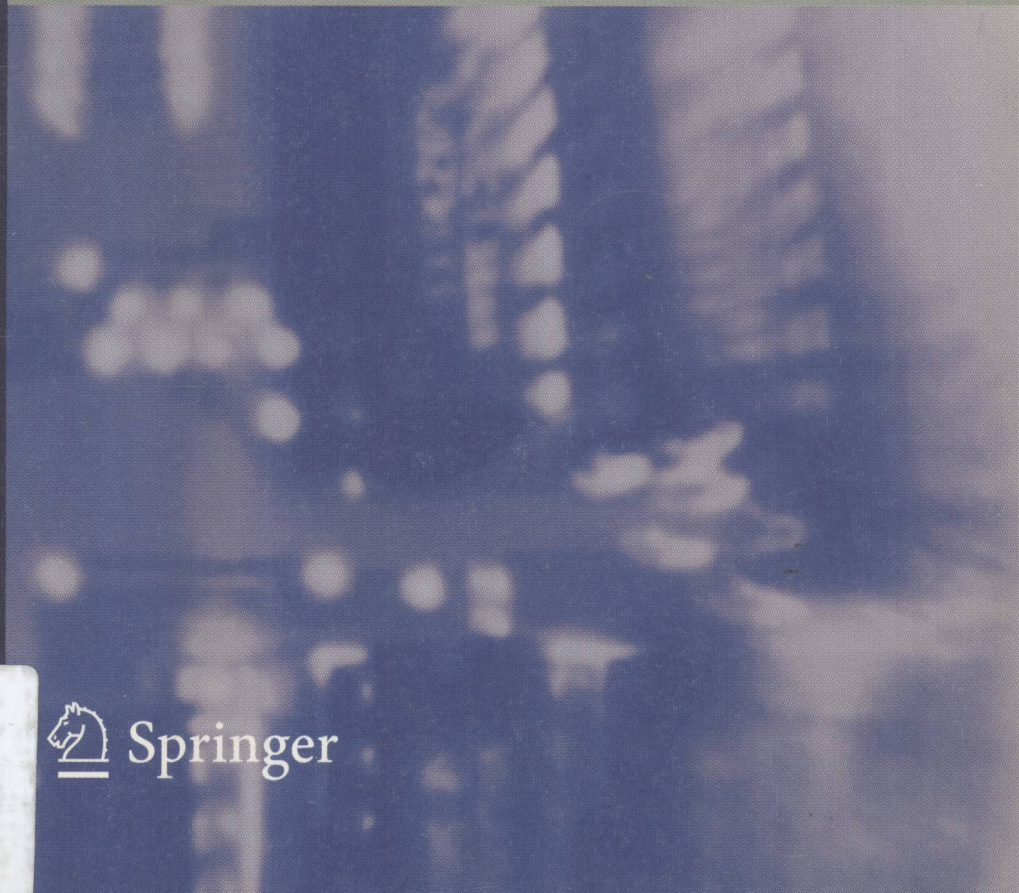
LNCS 3135 Journal Subline

Transactions on Rough Sets II

Rough Sets and Fuzzy Sets

James F. Peters · Andrzej Skowron
Editors-in-Chief



Springer

James F. Peters Andrzej Skowron
Didier Dubois Jerzy W. Grzymała-Busse
Masahiro Inuiguchi Lech Polkowski (Eds.)

Transactions on Rough Sets II

Rough Sets and Fuzzy Sets

Springer

Library of Congress Control Number: 2004115993

CR Subject Classification (1998): F.4.1, F.1, I.2, H.2.8, I.5.1, I.4

ISSN 0302-9743
ISBN 3-540-23990-1 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11322016 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3135

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

LNCS Transactions on Rough Sets

This journal subline has as its principal aim the fostering of professional exchanges between scientists and practitioners who are interested in the foundations and applications of rough sets. Topics include foundations and applications of rough sets as well as foundations and applications of hybrid methods combining rough sets with other approaches important for the development of intelligent systems.

The journal includes high-quality research articles accepted for publication on the basis of thorough peer reviews. Dissertations and monographs up to 250 pages that include new research results can also be considered as regular papers. Extended and revised versions of selected papers from conferences can also be included in regular or special issues of the journal.

Preface

This collection of articles is devoted to fuzzy as well as rough set theories. Both theories are based on rigorous ideas, methods and techniques in logic, mathematics, and computer science for treating problems for which approximate solutions are possible only, due to their inherent ambiguity, vagueness, incompleteness, etc. Vast areas of decision making, data mining, knowledge discovery in data, approximate reasoning, etc., are successfully explored using methods worked out within fuzzy and rough paradigms.

