Proceedings

Third IEEE International Conference on Data Mining

ICDM 2003

19-22 November 2003 • Melbourne, Florida

Editors
Xindong Wu and Alex Tuzhilin, *Program Chairs*
Jude Shavlik, *Conference Chair*

Sponsored by
IEEE Computer Society Technical Committee on Computational Intelligence (TCCI)
IEEE Computer Society Technical Committee on Pattern Analysis and Machine Intelligence (TCPAMI)





Los Alamitos, California

Washington • Brussels • Tokyo

Home Depot
Ex. 1006 - Page 1

QA76.9
.D343I133
2003

Copyright © 2003 by The Institute of Electrical and
Electronics Engineers, Inc.
All rights reserved

Copyright and Reprint Permissions: Abstracting is permitted with credit to the source. Libraries may photocopy beyond the limits of US copyright law, for private use of patrons, those articles in this volume that carry a code at the bottom of the first page, provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

Other copying, reprint, or republication requests should be addressed to: IEEE Copyrights Manager, IEEE Service Center, 445 Hoes Lane, P.O. Box 133, Piscataway, NJ 08855-1331.

The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors, the IEEE Computer Society, or the Institute of Electrical and Electronics Engineers, Inc.

IEEE Computer Society Order Number PR01978
ISBN 0-7695-1978-4
Library of Congress Number: 2003105566

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1-714-821-8380
Fax: + 1-714-821-4641
E-mail: cs.books@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1-732-981-0060
Fax: + 1-732-981-9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81-3-3408-3118
Fax: + 81-3-3408-3553
tokyo.ofc@computer.org

Editorial production by Stephanie Kawada
Cover art production by Joe Daigle/Studio Productions
Printed in the United States of America by The Printing House



Home Depot
Ex. 1006 - Page 2

ICDM 2003
Proceedings

Third IEEE International Conference on Data Mining

12-8-03 6a

Table of Contents

Welcome to ICDM 2003 xiv
Conference Organization xvi
Steering Committee xvii
Program Committee xviii
External Reviewers xxii
Corporate Sponsors xxiv
Invited Talks 751
Tutorials 752
Workshops 753

Research-Track Regular Papers

Efficient Multidimensional Quantitative Hypotheses Generation 3
A. Amir, R. Kashi, and N. S. Netanyahu

ExAMiner: Optimized Level-Wise Frequent Pattern Mining with Monotone Constraints 11
F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi

Mining High Utility Itemsets 19
R. Chan, Q. Yang, and Y.-D. Shen

LINDA HALL LIBRARY
Kansas City, Mo.

Zigzag: A New Algorithm for Mining Large Inclusion Dependencies in Databases 27
F. De Marchi and J.-M. Petit

Frequent Sub-Structure-Based Approaches for Classifying Chemical Compounds 35
M. Deshpande, M. Kuramochi, and G. Karypis

Optimized Disjunctive Association Rules via Sampling 43
J. Elble, C. Heeren, and L. Pitt

Is Random Model Better? On Its Accuracy and Efficiency 51
W. Fan, H. Wang, P. S. Yu, and S. Ma

Identifying Markov Blankets with Decision Tree Induction 59
L. Frey, D. Fisher, I. Tsamardinos, C. F. Aliferis, and A. Statnikov

Reliable Detection of Episodes in Event Sequences 67
R. Gwadera, M. Atallah, and W. Szpankowski

A Dynamic Adaptive Self-Organising Hybrid Model for Text Clustering 75
C. Hung and S. Wermter

Mining Significant Pairs of Patterns from Graph Structures with Class Labels 83
A. Inokuchi and H. Kashima

Scalable Model-Based Clustering by Working on Data Summaries 91
H. Jin, M.-L. Wong, and K.-S. Leung

On the Privacy Preserving Properties of Random Data Perturbation Techniques 99
H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar

Semantic Log Analysis Based on a User Query Behavior Model 107
N. Kawamae, T. Mukaigaito, and M. Hanaki

