedge f(x,q) = f(x_0,q) \ \ \forall x_0 \in \mathbf{U}\} \tag{1}$$

A key concept in rough set theory is *indiscernability*. Two objects $x$ and $y$ are indiscernible given a subset of attributes $\mathbf{P}$ iff $f(x,q) = f(y,q)$ $\forall q \in \mathbf{P}$. Let IND($\mathbf{P}$) be an indiscernability relation that partitions $\mathbf{U}$. Then rough set theory approximates traditional sets using a pair of sets called *lower* ($\underline{P}\mathbf{D}$) and *upper* ($P\mathbf{D}$) as follows:

$$\underline{P}\mathbf{D} = \cup\{X: X \in \mathbf{U}/\text{IND(P)}, \ X \subseteq \mathbf{D}\} \tag{2}$$

$$P\mathbf{D} = \cup\{X: X \in \mathbf{U}/\text{IND(P)}, \ X \cap \mathbf{D}\} \tag{3}$$

The *positive region* of $\mathbf{P}$, denoted by $\text{POS}_{\mathbf{P}}(Q)$ where $Q$ is the set of decision attributes, includes all the objects that are contained by the lower approximation. These objects can be classified using the information contained only in the attributes $\mathbf{P}$. Therefore, the information contained by a set of attributes can be measured by the *degree of dependency* as follows:

$$\gamma_{\mathbf{P}}(Q) = ||\text{POS}_{\mathbf{P}}(Q)||/||\mathbf{U}|| \tag{4}$$

where, $||\ ||$ denotes a set's cardinality.

Feature selection involves removing those attributes that have no significant information pertaining to the decision task. In a rough set method, the set of attributes $\mathbf{R}$, called the *reduct*, is a set of conditional attributes such that $\gamma_{\mathbf{R}}(\mathbf{D}) = \gamma_{\mathbf{C}}(\mathbf{D})$ for decision attributes $\mathbf{D}$. Computing the minimal reduct is an NP hard problem. Hence, we discuss our adaptation of heuristic methods below.

### 3.1 QuickReduct

QuickReduct greedily selects attributes by computing the marginal increase in the degree of dependency for every attribute it chooses to add to the reduct (see (Chouchoulas & Shen, 2001) for details). At each iteration, it chooses the attribute with the maximum marginal increase in $\gamma$. It starts out with an empty reduct and terminates when there is no change in the degree of dependency of the reduct.

We adapt QuickReduct for n-ary classification using the following indiscernability relations for conditional and decision attributes. We compute the similarity $\sigma(\mathbf{D})$ of decision attribute values $V$ of two objects $x$ and $y$ follows:

$$\sigma(\mathbf{D}) = ||(V_x \cap V_y)||/ \ ||(V_x \cup V_y)|| \tag{5}$$

When $\sigma(\mathbf{D}) > \tau$, where $\tau$ is a user specified threshold ($0 \leq \tau < 1$), then the two objects are considered to be indiscernible. Two text documents $x$ and $y$ are indiscernible with respect to a given term $t$ (i.e., a potential conditional attribute), if either both the documents contain $t$ or both exclude it. That is, we ignore the frequency of term occurrences in a document for establishing indiscernability.

The computational complexity of QuickReduct is $O(RVN^2)$, where $V$ is the vocabulary size of the text collection, $N$ is the number of documents, $R$ is the size of reduct, and we assume that ($R << V$). For massive data sets, where $N$ and $V$ are in the tens of

thousands, this method can be computationally infeasible. Below we present Johnson's (1974) heuristic, which has a lower computational complexity.

### 3.2 Johnson's Heuristic

Johnson's heuristic operates on the discernibility matrix that describes how two objects $x$ and $y$ differ from each other given the condition attributes (see Figure 1). Each cell $c_{x,y}$ in the matrix is defined as:

$$c_{x,y} = \{\{ a \in \mathbf{C}: a(x) \neq a(y)\} \text{ for } \mathbf{D}(x) \neq \mathbf{D}(y), \text{ and } \phi \text{ otherwise } \} \quad (6)$$

```
JOHNSONSREDUCT(C, D, U)
Input   C-conditional attributes, D-decision attribute,
        U-universe of objects
Output  R, Attribute reduct R ⊆ C
1   R←ϕ, O←U, A←C
2   do
3      M← computeDiscernibilityMatrix(O, A, D)
4      a_h← selectHighestScoringAttribute(M)
5      R ←R ∪ a_h
6      O ← O – entriesContaining(M, a_h)
7      A ← A - a_h
8   until O = ϕ
9   return R
```