By the very nature of fuzzy and rough paradigms, outlined above, they are related to distinct logical schemes: it is well-known that rough sets are related to modal logics $S5$ and $S4$ (Orłowska, E., Modal logics in the theory of information systems, *Z. Math. Logik Grund. Math.* 30, 1984, pp. 213 ff.; Vakarelov, D., Modal logics for knowledge representation systems, *LNCS* 363, 1989, pp. 257 ff.) and to finitely-valued logics (Pagliani, P., Rough set theory and logic-algebraic structures. In *Incomplete Information: Rough Set Analysis*, Orłowska, E., ed., Physica/Springer, 1998, pp. 109 ff.; Polkowski, L. A note on 3-valued rough logic accepting decision rules, *Fundamenta Informaticae* 61, to appear).

Fuzzy sets are related to infinitely-valued logics (fuzzy membership to degree $r \in [0,1]$ expressing truth degree r) (Goguen, J.A., The logic of inexact concepts, *Synthese* 18/19, 1968–9, pp. 325 ff.; Pavelka, J., On fuzzy logic I, II, III, *Z. Math. Logik Grund. Math.* 25, 1979, pp. 45 ff., pp. 119 ff., pp. 454 ff.; Dubois, D., Prade, H., *Possibility Theory*, Plenum Press, 1988; Hájek, P., *Metamathematics of Fuzzy Logic*, Kluwer, 1998).

Algebraic as well as topological features of roughness and fuzziness are distinct. Topologically, rough sets may be described by means of topologies on families of sets (Polkowski, L., *Rough Sets. Mathematical Foundations*, Physica/Springer, 2002) whereas fuzzy sets by their nature fall into the province of topologies on function spaces (Ying-Ming Liu, Mao Kang Luo, *Fuzzy Topology*, World Scientific, 1998). Algebraically, rough sets form structures known as Łukasiewicz algebras, Heyting algebras, Post algebras, etc. (Pagliani, op. cit., Polkowski, op. cit.), whereas fuzzy set algebra involves point-wise operations on fuzzy membership functions suggested by various logical interpretations of fuzzy union, intersection, etc. (Novák, V., Perfilieva, I., Močkoř, J., *Mathematical Principles of Fuzzy Logic*, Kluwer, 1999).

Despite some differences, there have been attempts to reconcile the two theories and to form a hybrid paradigm, rough-fuzzy, or fuzzy-rough, depending on whether rough constructs are introduced in the fuzzy set framework, or conversely, fuzzy constructs are defined in the rough set framework (Dubois, D., Prade, H., Putting rough sets and fuzzy sets together. In *Intelligent Decision Systems. Handbook of Applications and Advances of Rough Sets Theory*, Słowiński, R., ed., Kluwer, 1992, pp. 203 ff.; Dubois, D., Prade, H., Similarity

versus preference in fuzzy-set based logics. In *Incomplete Information: Rough Set Analysis*, Orłowska, E., ed., Physica/Springer, 1998, pp. 441 ff.; Nakamura, A., Fuzzy rough sets, *Notes on Multiple-Valued Logic in Japan*, 9, 1988, pp. 1 ff.; Cattaneo, G., Generalized rough sets. Preclusivity fuzzy-intuitionistic (BZ) lattices, *Studia Logica*, 58, 1997, pp. 47 ff.; Pedrycz, W., Shadowed sets: bridging fuzzy and rough sets. In *Rough Fuzzy Hybridization*, Pal, S.K., Skowron, A., eds., Springer, Singapore, 1999, pp. 179 ff.; Inuiguchi, M., Tanino, T., A new class of necessity measures and fuzzy rough sets based on certainty qualifications, *LNAI* 2005, 2000, pp. 261 ff.).

The volume presented to the reader contains papers devoted to rough set theory, to fuzzy set theory, and to both theories. These papers highlight important aspects of those theories from theoretical as well as application points of view.

It is our pleasure that this volume appears in the Lecture Notes in Computer Science series of Springer-Verlag in the newly initiated sub-series of Transactions on Rough Sets. We are indebted to the editors of the subseries, Profs. Peters and Skowron for their invitation to publish the volume in this subseries. Our thanks go also to Prof. Janusz Kacprzyk who suggested that we prepare a collection of papers devoted simultaneously to rough and fuzzy theories. We would like to thank the authors, whose chapters are included in this volume, for making this possible. Our thanks go to the editors of Springer-Verlag, for their dedicated work toward giving the volume its final shape as well as to Dr. Piotr Synak who helped us with LaTeX.