Clustering of Time Series Subsequences Is Meaningless: Implications for Previous and Future Research 115
E. Keogh, J. Lin, and W. Truppel

Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift 123
J. Z. Kolter and M. A. Maloof

Probabilistic Noise Identification and Data Cleaning 131
J. Kubica and A. Moore

Localized Prediction of Continuous Target Variables Using Hierarchical Clustering 139
A. Lazarevic, R. Kanapady, C. Kamath, V. Kumar, and K. Tamma

An Algebra for Inductive Query Evaluation 147
S. D. Lee and L. De Raedt

Direct Interesting Rule Generation 155
J. Li and Y. Zhang

Home Depot
Ex. 1006 - Page 4

Spatial Interest Pixels (SIPs): Useful Low-Level Features of Visual Media Data 163
Q. Li, J. Ye, and C. Kambhamettu

Unsupervised Link Discovery in Multi-Relational Data via Rarity Analysis 171
S. Lin and H. Chalupsky

Building Text Classifiers Using Positive and Unlabeled Examples 179
B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu

OP-Cluster: Clustering by Tendency in High Dimensional Space 187
J. Liu and W. Wang

Parsing without a Grammar: Making Sense of Unknown File Formats 195
L. Lloyd and S. Skiena

Probabilistic User Behavior Models 203
E. Manavoglu, D. Pavlov, and C. L. Giles

Privacy-Preserving Distributed Clustering Using Generative Models 211
S. Merugu and J. Ghosh

Change Profiles 219
T. Mielikäinen

Complex Spatial Relationships 227
R. Munro, S. Chawla, and P. Sun

TECNO-STREAMS: Tracking Evolving Clusters in Noisy Data Streams
with a Scalable Immune System Learning Model 235
O. Nasraoui, C. Cardona Uribe, C. Rojas Coronel, and F. Gonzalez

Efficient Nonlinear Dimension Reduction for Clustered Data Using Kernel Functions 243
C. H. Park and H. Park

Sequence Modeling with Mixtures of Conditional Maximum Entropy Distributions 251
D. Pavlov

MaPle: A Fast Algorithm for Maximal Pattern-Based Clustering 259
J. Pei, X. Zhang, M. Cho, H. Wang, and P. S. Yu

Exploiting Unlabeled Data for Improving Accuracy of Predictive Data Mining 267
K. Peng, S. Vucetic, B. Han, H. Xie, and Z. Obradovic

Statistical Relational Learning for Document Mining 275
A. Popescul, L. H. Ungar, S. Lawrence, and D. M. Pennock

Integrating Customer Value Considerations into Predictive Modeling 283
S. Rosset and E. Neumann

A High-Performance Distributed Algorithm for Mining Association Rules 291
A. Schuster, R. Wolff, and D. Trock

Home Depot
Ex. 1006 - Page 5

Introducing Uncertainty into Pattern Discovery in Temporal Event Sequences 299
X. Sun, M. E. Orlowska, and X. Li

Evolutionary Gabor Filter Optimization with Application to Vehicle Detection 307
Z. Sun, G. Bebis, and R. Miller

Detecting Interesting Exceptions from Medical Test Data with Visual Summarization 315
E. Suzuki, T. Watanabe, H. Yokoi, and K. Takabayashi

Learning Bayesian Networks from Incomplete Data Based on EMI Method 323
F. Tian, H. Zhang, and Y. Lu

Combining Multiple Weak Clusterings 331
A. Topchy, A. K. Jain, and W. Punch

Visualization of Rule's Similarity Using Multidimensional Scaling 339
S. Tsumoto and S. Hirano

TSP: Mining Top-K Closed Sequential Patterns 347
P. Tzvetkov, X. Yan, and J. Han

Interactive Visualization and Navigation in Large Data Collections
Using the Hyperbolic Space 355
J. Walter, J. Ontrup, D. Wessling, and H. Ritter

