A Rough Set-Aided System for Sorting WWW Bookmarks

Richard Jensen and Qiang Shen

Institute for Representation and Reasoning
Division of Informatics
The University of Edinburgh
Edinburgh EH1 1HN, UK

Abstract. Most people store 'bookmarks' to web pages. These allow the user to return to a web page later on, without having to remember the exact URL address. People attempt to organise their bookmark databases by filing bookmarks under categories, themselves arranged in a hierarchical fashion. As the maintenance of such large repositories is difficult and time-consuming, a tool that automatically categorises bookmarks is required. This paper investigates how rough set theory can help extract information out of this domain, for use in an experimental automatic bookmark classification system. In particular, work on rough set dependency degrees is applied to reduce the otherwise high dimensionality of the feature patterns used to characterize bookmarks. A comparison is made between this approach to data reduction and a conventional entropy-based approach.

1 Introduction

As the use of the Web becomes more prevalent and the size of personal repositories grows, adequately organising and managing bookmarks becomes crucial, somewhat analogous to the need to organise files in a private disk. Several years ago, in recognition of this problem, web browsers included support for tree-like folder structures for organising bookmarks. These enable the user to browse through their repository to find the necessary information. However manual URL classification and organisation can be difficult and tedious when there are more than a few bookmarks to classify - something that goes against the whole grain of the bookmarking concept.

An empirical study on users' World Wide Web page revisitation patterns (as reported in [1]) found that 58% of pages viewed are revisits. So over half of the instances where a user accesses a page, they are revisiting it (probably via their bookmark database). Another survey was carried out by the GVU's WWW Surveying Team [2] to determine which bookmarking activities are performed by different groups of people. Most respondents create entries (86%), delete entries (74%), create folders (70%) and rearrange entries (63%), with only 4% saying that they do not use them at all. Those creating sub-folders, however, were comparatively low.

This suggests that although people spend time creating and rearranging their bookmarks, the hierarchy tends to have a shallow tree-like structure. This could be for the following reasons:

- Many usability studies, for example [3], indicate that a deep hierarchy results in less efficient information retrieval as many traversal steps are required, so users are more likely to make mistakes.
- Users do not have the time/patience to arrange their collection into a well-ordered hierarchy. Also, if the tree has been ordered and is quite deep, it can take too long to traverse the sub-folders to reach the desired bookmark.

It seems, then, that there is a need for a tool that can automatically create folders and sub-folders and classify bookmarks into them. Surprisingly, few such systems are in existence; two worth noting are the Bookmark Organiser [4] and PowerBookmarks [5]. However, these approaches rely on information other than that contained in the bookmark databases. Both applications use the information contained in the documents pointed to by the URLs in order to generate classifications.

Many classification problems involve high dimensional descriptions of input features. It is therefore not surprising that much research has been done on dimensionality reduction. However, existing work tends to destroy the underlying semantics of the features after reduction (e.g. transformation-based approaches [6]) or require additional information about the given data set for thresholding (e.g. entropy-based approaches [7]). A technique that can reduce dimensionality using information contained within the data set and preserving the meaning of the features is clearly desirable. Rough set theory can be used as such a tool to discover data dependencies and reduce the number of attributes contained in a dataset by purely structural methods.

The rest of this paper is structured as follows. Section 2 introduces the main approach to dimensionality reduction, namely *Rough Set Attribute Reduction* and also highlights the operation of an additional technique, *Entropy-based Reduction*. The modular design of the bookmark classification system is described in section 3; each module involved is detailed. Section 4 presents the experimental results obtained and section 5 concludes the paper and mentions some important future work.

2 Dimensionality Reduction

The datasets generated in Information Retrieval systems tend to be extremely large, rendering most classifiers intractable. This results in the need for a mechanism that will greatly reduce the dimensionality of these datasets, whilst retaining important information. To be self-contained, this section presents those techniques that have been developed for this purpose.

2.1 Rough Set-based Reduction

A rough set [8] is an approximation of a vague concept by a pair of precise concepts, called lower and upper approximations (which are a classification of the domain of interest into disjoint categories). The classification formally represents knowledge about the problem domain. Objects belonging to the same category characterized by the same attributes (or features) are not distinguishable.