May 2004

Didier Dubois
Jerzy Grzymała-Busse
Masahiro Inuiguchi
Lech Polkowski

Lecture Notes in Computer Science

For information about Vols. 1–3232

please contact your bookseller or Springer

Vol. 3337: J.M. Barreiro, F. Martin-Sanchez, V. Maojo, F. Sanz (Eds.), Biological and Medical Data Analysis. XI, 508 pages. 2004.

Vol. 3333: K. Aizawa, Y. Nakamura, S. Satoh (Eds.), Advances in Multimedia Information Processing - PCM 2004, Part III. XXXV, 785 pages. 2004.

Vol. 3332: K. Aizawa, Y. Nakamura, S. Satoh (Eds.), Advances in Multimedia Information Processing - PCM 2004, Part II. XXXVI, 1051 pages. 2004.

Vol. 3331: K. Aizawa, Y. Nakamura, S. Satoh (Eds.), Advances in Multimedia Information Processing - PCM 2004, Part I. XXXVI, 667 pages. 2004.

Vol. 3329: P.J. Lee (Ed.), Advances in Cryptology - ASIACRYPT 2004. XVI, 546 pages. 2004.

Vol. 3323: G. Antoniou, H. Boley (Eds.), Rules and Rule Markup Languages for the Semantic Web. X, 215 pages. 2004.

Vol. 3322: R. Klette, J. Žunić (Eds.), Combinatorial Image Analysis. XII, 760 pages. 2004.

Vol. 3321: M.J. Maher (Ed.), Advances in Computer Science - ASIAN 2004. XII, 510 pages. 2004.

Vol. 3316: N.R. Pal, N.K. Kasabov, R.K. Mudi, S. Pal, S.K. Parui (Eds.), Neural Information Processing. XXX, 1368 pages. 2004.

Vol. 3315: C. Lemaître, C.A. Reyes, J.A. González (Eds.), Advances in Artificial Intelligence – IBERAMIA 2004. XX, 987 pages. 2004. (Subseries LNAI).

Vol. 3312: A.J. Hu, A.K. Martin (Eds.), Formal Methods in Computer-Aided Design. XI, 445 pages. 2004.

Vol. 3311: V. Roca, F. Rousseau (Eds.), Interactive Multimedia and Next Generation Networks. XIII, 287 pages. 2004.

Vol. 3309: C.-H. Chi, K.-Y. Lam (Eds.), Content Computing. XII, 510 pages. 2004.

Vol. 3308: J. Davies, W. Schulte, M. Barnett (Eds.), Formal Methods and Software Engineering. XIII, 500 pages. 2004.

Vol. 3307: C. Bussler, S.-k. Hong, W. Jun, R. Kaschek, Kinshuk, S. Krishnaswamy, S.W. Loke, D. Oberle, D. Richards, A. Sharma, Y. Sure, B. Thalheim (Eds.), Web Information Systems – WISE 2004 Workshops. XV, 277 pages. 2004.

Vol. 3306: X. Zhou, S. Su, M.P. Papazoglou, M.E. Orlowska, K.G. Jeffery (Eds.), Web Information Systems – WISE 2004. XVII, 745 pages. 2004.

Vol. 3305: P.M.A. Sloot, B. Chopard, A.G. Hoekstra (Eds.), Cellular Automata. XV, 883 pages. 2004.

Vol. 3303: J.A. López, E. Benfenati, W. Dubitzky (Eds.), Knowledge Exploration in Life Science Informatics. X, 249 pages. 2004. (Subseries LNAI).

Vol. 3302: W.-N. Chin (Ed.), Programming Languages and Systems. XIII, 453 pages. 2004.

Vol. 3299: F. Wang (Ed.), Automated Technology for Verification and Analysis. XII, 506 pages. 2004.

Vol. 3298: S.A. McIlraith, D. Plexousakis, F. van Harmelen (Eds.), The Semantic Web – ISWC 2004. XXI, 841 pages. 2004.

Vol. 3295: P. Markopoulos, B. Eggen, E. Aarts, J.L. Crowley (Eds.), Ambient Intelligence. XIII, 388 pages. 2004.

Vol. 3294: C.N. Dean, R.T. Boute (Eds.), Teaching Formal Methods. X, 249 pages. 2004.

Vol. 3293: C.-H. Chi, M. van Steen, C. Wills (Eds.), Web Content Caching and Distribution. IX, 283 pages. 2004.