Association Rule Mining in Peer-to-Peer Systems 363
R. Wolff and A. Schuster

MPIS: Maximal-Profit Item Selection with Cross-Selling Considerations 371
R. C.-W. Wong, A. W.-C. Fu, and K. Wang

Efficient Data Mining for Maximal Frequent Subtrees 379
Y. Xiao, J.-F. Yao, Z. Li, and M. H. Dunham

Mining Strong Affinity Association Patterns in Data Sets with Skewed
Support Distribution 387
H. Xiong, P.-N. Tan, and V. Kumar

On Precision and Recall of Multi-Attribute Data Extraction
from Semistructured Sources 395
G. Yang, S. Mukherjee, and I. V. Ramakrishnan

Mining Plans for Customer-Class Transformation 403
Q. Yang and H. Cheng

Segmenting Customer Transactions Using a Pattern-Based Clustering Approach 411
Y. Yang and B. Padmanabhan

A New Optimization Criterion for Generalized Discriminant Analysis
on Undersampled Problems 419
J. Ye, R. Janardan, C. H. Park, and H. Park

Home Depot
Ex. 1006 - Page 6

Sentiment Analyzer: Extracting Sentiments about a Given Topic Using Natural Language Processing Techniques 427
J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack

Cost-Sensitive Learning by Cost-Proportionate Example Weighting 435
B. Zadrozny, J. Langford, and N. Abe

CBC: Clustering Based Text Classification Requiring Minimal Labeled Data 443
H.-J. Zeng, X.-H. Wang, Z. Chen, H. Lu, and W.-Y. Ma

Regression Clustering 451
B. Zhang

Model-Based Clustering with Soft Balancing 459
S. Zhong and J. Ghosh

Research-Track Short Papers

Integrating Fuzziness into OLAP for Multidimensional Fuzzy Association Rules Mining 469
R. Alhajj and M. Kaya

Analyzing High-Dimensional Data by Subspace Validity 473
A. Amir, R. Kashi, N. S. Netanyahu, D. Keim, and M. Wawryniuk

Objective and Subjective Algorithms for Grouping Association Rules 477
A. An, S. Khan, and X. Huang

Efficient Subsequence Matching in Time Series Databases under Time and Amplitude Transformations 481
T. Argyros and C. Ermopoulos

A Fast Algorithm for Computing Hypergraph Transversals and Its Application in Mining Emerging Patterns 485
J. Bailey, T. Manoukian, and K. Ramamohanarao

Mining Relevant Text from Unlabelled Documents 489
D. Barbará, C. Domeniconi, and N. Kang

A User-Driven and Quality-Oriented Visualization for Mining Association Rules 493
J. Blanchard, F. Guillet, and H. Briand

Towards Simple, Easy-to-Understand, yet Accurate Classifiers 497
D. Caragea, D. Cook, and V. Honavar

Validating and Refining Clusters via Visual Rendering 501
K. Chen and L. Liu

Icon-Based Visualization of Large High-Dimensional Datasets 505
P. Chen, C. Hu, W. Ding, H. Lynn, and Y. Simon

Home Depot
Ex. 1006 - Page 7

Indexing and Mining Free Trees 509
Y. Chi, Y. Yang, and R. R. Muntz

T-Trees, Vertical Partitioning and Distributed Association Rule Mining 513
F. Coenen, P. Leng, and S. Ahmed

Information Theoretic Clustering of Sparse Co-Occurrence Data 517
I. S. Dhillon and Y. Guan

Links between Kleinberg's Hubs and Authorities, Correspondence Analysis, and Markov Chains 521
F. Fouss, M. Saerens, and J.-M. Renders

Fast PNN-Based Clustering Using K-Nearest Neighbor Graph 525
P. Fränti, O. Virmajoki, and V. Hautamäki

The Rough Set Approach to Association Rule Mining 529
J. W. Guan, D. A. Bell, and D. Y. Liu