Central to Rough Set Attribute Reduction (RSAR) is the concept of indiscernibility. Let $I = (U, A)$ be an information system, where $\mathbf{U}$ is a non-empty set of finite objects (the universe). A is a non-empty finite set of attributes such that $a : \mathbf{U} \to V_a$ for every $a \in A$; V_a is the value set for attribute a. In a decision system, $A = \{C \cup D\}$ where C is the set of conditional attributes and D is the set of decision attributes. With any $P \subseteq A$ there is an associated equivalence relation $IND(P)$:

$$IND(P) = \{(x, y) \in U^2 \mid \forall\, a \in P\; a(x) = a(y)\} \tag{1}$$

If $(x, y) \in IND(P)$, then x and y are indiscernible by attributes from P. The equivalence classes of the P-indiscernibility relation are denoted $[x]_P$. Let $X \subseteq U$, the P-*lower* approximation of a set can now be defined as:

$$\underline{P}X = \{x \mid [x]_P \subseteq X\} \tag{2}$$

Let P and Q be equivalence relations over $\mathbf{U}$, then the positive region can be defined as:

$$POS_P(Q) = \bigcup_{X \in \mathbf{U}/Q} \underline{P}X \tag{3}$$

The positive region contains all objects of $\mathbf{U}$ that can be classified to classes of $\mathbf{U}/Q$ using the knowledge in attributes P.

An important issue in data analysis is discovering dependencies between attributes. Intuitively, a set of attributes Q depends totally on a set of attributes P, denoted $P \Rightarrow Q$, if all attribute values from Q are uniquely determined by values of attributes from P. If there exists a functional dependency between values of Q and P, then Q depends totally on P. Dependency can be defined in the following way:

For $P, Q \subset A$, Q depends on P in a degree k $(0 \le k \le 1)$, denoted $P \Rightarrow_k Q$, if

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|\mathbf{U}|} \tag{4}$$

If $k = 1$ Q depends totally on P, if $k < 1$ Q depends partially (in a degree k) on P, and if $k = 0$ Q does not depend on P.

By calculating the change in dependency when an attribute is removed from the set of considered conditional attributes, a measure of the significance of the attribute can be obtained. The higher the change in dependency, the more significant the attribute is. If the significance is 0, then the attribute is dispensible. More formally, given P, Q and an attribute $x \in P$,

$$\sigma_P(Q, x) = \gamma_P(Q) - \gamma_{P-\{x\}}(Q) \tag{5}$$

The reduction of attributes is achieved by comparing equivalence relations generated by sets of attributes. Attributes are removed so that the reduced set provides the same quality of classification as the original. A *reduct* is defined as a subset R of the conditional attribute set C such that $\gamma_R(D) = \gamma_C(D)$. A given dataset may have many attribute reduct sets, so the set R of all reducts is defined as:

$$\mathsf{R} = \{X : X \subseteq C, \gamma_X(D) = \gamma_C(D)\} \tag{6}$$

The intersection of all the sets in R is called the *core*, the elements of which are those attributes that cannot be eliminated without introducing more contradictions to the dataset. In RSAR, a reduct with minimum cardinality is searched for; in other words an attempt is made to locate a single element of the minimal reduct set $\mathsf{R}_{min} \subseteq \mathsf{R}$:

$$\mathsf{R}_{\mathsf{min}} = \{X : X \in \ \mathsf{R}, \ \forall Y \in \ \mathsf{R}, |X| \leq |Y|\} \tag{7}$$

A basic way of achieving this is to calculate the dependencies of all possible subsets of C. Any subset with $\gamma(D) = 1$ is a reduct; the smallest subset with this property is a minimal reduct. However, for large datasets this method is impractical and an alternative strategy is required.

```
$R \leftarrow \{\}$
do
    $T \leftarrow R$
    $\forall x \in (C - R)$
        if $\gamma_{R \cup \{x\}}(D) > \gamma_T(D)$
            $T \leftarrow R \cup \{x\}$
    $R \leftarrow T$
until $\gamma_R(D) = \gamma_C(D)$
return $R$
```

Fig. 1. The QuickReduct Algorithm

The QuickReduct algorithm [9] attempts to calculate a minimal reduct without exhaustively generating all possible subsets. It starts off with an empty set and adds in turn those attributes that result in the greatest increase in $\gamma_P(Q)$, until this produces its maximum possible value for the dataset (usually 1). However, it has been proved that this method does not always generate a *minimal* reduct, as $\gamma_P(Q)$ is not a perfect heuristic. It does result in a close-to-minimal reduct, though, which is still useful in greatly reducing dataset dimensionality.

An intuitive understanding of QuickReduct implies that, for a dimensionality of n, $n!$ evaluations of the dependency function may be performed for the worst-case dataset. From experimentation, the average complexity has been determined to be approximately O(n).