Vol. 3292: R. Meersman, Z. Tari, A. Corsaro (Eds.), On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops. XXIII, 885 pages. 2004.

Vol. 3291: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, Part II. XXV, 824 pages. 2004.

Vol. 3290: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, Part I. XXV, 823 pages. 2004.

Vol. 3289: S. Wang, K. Tanaka, S. Zhou, T.W. Ling, J. Guan, D. Yang, F. Grandi, E. Mangina, I.-Y. Song, H.C. Mayr (Eds.), Conceptual Modeling for Advanced Application Domains. XXII, 692 pages. 2004.

Vol. 3288: P. Atzeni, W. Chu, H. Lu, S. Zhou, T.W. Ling (Eds.), Conceptual Modeling – ER 2004. XXI, 869 pages. 2004.

Vol. 3287: A. Sanfeliu, J.F. Martínez Trinidad, J.A. Carrasco Ochoa (Eds.), Progress in Pattern Recognition, Image Analysis and Applications. XVII, 703 pages. 2004.

Vol. 3286: G. Karsai, E. Visser (Eds.), Generative Programming and Component Engineering. XIII, 491 pages. 2004.

Vol. 3285: S. Manandhar, J. Austin, U.B. Desai, Y. Oyanagi, A. Talukder (Eds.), Applied Computing. XII, 334 pages. 2004.

Vol. 3284: A. Karmouch, L. Korba, E.R.M. Madeira (Eds.), Mobility Aware Technologies and Applications. XII, 382 pages. 2004.

Vol. 3283: F.A. Aagesen, C. Anutariya, V. Wuwongse (Eds.), Intelligence in Communication Systems. XIII, 327 pages. 2004.

Vol. 3282: V. Guruswami, List Decoding of Error-Correcting Codes. XIX, 350 pages. 2004.

Vol. 3281: T. Dingsøyr (Ed.), Software Process Improvement. X, 207 pages. 2004.

Vol. 3280: C. Aykanat, T. Dayar, İ. Körpeoğlu (Eds.), Computer and Information Sciences - ISCIS 2004. XVIII, 1009 pages. 2004.

Vol. 3278: A. Sahai, F. Wu (Eds.), Utility Computing. XI, 272 pages. 2004.

Vol. 3275: P. Perner (Ed.), Advances in Data Mining, Applications in Image Mining, Medicine and Biotechnology, Management and Environmental Control, and Telecommunications. VIII, 173 pages. 2004. (Subseries LNAI).

Vol. 3274: R. Guerraoui (Ed.), Distributed Computing. XIII, 465 pages. 2004.

Vol. 3273: T. Baar, A. Strohmeier, A. Moreira, S.J. Mellor (Eds.), <<UML>> 2004 - The Unified Modelling Language. XIII, 454 pages. 2004.

Vol. 3271: J. Vicente, D. Hutchison (Eds.), Management of Multimedia Networks and Services. XIII, 335 pages. 2004.

Vol. 3270: M. Jeckle, R. Kowalczyk, P. Braun (Eds.), Grid Services Engineering and Management. X, 165 pages. 2004.

Vol. 3269: J. Lopez, S. Qing, E. Okamoto (Eds.), Information and Communications Security. XI, 564 pages. 2004.

Vol. 3268: W. Lindner, M. Mesiti, C. Türker, Y. Tzitzikas, A. Vakali (Eds.), Current Trends in Database Technology - EDBT 2004 Workshops. XVIII, 608 pages. 2004.

Vol. 3266: J. Solé-Pareta, M. Smirnov, P.V. Mieghem, J. Domingo-Pascual, E. Monteiro, P. Reichl, B. Stiller, R.J. Gibbens (Eds.), Quality of Service in the Emerging Networking Panorama. XVI, 390 pages. 2004.

Vol. 3265: R.E. Frederking, K.B. Taylor (Eds.), Machine Translation: From Real Users to Research. XI, 392 pages. 2004. (Subseries LNAI).

Vol. 3264: G. Paliouras, Y. Sakakibara (Eds.), Grammatical Inference: Algorithms and Applications. XI, 291 pages. 2004. (Subseries LNAI).

Vol. 3263: M. Weske, P. Liggesmeyer (Eds.), Object-Oriented and Internet-Based Technologies. XII, 239 pages. 2004.

Vol. 3262: M.M. Freire, P. Chemouil, P. Lorenz, A. Gravey (Eds.), Universal Multiservice Networks. XIII, 556 pages. 2004.

Vol. 3261: T. Yakhno (Ed.), Advances in Information Systems. XIV, 617 pages. 2004.