Comparing Pure Parallel Ensemble Creation Techniques against Bagging 533
L. O. Hall, K. W. Bowyer, R. E. Banfield, D. Bhadoria, W. P. Kegelmeyer, and S. Eschrich

Improving Home Automation by Discovering Regularly Occurring Device Usage Patterns 537
E. O. Heierman III and D. J. Cook

Ontologies Improve Text Document Clustering 541
A. Hotho, S. Staab, and G. Stumme

The Hybrid Poisson Aspect Model for Personalized Shopping Recommendation 545
C.-N. Hsu, H.-H. Chung, and H.-S. Huang

Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism 549
J. Huan, W. Wang, and J. Prins

Comparing Naive Bayes, Decision Trees, and SVM with AUC and Accuracy 553
J. Huang, J. Lu, and C. X. Ling

SVM Based Models for Predicting Foreign Currency Exchange Rates 557
J. Kamruzzaman, R. A. Sarker, and I. Ahmad

Facilitating Fuzzy Association Rules Mining by Using Multi-Objective Genetic Algorithms for Automated Clustering 561
M. Kaya and R. Alhajj

PixelMaps: A New Visual Data Mining Approach for Analyzing Large Spatial Data Sets 565
D. A. Keim, C. Panse, M. Sips, and S. C. North

Home Depot
Ex. 1006 - Page 8

Effectiveness of Information Extraction, Multi-Relational, and Semi-Supervised Learning for Predicting Functional Properties of Genes 569
M.-A. Krogel and T. Scheffer

Tractable Group Detection on Large Link Data Sets 573
J. Kubica, A. Moore, and J. Schneider

Tree-Structured Partitioning Based on Splitting Histograms of Distances 577
L. J. Latecki, R. Venugopal, M. Sobel, and S. Horvath

CoMine: Efficient Mining of Correlated Patterns 581
Y.-K. Lee, W.-Y. Kim, Y. D. Cai, and J. Han

Ensembles of Cascading Trees 585
J. Li and H. Liu

Using Discriminant Analysis for Multi-Class Classification 589
T. Li, S. Zhu, and M. Ogihara

Interpretations of Association Rules by Granular Computing 593
Y. Li and N. Zhong

Algorithms for Spatial Outlier Detection 597
C.-T. Lu, D. Chen, and Y. Kou

Learning Rules for Anomaly Detection of Hostile Network Traffic 601
M. V. Mahoney and P. K. Chan

An Algorithm for the Exact Computation of the Centroid of Higher Dimensional Polyhedra and Its Application to Kernel Machines 605
F. Maire

Simple Estimators for Relational Bayesian Classifiers 609
J. Neville, D. Jensen, and B. Gallagher

Protecting Sensitive Knowledge by Data Sanitization 613
S. R. M. Oliveira and O. R. Zaïane

Mining Frequent Itemsets in Distributed and Dynamic Databases 617
M. E. Otey, C. Wang, S. Parthasarathy, A. Veloso, and W. Meira, Jr.

Structure Search and Stability Enhancement of Bayesian Networks 621
H. Peng and C. Ding

Privacy-Preserving Collaborative Filtering Using Randomized Perturbation Techniques 625
H. Polat and W. Du

Semantic Role Parsing: Adding Semantic Structure to Unstructured Text 629
S. Pradhan, K. Hacioglu, W. Ward, J. H. Martin, and D. Jurafsky

Home Depot
Ex. 1006 - Page 9

Mining Semantic Networks for Knowledge Discovery 633
K. Rajaraman and A.-H. Tan

Impact Studies and Sensitivity Analysis in Medical Data Mining with ROC-Based Genetic Learning 637
M. Sebag, J. Azé, and N. Lucas

K-D Decision Tree: An Accelerated and Memory Efficient Nearest Neighbor Classifier 641
T. Shibata, T. Kato, and T. Wada

Mining the Web to Discover the Meanings of an Ambiguous Word 645
R. Tamir and R. Rapp

A Hybrid Data-Mining Approach in Genomics and Text Structures 649
H.-N. Teodorescu and L. I. Fira