2.2 Entropy-based Reduction

To support the comparative study of the performance of RSAR for use in bookmark classification, the Entropy-based Reduction (EBR) technique is summarised here. This approach is based on the entropy heuristic employed by machine learning techniques such as ID3 [10]. A similar approach has been adopted in [11] where an entropy measure is used for ranking features.

EBR is concerned with examining a dataset and determining those attributes that provide the most gain in information. The entropy of attribute A (which can take values $a_1...a_m$) with respect to the conclusion C (values $c_1...c_n$) is defined as:

$$E(A) = -\sum_{j=1}^{m} p(a_j) \sum_{i=1}^{n} p(c_i|a_j)\ log_2\ p(c_i|a_j) \tag{8}$$

Using this function, the entropy of each conditional attribute appearing in a decision table can be calculated. The attribute with the lowest entropy is deemed to be the one that has the highest information gain, and so is the most useful determiner. By selecting only a certain number of attributes with the lowest entropies, a reduct[1] for the dataset can be constructed. Note that the determination of the number of attributes required to construct the reduct needs additional information other than given in the dataset.

In this work, for comparison, such a number is decided on by the size of a reduct produced by the rough set-based approach (which is solely determined by the dataset itself).

3 Bookmark Classification System Design

The application of rough sets to the domain of text classification has been attempted previously with some success [12], but has not yet been applied to bookmark classification. Bookmark databases are very information-poor, the useful information can only be found in the URL and title fields. Therefore, steps must be taken to ensure that all relevant information is used in the classification process, with any misleading or useless data removed.

The sorting system developed here is modular in structure, allowing various sub-components to be replaced with alternative implementations if the need arises. The main modules are *Keyword Acquisition*, *Dimensionality Reduction* and *Classification*.

To clarify the operation of the system, an example is included. The following bookmark is one of many contained in a database under the category *Programming/Java*:

```
<A HREF="http://java.sun.com/Performance/">
Ways to Increase Java Performance</A>
```

[1] The term ‘reduct’ is used loosely here.

3.1 Keyword Acquisition

In order to compare the similarity of bookmarks, a suitable representation must be chosen. Each bookmark is considered to be a vector where the ith element is the weight of term i according to some weighting method (a metric). The size of the vector is equal to the total number of keywords determined from the training documents.

This module produces weight-term pairs given a dataset. Each encountered word in a URL or title field is assigned a weight according to the metric used. Several metrics were implemented for this purpose:

- *Boolean Existential Metric.* All keywords that exist in the document are given a weight of 1, those that are absent are assigned 0 [15].
- *Frequency Count Metric.* The normalized frequency of the keywords in the document is used as the weight [14].
- *TF-IDF.* The Term Frequency-Inverse Document Frequency Metric [16] assigns higher weights to those keywords that occur frequently in the current document but not in most others. It is calculated using the formula: $w(t,i) = F_i(t) \times log\frac{N}{N_t}$ where $F_i(t)$ is the frequency of term t in document i, N is the number of documents in the collection, and N_t is the total number of documents that contain t.

For the example bookmark, the keywords $\{$*java,sun,com,performance*$\}$ are obtained from the URL, and the keywords $\{$*ways,increase,java,performance*$\}$ from the title field. Using the simple boolean existential metric, the vector elements relating to these keywords will each contain the value 1, the remainder 0.

The resulting sets of weight-term pairs, no matter which keyword acquisition metric is adopted, are large in size and need to be greatly reduced to be of any practical use for classification. Hence, the next step: *Dimensionality Reduction.*

3.2 Dimensionality Reduction

Given the weight-term sets, this module aims to significantly reduce their size whilst retaining their information content and preserving the semantics of those remaining keywords. As mentioned earlier, two approaches have been developed for this purpose, namely *RSAR* and *EBR*. Once a reduct has been calculated, the dataset can then be reduced by deleting those attributes that are absent from the reduct. The reduced dataset is now in a form that can be used by the classification module.

Returning to the example, it may be decided by this module that the term “com” provides little or no useful information. The column relating to this term is removed from the main dataset. This process is repeated for all keywords deemed to be information-poor.