Vol. 3260: I.G.M.M. Niemegeers, S.H. de Groot (Eds.), Personal Wireless Communications. XIV, 478 pages. 2004.

Vol. 3259: J. Dix, J. Leite (Eds.), Computational Logic and Multi-Agent Systems. XII, 251 pages. 2004. (Subseries LNAI).

Vol. 3258: M. Wallace (Ed.), Principles and Practice of Constraint Programming – CP 2004. XVII, 822 pages. 2004.

Vol. 3257: E. Motta, N.R. Shadbolt, A. Stutt, N. Gibbins (Eds.), Engineering Knowledge in the Age of the Semantic Web. XVII, 517 pages. 2004. (Subseries LNAI).

Vol. 3256: H. Ehrig, G. Engels, F. Parisi-Presicce, G. Rozenberg (Eds.), Graph Transformations. XII, 451 pages. 2004.

Vol. 3255: A. Benczúr, J. Demetrovics, G. Gottlob (Eds.), Advances in Databases and Information Systems. XI, 423 pages. 2004.

Vol. 3254: E. Macii, V. Paliouras, O. Koufopavlou (Eds.), Integrated Circuit and System Design. XVI, 910 pages. 2004.

Vol. 3253: Y. Lakhnech, S. Yovine (Eds.), Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems. X, 397 pages. 2004.

Vol. 3252: H. Jin, Y. Pan, N. Xiao, J. Sun (Eds.), Grid and Cooperative Computing - GCC 2004 Workshops. XVIII, 785 pages. 2004.

Vol. 3251: H. Jin, Y. Pan, N. Xiao, J. Sun (Eds.), Grid and Cooperative Computing - GCC 2004. XXII, 1025 pages. 2004.

Vol. 3250: L.-J. (LJ) Zhang, M. Jeckle (Eds.), Web Services. X, 301 pages. 2004.

Vol. 3249: B. Buchberger, J.A. Campbell (Eds.), Artificial Intelligence and Symbolic Computation. X, 285 pages. 2004. (Subseries LNAI).

Vol. 3246: A. Apostolico, M. Melucci (Eds.), String Processing and Information Retrieval. XIV, 332 pages. 2004.

Vol. 3245: E. Suzuki, S. Arikawa (Eds.), Discovery Science. XIV, 430 pages. 2004. (Subseries LNAI).

Vol. 3244: S. Ben-David, J. Case, A. Maruoka (Eds.), Algorithmic Learning Theory. XIV, 505 pages. 2004. (Subseries LNAI).

Vol. 3243: S. Leonardi (Ed.), Algorithms and Models for the Web-Graph. VIII, 189 pages. 2004.

Vol. 3242: X. Yao, E. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J. Rowe, P. Tiňo, A. Kabán, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature - PPSN VIII. XX, 1185 pages. 2004.

Vol. 3241: D. Kranzlmüller, P. Kacsuk, J.J. Dongarra (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. XIII, 452 pages. 2004.

Vol. 3240: I. Jonassen, J. Kim (Eds.), Algorithms in Bioinformatics. IX, 476 pages. 2004. (Subseries LNBI).

Vol. 3239: G. Nicosia, V. Cutello, P.J. Bentley, J. Timmis (Eds.), Artificial Immune Systems. XII, 444 pages. 2004.

Vol. 3238: S. Biundo, T. Frühwirth, G. Palm (Eds.), KI 2004: Advances in Artificial Intelligence. XI, 467 pages. 2004. (Subseries LNAI).

Vol. 3237: C. Peters, J. Gonzalo, M. Braschler, M. Kluck (Eds.), Comparative Evaluation of Multilingual Information Access Systems. XIV, 702 pages. 2004.

Vol. 3236: M. Núñez, Z. Maamar, F.L. Pelayo, K. Pousttchi, F. Rubio (Eds.), Applying Formal Methods: Testing, Performance, and M/E-Commerce. XI, 381 pages. 2004.

Vol. 3235: D. de Frutos-Escrig, M. Nunez (Eds.), Formal Techniques for Networked and Distributed Systems – FORTE 2004. X, 377 pages. 2004.

Vol. 3234: M.J. Egenhofer, C. Freksa, H.J. Miller (Eds.), Geographic Information Science. VIII, 345 pages. 2004.

Vol. 3233: K. Futatsugi, F. Mizoguchi, N. Yonezaki (Eds.), Software Security - Theories and Systems. X, 345 pages. 2004.