Model Stability: A Key Factor in Determining Whether an Algorithm Produces an Optimal Model from a Matching Distribution 653
K. M. Ting and R. J. Y. Quek

Enhancing Techniques for Efficient Topic Hierarchy Integration 657
J.-J. Tsay, C.-F. Chang, H.-Y. Chen, and C.-H. Lin

Pattern Discovery Based on Rule Induction and Taxonomy Generation 661
S. Tsumoto and S. Hirano

Active Sampling for Feature Selection 665
S. Veeramachaneni and P. Avesani

Combining the Web Content and Usage Mining to Understand the Visitor Behavior in a Web Site 669
J. Velásquez, H. Yasuda, and T. Aoki

Class Decomposition via Clustering: A New Framework for Low-Variance Classifiers 673
R. Vilalta, M.-K. Achari, and C. F. Eick

Bootstrapping Rule Induction 677
L. R. Waitman, D. H. Fisher, and P. H. King

Center-Based Indexing for Nearest Neighbors Search 681
A. Wojna

Postprocessing Decision Trees to Extract Actionable Knowledge 685
Q. Yang, J. Yin, C. X. Ling, and T. Chen

Frequent-Pattern Based Iterative Projected Clustering 689
M. L. Yiu and N. Mamoulis

General MC: Estimating Boundary of Positive Class from Small Positive Data 693
H. Yu

Home Depot
Ex. 1006 - Page 10

Clustering Item Data Sets with Association-Taxonomy Similarity 697
C.-H. Yun, K.-T. Chuang, and M.-S. Chen

Dimensionality Reduction Using Kernel Pooled Local Discriminant Information 701
P. Zhang, J. Peng, and C. Domeniconi

A Feature Selection Framework for Text Filtering 705
Z. Zheng, R. Srihari, and S. Srihari

A K-NN Associated Fuzzy Evidential Reasoning Classifier with Adaptive Neighbor Selection 709
H. Zhu and O. Basir

Industry-Track Papers

Findings from a Practical Project Concerning Web Usage Mining 715
F. Dellmann, H. Wulff, and S. Schmitz

Predicting Distribution of a New Forest Disease Using One-Class SVMs 719
Q. Guo, M. Kelly, and C. Graham

Understanding *Helicoverpa armigera* Pest Population Dynamics Related to Chickpea Crop Using Neural Networks 723
R. Gupta, B. V. L. Narayana, P. Krishna Reddy, G. V. Ranga Rao, C. L. L. Gowda, Y. V. R. Reddy, and G. Rama Murthy

Text Mining for a Clear Picture of Defect Reports: A Praxis Report 727
J. Kreyss, S. Selvaggio, M. White, and Z. Zakharian

Mining Production Data with Neural Network and CART 731
M. Li, S. Feng, I. K. Sethi, J. Luciow, and K. Wagner

Inference of Protein-Protein Interactions by Unlikely Profile Pair 735
B.-H. Park, G. Ostrouchov, G.-X. Yu, A. Geist, A. Gorin, and N. F. Samatova

Regulatory Element Discovery Using Tree-Structured Models 739
T. M. Phuong, D. Lee, and K. H. Lee

Applying Noise Handling Techniques to Genomic Data: A Case Study 743
C. M. Teng

Detecting Patterns of Change Using Enhanced Parallel Coordinates Visualization 747
K. Zhao, B. Liu, T. M. Tirpak, and A. Schaller

Author Index 754

Home Depot
Ex. 1006 - Page 11

Segmenting Customer Transactions Using a Pattern-Based Clustering Approach

Yinghui Yang and Balaji Padmanabhan
Operations and Information Management Department
The Wharton School, University of Pennsylvania
{yiyang, balaji}@wharton.upenn.edu

Abstract

Grouping customer transactions into categories helps understand customers better. The marketing literature has concentrated on identifying important segmentation variables (e.g. customer loyalty) and on using clustering and mixture models for segmentation. The data mining literature has provided various clustering algorithms for segmentation. In this paper we investigate using "pattern-based" clustering approaches to grouping customer transactions. We argue that there are clusters in transaction data based on natural behavioral patterns, and present a new technique, YACA, that groups transactions such that itemsets generated from each cluster, while similar to each other, are different from ones generated from others. We present experimental results from user-centric Web usage data that demonstrates that YACA generates a highly effective clustering of transactions.