3.3 Classification

This module attempts to classify a given bookmark or bookmarks using the reduced keyword datasets obtained by the dimensionality reduction stage. Each bookmark has been transformed into a weight-term vector by the keyword acquisition process. For comparison purposes, three different inference techniques were implemented to perform classification:

- *Boolean Inexact Model* [15]. This uses Boolean matching and scoring techniques. If a term exists in a document and is also present in the corresponding rule, then the score for that rule is increased; the rule with the highest score classifies the document.
- *Vector Space Model*. The vector space model [17] procedure can be divided in to three stages. The first stage is document indexing, where content bearing terms are extracted from the document text. The second stage is the weighting of the indexed terms to enhance retrieval of documents relevant to the user. The last stage ranks the document with respect to the query according to the similarity measure. The similarity measure used here is the cosine coefficient, which measures the angle between the rule vector and the query vector, and is defined as:

$$Sim(X,Y) = \frac{|X \cap Y|}{\sqrt{|X|}\sqrt{|Y|}} \tag{9}$$

- *Fuzzy Reasoner*. This follows the usual approach for the construction of fuzzy rule-based systems [18]. Reasoning is carried out by the fuzzy classifier using the dataset generated previously. All precondition memberships are evaluated, and the necessary logical conjunctions integrated (using the conventional minimum operator in the present implementation of the system). The rule with the highest score classifies the document.

4 Results

A large set of bookmarks was used as the training dataset. This database was generated by collating various online bookmark lists into one uniform collection. Each bookmark is pre-classified into a relevant category (for example, “Sports” or “Computing/Java”). An additional testing dataset of “unseen” bookmarks was also compiled from online resources.

The experiments presented here attempt to test whether RSDR is a useful tool for reducing data whilst retaining the information content. Additionally, experiments are carried out that compare the performance of RSDR with that of using EBR. Random-reduct (RR) generation (i.e. generating reducts randomly) was also used to compare the results. This method deletes random attributes from the dataset, but is constrained to leave the same number of attributes present as the RSAR method. The results of these approaches can be found in table 2.

The classification modules (vector space model (VSM), boolean inexact model (BIM) and the fuzzy reasoner (FR)) are combined in order to improve the accuracy of the system; each combination is investigated.

From table 1 it can be seen that using rough set theory, the amount of attributes was reduced to around 35%. For email classification, the average reduction

Dataset	Attributes (URL)	Attributes (Title)
Unreduced	1397	1283
RS-reduced	514	424

Table 1. Comparison of Unreduced and RS-reduced classification accuracy

of attributes was 3.5 orders of magnitude. This demonstrates that there is much less redundancy in the original datasets for the bookmark domain, which is intuitive as there is much less information in a bookmark than a document.

Dataset	VSM + BIM	VSM + FR	FR + BIM
Unreduced	55.6%	49.7%	45.0%
RS-reduced	49.1%	47.3%	42.0%
EBR-reduced	50.9%	52.7%	43.2%
RR-reduced	37.3%	34.9%	26.3%

Table 2. Comparison of reduction strategies with unreduced dataset

A comparison of the performance of the dimensionality reduction techniques is presented in table 2. The table shows that the overall accuracy is poor (obviously, the random reduction gives worst results). The main point to make here is that the ability of the system to classify new data depends entirely on the quality (and to a certain extent the quantity) of the training data. It cannot, in general, be expected that the RS-reduced or the EBR-reduced experiments should perform much better than the original unreduced dataset, which itself only allows a rather low classification rate.

In light of the fact that bookmarks contain very little useful information, the results are unsurprising and perhaps a little better than anticipated. As stated earlier, the goal is to investigate how useful rough set theory is in reducing the training dataset. For this, it is interesting to compare how well the rough set-reduced approach fares against the unreduced dataset. Consider the unreduced dataset results to be the optimum, the table can then be rewritten as:

Dataset	VSM + BIM	VSM + FR	FR + BIM
RS-reduced	88.3%	95.2%	93.3%
EBR-reduced	91.5%	106%	96.0%
RR-reduced	67.1%	70.2%	58.4%

Table 3. Comparison of reduction strategies

Viewed this way, it can be seen that EBR has the best results for each classifier pair, and is in fact better than the unreduced dataset in one instance.

This could be due to the fact that EBR selects those attributes that provide the largest gain in information. This process might ignore otherwise misleading attributes that the unreduced dataset contains. The RS-reduced dataset can be thought of as a smaller version of the original dataset, and so this will fall prey to the same mistakes.

Importantly, the performance of the RS-reduced dataset is almost as good. Although a very small amount of important information may have been lost in the rough set reduction approach, this information loss is not significant enough to reduce classification accuracy significantly, while the reduction of dimensionality is substantial.

The success of EBR in generating useful reducts is a little surprising, due to its straightforward approach. As an alternative data reduction technique, it fares well against RSDR. However, with EBR a threshold needs to be specified beforehand. With no RSDR reducts to estimate this value, there is no method available for discovering the appropriate number of attributes that should appear. Another drawback with EBR is that it cannot find more than one possible reduct, which is perfectly fine for applications such as this, but may not be for more theoretical investigations.