Table of Contents

Feature Selection with Rough Sets for Web Page Classification

Aijun An[1], Yanhui Huang[2], Xiangji Huang[2], and Nick Cercone[3]

[1] York University, Toronto, Ontario, M3J 1P3, Canada
[2] University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada
[3] Dalhousie University, Halifax, Nova Scotia, B3H 1W5, Canada

Abstract. Web page classification is the problem of assigning predefined categories to web pages. A challenge in web page classification is how to deal with the high dimensionality of the feature space. We present a feature reduction method based on the rough set theory and investigate the effectiveness of the rough set feature selection method on web page classification. Our experiments indicate that rough set feature selection can improve the predictive performance when the original feature set for representing web pages is large.

1 Introduction

With the rapid growth of information on the World Wide Web, automatic classification of web pages has become important for effective indexing and retrieval of web documents. One approach to automatic web page classification is to apply machine learning techniques to pre–classified web data to induce profiles of categories and compare the profiles of categories with the representation of a given document in order to classify the document. A major characteristic, or difficulty, of this application is the high dimensionality of the feature space. A common approach to representing a text document is to use a "bag of words" that appear in the document. Since a web page can contain thousands of words, the feature space for representing web pages is potentially huge. Few machine learning systems can handle such a large number of features. In addition, too many features may present noise to the learning system. Therefore, it is highly desirable to reduce the feature space in order to make use of existing learning systems, to improve classification accuracy, and to speed up the learning process. It is also desirable to achieve such a goal automatically, i.e., no manual selection or construction of features is required.

Automatic feature selection methods have been used in text classification. Lewis and Ringuette [6] used an information gain measure to reduce the document vocabulary in naive Bayes classification and decision tree learning. Yang [15] used the principal component analysis to find orthogonal dimensions in the vector space of documents. Wiener *et al.* [14] used mutual information and a χ^2 statistic to select features for input to neural networks. Lang [4] used a minimum description length principle to select terms for news categorization. It has been asserted that feature selection is the most critical stage of the learning process in text classification [6].

J.F. Peters et al. (Eds.): Transactions on Rough Sets II, LNCS 3135, pp. 1–13, 2004.

We investigate the effectiveness of feature selection by rough sets on web page classification. Rough set theory is a mathematical tool for modelling incomplete or imprecise information [9]. It has been used for both feature selection and knowledge discovery in a number of real world domains, including medicine, pharmacology, control systems, social sciences, switching circuits, and image processing [13], [11]. In this paper, we apply rough set theory to feature selection for web page classification. In our application, web pages in a training data set are first represented using top frequent words. Then a feature selection method based on rough sets is applied to remove redundant features from the training data. A rule induction system, named ELEM2 [1], is then used to learn classification rules from the reduced training data. Therefore, in our application, the rough sets based feature selection is used as a pre–processing step for ELEM2. To evaluate the effectiveness of rough set feature selection on web page classification, we conduct experiments to compare the predictive performances of ELEM2 on web page classification with and without rough set feature selection. We describe our experiments and report the evaluation results.

The chapter is organized as follows. In the next section, we describe the importance of web page classification and the problems that need to be solved for web page classification. We also present our data collection and representation methods. In Sect. 3, we present the basic concepts in rough sets and describe an algorithm for computing a *reduct*,i.e., a non–redundant subset of features. The ELEM2 rule induction method is briefly introduced in Sect. 4. Our method for classifying a web page is presented in Sect. 5. In Sect. 6, we describe our evaluation methods and report experimental results. Finally, we conclude the paper in Sect. 7.

2 The Problem of Web Page Classification

The World Wide Web contains an estimate of 968 million pages as of March 2002 in the Google search engine [8] and an estimate of 7 million or more pages being added daily [5]. Describing and organizing this vast amount of content is essential for realizing the web as an effective information resource. Text classification has become an important process for helping web search engines to organize this vast amount of data. For instance, most Internet search engines, such as Yahoo and Looksmart, divide the indexed web documents into a number of categories for the users to limit the search scope. Moreover, text classification makes the results easier to browse. When the results returned by the search engine have been classified into a specified category, the users can choose the interesting category to continue browsing. Traditionally, text classification is performed manually by domain experts. However, human classification is unlikely to keep pace with the rate of growth of the web. Hence, as the web continues to increase, the importance of automatic web page classification becomes obvious. In addition, automatic classification is much cheaper and faster than human classification.