1. Introduction

Firms are increasingly realizing the importance of understanding and leveraging customer level data, and critical business decision models are being built upon analyzing such data. For example, Amazon.com offers distinct home pages and recommends new products for customers based on personalization models built from data. Most credit card and cellular phone fraud alerts are also issued from the analysis of customer level data. Consumer brand choice models and pricing models are heavily used in marketing endeavors.

While the expectation for customer level data analysis is high, there are still problems with existing analysis methods. Consumers still receive significant amount of mail advertising products that they are not interested in, and online recommendations are still far from perfect. In order to create more successful personalized systems and build more accurate consumer behavior models for customers, firms need to understand their customers better. This includes understanding customers' preferences through facts and customers' behavior through analyzing their transaction data. There has been much research done in this direction, and clustering transactions to learn segments has been one research stream that has generated a variety of useful approaches.

In the marketing literature, market segmentation approaches have often been used to divide customers into groups in order to implement different strategies. It has been long established that customers demonstrate heterogeneity in their product preferences and buying behaviors [2], and that the model built on the market in aggregate is less efficient than models built for individual segments. Much of this research focuses on examining how variables like demographics, socioeconomic status, personality and attitudes can be used in predicting differences in consumption and brand loyalty. Distance-based clustering techniques, such as *k*-means, and parametric mixture models, such as gaussian mixture models, are two main approaches used in segmentation. While both these approaches have produced good results in various applications, there are well known drawbacks to these methods. For clustering approaches to segmentation, the actual technique used has often been ad-hoc. It is generally not clear why a distance-based clustering in an *n* dimensional space, while convenient, is the appropriate manner to group customers. For mixture models, using changing model parameters to represent the difference between segments can often oversimplify the difference between segments and can ignore variables and patterns that are not captured by the parametric models.

In our research we study a new approach to segmenting customer transactions, one that is based on the idea that there may exist natural behavioral patterns in different groups of transactions. For example, a set of behavioral patterns that distinguish a group of wireless subscribers may be:

- Their call duration during weekday mornings is short and these calls are within the same geographical area.
- They call from outside the home area on weekdays and from the home area on weekends.
- They have several "data" calls on weekdays.

0-7695-1978-4/03 $17.00 © 2003 IEEE

Home Depot
Ex. 1006 - Page 12

The above set of three patterns may be representative of a group of 'consultants' who travel frequently and exhibit a set of common behavioral patterns. This example suggests that there may be natural clusters in data, characterized by a set of typical behavioral patterns. In such cases, appropriate "pattern-based clustering" approaches can be an intuitive method for grouping customer transactions.

At the highest level, the idea here is to cluster customer transactions such that patterns generated from each cluster, while similar to each other, are very different from the patterns generated from other clusters. We suggest that different domains may have different representations for what "patterns" are, and on how to define differences between sets of patterns. In the above example, rules are an effective representation for patterns generated from the wireless data, but in a different domain, such as time series data on stock prices, representations for patterns may be based on "shapes" in the time series. It is easy to see that traditional distance-based clustering techniques and mixture models are not well suited to learning clusters where the fundamental characterization is a set of patterns such as the ones above.