5 Conclusion

Results clearly show that rough set theory can be used to significantly reduce the dimensionality of the training dataset without much loss in information content. The measured drop in classification accuracy was between 0.6% and 4% for the training dataset, which is within acceptable bounds.

The main limitation of this system is that it will only be as good as the training dataset itself. Ideally, a much larger database of bookmarks would have been used, but this would have required far too much time. It is not known how long it would take the QuickReduct algorithm to find a reduct for such a large dataset as it takes many hours to find one for the existing training dataset. A related problem is how to effectively handle the dynamic aspect of bookmarking. Typically, a user's collection changes gradually over time, so an interesting extension to this work would be to incorporate these types of changes into the the learning framework.

It has already been mentioned that the QuickReduct algorithm is not always guaranteed to find a minimal reduct. One potential solution to this problem is to include an N-lookahead step before choosing the next attribute. This and other approaches are being investigated, including the use of distinction tables to determine the choice of attribute. Work is also being carried out that focuses on improving the speed and efficiency of QuickReduct. A promising research area being investigated is that of fuzzifying reducts [19]. This could be achieved by fuzzifying the dependency degree (the γ function), using fuzzy-rough sets.

6 Acknowledgements

The authors are grateful to the UK EPSRC for their support of this research, under grant 99407338 and 00317404. They are also very grateful to Alexios

Chouchoulas for helpful discussions and contributions, whilst taking full responsibility of the views expressed in this paper.

References

1. L. Tauscher and S. Greenberg, Revisitation patterns in World Wide Web navigation, in: Proc. 1997 ACM CHI Conference, Atlanta, GA, March 1997.
2. Georgia Tech Research Corporation, GVU's 8th WWW User Survey, 1997, information available at http://www.gvu.gatech.edu/user_surveys/survey-1997-10/
3. K. Larson and M. Czerwinski, Web page design: implications of memory, structure and scent for information retrieval, in: Proc. 1998 ACM SIGCHI Conf. on Human Factors in Computing Systems, Los Angeles, CA, April 1998, pp. 25-32.
4. Y. S. Maarek, I. Z. Ben Shaul. Automatically Organizing Bookmarks per Contents. Fifth International World Wide Web Conference 1996, Paris, France. http://www5conf.inria.fr/fich_html/papers/P37/Overview.html
5. W. Li, Q. Vu, D. Agrawal, Y. Hara, H. Takano. PowerBookmarks: a system for personalizable Web information organization, sharing, and management. Proceedings of the Eighth International World Wide Web Conference, Toronto, Canada, 11-14 May 1999, ISBN 0-444-50264-5.
6. P. Devijver and J. Kittler, (1982) *Pattern Recognition: A Statistical Approach.* Prentice Hall.
7. T. Mitchell (1997) *Machine Learning.* McGraw-Hill.
8. Z. Pawlak. Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishing, Dordrecht, 1991.
9. Q. Shen and A. Chouchoulas. A Modular Approach to Generating Fuzzy Rules with Reduced Attributes for the Monitoring of Complex Systems. Engineering Applications of Artificial Intelligence, 13(3):263-278, 2000.
10. J.R. Quinlan. Induction of Decision Trees. Machine Learning 1(1), pp. 81-106. 1986.
11. M. Dash, H. Liu, J. Yao. Dimensionality Reduction of Unsupervised Data. Proceedings of the 9th International Conference on Tools with Artificial Intelligence (ICTAI'97).
12. A. Chouchoulas and Q. Shen. Rough set-aided keyword reduction for text categorisation. Applied Artificial Intelligence, 2001.
13. H. S. Heaps, Information retrieval, computational and theoretical aspects. Academic Press, 1978.
14. G. Salton, Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
15. G. Salton, E. A. Fox, and H. Wu, (Cornell Technical Report TR82-511) Extended Boolean Information Retrieval. Cornell University. August 1982.
16. G. Salton, and C. Buckley. Term Weighting Approaches in Automatic Text Retrieval. Technical Report TR87-881, Department of Computer Science, Cornell University, 1987. Information Processing and Management Vol.32 (4), p. 431-443, 1996.
17. C.J. van Rijsbergen. Information Retrieval. Butterworths, London, United Kingdom, 1979. http://www.dcs.gla.ac.uk/Keith/Preface.html.
18. W. Pedrycz, and F. Gomide. An Introduction to Fuzzy Sets: Analysis and Design. The MIT Press, 1998.
19. R. Jensen. Rough-Fuzzy Methods for Determining Fuzzy Reducts. Project Report. The University of Edinburgh, 2001.

This article was processed using the LaTeX macro package with LLNCS style