To make the text classification process automatic, machine learning techniques can be applied to generate classification models from a set of text docu-

ments with pre-labelled categories. The classification model can then be used to automatically assign natural language texts to the predefined categories based on their contents. In order to apply a machine learning technique to web page classification, the following problems need to be solved. First, to build a web page classifier, we need to collect a set of web pages as training examples to train the machine learning system. These training examples should have pre-defined class labels. Second, the content of a web page in the training set should be analyzed and the page should be represented using a formalism that the learning system requires for representing training examples. This text representation issue is central to our application. Finally, how to classify new pages with induced rules is another challenge in our application. We use different sets of features to represent training pages of different categories. Therefore, the rules for different categories are expressed using different sets of features. When classifying a new page, these different sets of rules should be used together in some way to determine the category or categories of the new page.

2.1 Data Collection

We use the Yahoo web site to collect training examples for our learning problem. Yahoo is best known for maintaining a web categorization directory. The web directory in Yahoo is a multi–level tree–structured hierarchy. The top level of the tree, which is the first level below the root of the tree, contains 14 categories. Each of these 14 categories contains sub–categories that are placed in the second level below the root. The third and fourth levels of the tree contain both further–refined categories and web pages. We use the top–level categories in Yahoo to label the web pages in our training set. Only 13 of 14 top–level categories are used and one category, named "Regional", is excluded because it has too much overlap with other categories.

We randomly selected over 7600 pages from the Yahoo category. We originally planned to gather 500 example pages from each category. Unfortunately, some web pages that have links in Yahoo were either eliminated or not connected to the Internet. In addition, some web pages contain very few terms after the removal of stop–words because these pages consist of a brief greeting sentence, image, Java script, flash, and other non–textual information. Hence, our number of training examples for each category is different. The distribution of these training examples among 13 categories considered is shown in Table 1. Categories considered may overlap. For example, a document discussing sport action may be reasonably classified into Entertainment and Recreation & Sports categories.

2.2 Representation of Web Pages

After training pages are selected, we apply Porter's stemming algorithm [10] to transfer each word in a web page into its stem. We then remove all the stop words according to a standard stop word list. For each category, we count the number of occurrences of each remaining word stem in all the pages that belong to the category. The word stems in each category are then sorted according to

Table 1. Distribution of the training data.

Category	Number of web pages
Arts & Humanities	783
Business & Economy	997
Computers & Internet	745
Education	485
Entertainment	957
Government	229
Health	772
News & Media	747
Recreation & Sports	506
Reference	501
Society & Culture	253
Science	230
Social Science	510
Total	7615

the number of occurrences. This process results in two sets of documents. One is 13 sorted lists of word stems, one for each category. These lists will be used to select features for the training data. The other set of results is the set of web pages, each represented by remaining word stems and their counts.

We use word stem counts to represent a web document. A web document may contain a huge number of words and not all the words in the global space appear in every document. If we use all the words in the global space to represent the documents, the dimensionality of the data set is prohibitively high for the learning system. In addition, even though our learning system can handle thousands of features, many of the features are irrelevant to the learning task. The presence of irrelevant features in the training data introduces noise and extra learning time. Therefore, it is necessary to reduce the dimensionality of the feature set by removing words with low frequencies. Removing infrequent words is also suggested in [7] and [16].

Different categories have different top frequent words. We collected top 60 frequent terms for each category. Since the top frequent terms differ among categories, there is no common set of features that we can use to represent all documents in all categories. Therefore, even though our learning system can deal with multi-category learning directly, we transform our learning problem into multiple two-class learning problems. That is, for each web page category, we prepare the training data using top n ($n = 20, 30, 40, 50$ or 60 in our experiments) frequent words in the category and then learn a set of rules that can be used to predict whether a new page belongs to this category or not. Therefore, for a given n, we have 13 training sets , each of them containing 7,615 documents, and represented by top n frequent terms of the corresponding category. After applying our learning algorithm, the 13 training sets lead to the generation of 13

classifiers. These 13 classifiers will vote to determine which category or categories a new page belongs to.

3 Feature Selection with Rough Sets

We use frequent words to represent the web pages in our training data. However, some frequent words may not be very relevant to our learning task. These words may have little power in discriminating documents of different categories. Therefore, further selection of relevant features is important. We apply a rough set based feature selection method for this purpose. In this section, we first introduce some concepts of rough sets and then describe an algorithm for removing unnecessary attributes.