**Fig. 1.** Pseudocode for Johnson's heuristic

Given such a matrix M, for each attribute, it counts the number of entries in the matrix. The attribute $a_h$ with the highest number of entries is selected for addition to the reduct **R**. Then all the objects that contain $a_h$ are removed and the procedure is repeated until no objects remain. We adapt Johnson's heuristic for n-ary classification problems as follows. For each attribute, we compute a score equivalent to the highest number of entries by weighting the respective entries with a *dissimilarity factor* (1- $\sigma(\mathbf{D})$), which represents the value differences in the decision attributes of objects $x$ and $y$. If $\sigma(\mathbf{D})= 0$, then it reduces to the standard heuristic.

The *worst case* computational complexity of JOHNSONSREDUCT is $O(RN^2)$, which is independent of the vocabulary size $V$. Therefore, we expect it to be much faster than QuickReduct. However, like QuickReduct, the computational cost still increases as a square of the number of documents in the database. To combat this problem, we introduce the following feature selection method.

### 3.3 Feature Selection with Randomized Partitions

Feature selection with randomized partitions builds on the idea of divide and conquer. By reducing the number of documents that must be processed at one time, it can significantly reduce the training time. It proceeds as follows:

1. Randomly create $m$ partitions of the training set.
2. From each partition, select features using a rough set approach such as QuickReduct or JOHNSONSREDUCT.
3. Define the final feature set as the union of features selected from each partition.

This approach should reduce the training time by a factor $1/m^2$. Next, we describe an evaluation of the rough set approach with randomized partitions.

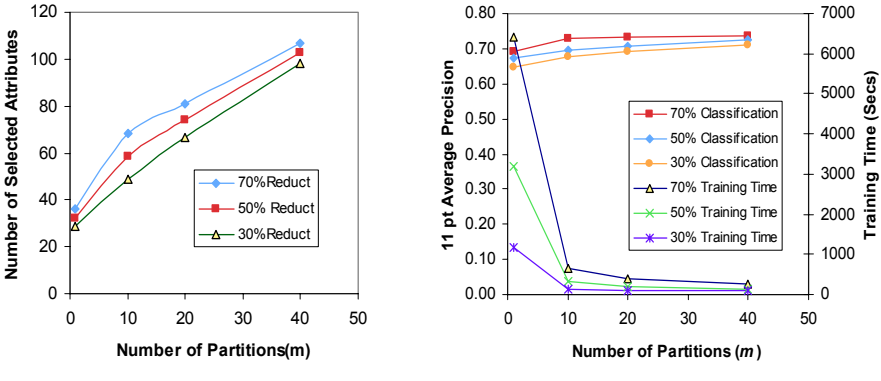
## 4 Evaluation

We designed an experiment to measure the classification accuracy and training time of the rough set feature method. We also examined the effectiveness of randomized partitions for reducing the training time.

We selected the Reuters-21578 data set (Reuters, 2005) for our evaluation. It includes 11,330 news briefs (i.e., *cases*) with one or more topics (the number of topics ranges from 1 to 16, with an average of 1.26 per case). Each brief has an average of 137 words. In addition to the two rough set methods we described in Section 3, we implemented an information gain feature selection method for comparison (see (Yang & Pederson, 1997)). Part-of-speech tagging and morphotactic processing were used to create the features prior to their selection. Finally, we used the k-nearest-neighbor (kNN) classifier, where k was empirically set to 100.

We measured the three algorithms using *11-point average precision*, which is the average precision obtained at recall thresholds of 0%, 20%, … 100%. The system assigns as many topics as needed until a given recall is achieved (Yang & Pederson, 1997). We also measured the *training time* in seconds (we ran our experiments on a Mac Power PC G4, 1.5 GHz). Each algorithm was run multiple times with a varying number *m* of training set partitions (1, 10, 20, and 40) and a varying percentage of cases used for training (30%, 50%, and 70%).

QuickReduct could only be run on a small number of documents due to its long training time (270 seconds for 100 cases, and 15,500 seconds for 200 cases). On this basis, we concluded that it is not suitable for large scale n-ary classification tasks.