3.1 Basic Notations

A data set can be formally described using a *decision table*. A decision table (also called an information system [9]) is defined as a quadruple $< U, A, V, f >$, where $U = x_1, x_2, ..., x_N$ is a finite set of objects or examples; A is a finite set of attributes; the attributes in A are further classified into two disjoint subsets, *condition* attributes C and *decision* attributes D such that $A = C \cup D$ and $C \cap D = \emptyset$; $V = \bigcup_{a \in A} V_a$ is a set of attribute values and V_a is the *domain* of attribute a (the set of values of attribute a); $f : U \times A \rightarrow V$ is an *information function* which assigns particular values from domains of attributes to objects such that $f(x_i, a) \in V_a$, for all $x_i \in U$ and $a \in A$. In our application, $D = \{d\}$ is a singleton set, where d is the class attribute that denotes classes of examples.

Given a decision table $DT =< U, A, V, f >$, let B be a subset of A, and let x_i and x_j be members of U; a relation $R(B)$, called an *indiscernibility relation* [9] over B, is defined as follows:

$$R(B) = \{(x_i, x_j) \in U^2 : \forall a \in B, f(x_i, a) = f(x_j, a)\}. \quad (1)$$

Let C be the set of condition attributes, and $R(C)$ be the corresponding indiscernibility relation on U; an ordered pair $AS =< U, R(C) >$ is called an *approximation space* based on C.

Let $Y \subseteq U$ be a subset of objects representing a *concept*, and $R^*(C) = \{X_1, X_2, ..., X_n\}$ be the collection of equivalence classes induced by the relation $R(C)$. The *lower approximation* [1] of a set Y in the approximation space AS denoted as $LOW_{R(C)}(Y)$, is defined as the union of those equivalence classes in the collection of $R^*(C)$ which are completely contained in the set Y, i.e.,

$$LOW_{R(C)}(Y) = \bigcup \{X \in R^*(C) : X \subseteq Y\}. \quad (2)$$

Let $R^*(D) = \{Y_1, Y_2, ..., Y_m\}$ be the collection of equivalence classes of the relation $R(D)$. The *positive region* $POS_C(D)$ of D with respect to C is defined as,

$$POS_C(D) = \bigcup_{i=1,...,m} \{LOW_{R(C)}(Y_i) : Y_i \in R^*(D)\}. \quad (3)$$

The positive region $POS_C(D)$ includes all examples of the equivalence classes of $R^*(C)$ in AS which can be certainly classified into classes of $R^*(D)$. $POS_C(D) = U$ means that there is no conflict between sets C and D in the sense that the classification by C determines the classification by D on objects in U.

3.2 Attribute Reduction

Attribute reduction techniques eliminate superfluous attributes and create a minimal sufficient subset of attributes for a decision table. Such minimal sufficient subset of attributes, called a *reduct*, is an essential part of the decision table which can discern all examples discernible by the original table and cannot be reduced any more. A subset B of a set of attributes C is a *reduct* of C with respect to D if and only if,

(1) $POS_B(D) = POS_C(D)$,
(2) $POS_{B-\{a\}}(D) \neq POS_C(D)$, for any $a \in B$.

A set C of condition attributes may contain more than one reduct. The set of attributes common to all reducts of C is called the *core* of C. The core contains all indispensable attributes of a decision table and can be defined as,

$$CORE_C(D) = \{c \in C : \forall c \in C, POS_{C-\{c\}}(D) \neq POS_C(D)\}. \quad (4)$$

A good procedure for computing a reduct for a decision table is to compute the core first and then check the other attributes one by one to see if they are essential to the system. If for any attribute $c \in C - CORE_C(D)$, $POS_{C-\{c\}}(D) \neq POS_C(D)$, then c cannot be removed from C. Since the order in which the attributes are removed affects the result of reduction, a concept called *relative significance coefficient (RSC)* is introduced to rank the condition attributes. The *relative significance coefficient* (RSC) of the attribute $c \in C$ based on the set of attributes C with respect to attributes D is defined as,

$$RSC_c(C, D) = \frac{card(POS_{C-\{c\}}(D))}{card(POS_C(D))}, \quad (5)$$

where *card* denotes set cardinality. Our algorithm for computing a reduct is outlined as follows.

1. Compute $CORE_C(D)$. For each condition attribute in C, remove it from C and check whether it changes the positive region. Let $CORE_C(D)$ be the set of all condition attributes whose removal does not change the positive region.
2. Check whether $CORE_C(D)$ is a reduct of the rule set. If yes, stop and $CORE_C(D)$ is a reduct.
3. Let $T = C - CORE_C(D)$. Rank the attributes in T in descending order of their RSC value. Let a be the first attribute in T and let C' be C.
4. Check whether $POS_{C'-\{a\}}(D) = POS_C(D)$. If yes, remove a from C'.
5. Let a be the next attribute in T. If a exists, repeat step 4; otherwise, stop and C' is a reduct.