(a) (b)

**Fig. 2.** (a) JOHNSONSREDUCT attribute selection performance. (b) JOHNSONSREDUCT classification and training time performance.

Figures 2(a) and 2(b) display the average (over 5 runs) performance of JOHNSONSREDUCT for feature selection, classification accuracy, and training time. It dramatically reduced the number of attributes (e.g., 11,500 to 32 for no partitions and to 107 for 40 partitions). Increasing the number of randomized partitions increased the number of selected attributes and increased the classification accuracy (69.3% for no

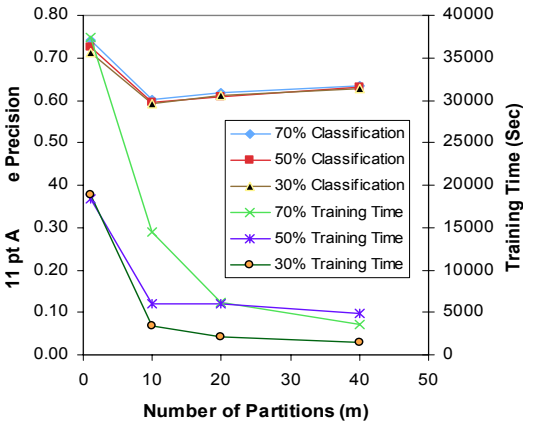



**Fig. 3.** Information Gain performance

partitions to 73.7% for 40 partitions). Increasing the number of partitions also drastically reduced the computation time (6400 seconds for no partitions to 260 seconds for 40). Furthermore, decreasing the percentage of training cases from 70% to 30% of the dataset only marginally reduced the classification accuracy. Thus, JOHNSONSREDUCT is effective for feature selection, and using randomized partitions is effective for reducing training time. For comparison, Figure 3 shows the classification and training time performance of information gain (IG) at the same number of attributes as JOHNSONSREDUCT. Classification accuracy is significantly lower with randomized partitions (e.g., 61% for 10 partitions). This is consistent with our expectations that IG relies on more data for reliable statistics. Further, IG training time (e.g., 37,000 seconds for no partitions) is substantially higher than JOHNSONSREDUCT (6000 seconds). This was expected due to IG's dependency on the vocabulary size.

## 5 Conclusion

We adapted two rough set feature selection techniques for n-ary text categorization problems. We also introduced a method of training with randomized partitions to drastically reduce their training time. Based on our preliminary evaluations, we concluded that the JOHNSONSREDUCT technique is a robust and feasible technique for feature selection for n-ary text categorization problems, while QuickReduct is unsuitable for the same. Although randomized partitions are effective for rough set feature selection, it is not effective for some conventional methods (e.g., information gain). In our future work, we will extend our evaluations to additional data sets.

## Acknowledgement

We thank the Naval Research Laboratory for supporting this research.

## References

Chouchoulas, A., & Shen, Q. (2001). Rough-set aided keyword reduction for text categorization. *Applied Artificial Intelligence*, **15**, 843-873.

Johnson, D.S. (1974). Approximation algorithms for combinatorial problems, *Journal of Computer and System Sciences*, **9**, 256-278.

Li, Y., Shiu, S.C.K., & Pal, S.K. (2005). Combining feature reduction and case selection in building CBR classifiers. To appear in S.K. Pal, David W. Aha, & K.M Gupta (Eds.) *Case-based reasoning in knowledge discovery and data mining*. New York, NY: Wiley.

Pal, S.K., & Shiu, S.C.K. (2004). *Foundations of soft case-based reasoning*. Hoboken, NJ: Wiley.

Pawlak, Z. (1991). *Rough sets: Theoretical aspects of reasoning about data*. Dordrecht, Poland: Kluwer.

Popova, V.N. (2004). *Knowledge discovery and monotonicity*. Doctoral dissertation, Rotterdam School of Economics, Erasmus University, The Netherlands.

Reuters (2005). [http://www.daviddlewis.com/resources/testcollections/reuters21578]

Wiratunga, N., Koychev, I., & Massie, S. (2004). Feature selection and generalization for retrieval of textual cases. *Proceedings of the Seventh European Conference on Case-Based Reasoning* (pp. 806-820). Madrid, Spain: Springer.

Yang, Y., & Pederson, J. (1997). A comparative study of feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 412-420). Nashville, TN: Morgan Kaufmann.