A reason why pattern-based clustering techniques can generate natural clusters from customer transactions is that such transactions often have natural categories *that are not directly observable from the data.* For example, Web transactions may be for work, for entertainment, shopping for self, shopping as gifts, transactions made while in a happy mood etc. However, customers do not indicate whether they are happy or sad, for example, before starting each transaction. However, the set of patterns corresponding to transactions in each category will be different. Transactions at work may be quicker and more focused while transactions for entertainment may be long and across a broader set of sites. Hence grouping transactions such that the patterns generated from each cluster are 'very different' from those generated from another cluster may be an effective method for learning the natural categorizations. This argument suggests a natural evaluation technique that is used in this paper. We combine transactions with a known natural category – Web transactions from different users (without maintaining the user ID) - and examine how well pattern-based clustering does in separating transactions that belong to the individual users as compared to traditional clustering techniques.

Motivated by the above argument, in this paper we investigate the utility of pattern-based clustering for grouping Web transactions. In particular, we argue that itemsets are a natural representation for patterns in Web transactions and present YACA, a pattern-based clustering algorithm for domains where itemsets are the natural representation.

In general the idea of using patterns to group data has been applied for other specific examples in the literature [11] and in particular, there has been prior work that clustered based on items [3, 8, 12, 13]. We describe these in the related work in Section 5. The rest of this paper is structured as follows. In Section 2 we describe the domain and develop an objective function that will be used to generate the pattern-based clusters. This is followed by a description of YACA, the algorithm used to generate the clusters. We present experimental results in Section 4 followed by related work and conclusions.

2. Pattern-Based Clustering of Web Transactions

In this paper we assume that users' Web transactions are data records containing categorical and numeric attributes created from the raw data consisting of a series of URLs visited. In our experiments, we start with real session-level Web browsing data for users (provided by a market data vendor [10]) and create 46 features describing the session. The specific features are listed in Table 1.

Note that the features include those about time (e.g. average time spent per page), quantity (e.g. number of sites visited) and the order of pages visited (e.g. first site) and that the features include both categorical and numeric types. A conjunction of atomic conditions on these attributes is a good representation for common behavioral patterns in the transaction data. For example $\{starting\_time = morning, average\_time\_page < 2 minutes, num\_categories = 3, total\_time < 10 minutes\}$ is a behavioral pattern that may capture a user's specific 'morning' pattern of Web usage that involves looking at multiple sites (e.g. work email, news, finance etc) in a focused manner such that the total time spent is low. Another common pattern for this (same) user may be $\{starting\_time = night, most\_visted\_category = games\}$ reflecting her typical behavior at the end of days. Hence, in order to capture the typical behavioral patterns in Web transactions, we use itemsets as the representation for patterns. Note that in general we assume that the items in the itemsets can involve both categorical and numeric attributes as described in the examples above, and the functions defined below for the clusters are for the general case. However in the implementation described in Section 4, we bin the numeric attributes into categories to enable the application of Apriori-type methods for learning the frequent itemsets from the data. Below we describe the objective function of the pattern-based clustering method.

Home Depot
Ex. 1006 - Page 13

Table 1. Features of a session

Categories	Metric	Definition
Time-related	Total time	
	Average time per page	Total time/ # of pages
	Average time per site	Total time/ # of sites
	Average time per category	Total time/ # of categories
	Starting time	
	Starting day	
	Most visited site	
	Most visited category	
Quantity-related	number of pages	
	number of sites	
	number of categories	
	average # of pages per site	# of pages / # of sites
	average # of sites per categories	# of sites / # of categories
	# of sites for each category (total 27 categories)	
Order-related	first site	
	second site	
	last site	
	first category	
	second category	
	last category	

Page: individual Web page, each hit is a page;
Site: domain name, such as www.yahoo.com ;
Category: Such as "travel site", "news site" etc.

Consider a collection of transactions to be clustered $\{ T_1, T_2, \ldots, T_n \}$. Each transaction T_i contains a subset of a list of candidate items $\{ i_1, i_2, \ldots, i_m \}$. A clustering C is a partition $\{ C_1, C_2, \ldots, C_k \}$ of $\{ T_1, T_2, \ldots, T_n \}$ and each C_i is a cluster. The goal of our method is to maximize the difference between clusters and the similarity of transactions within clusters. In another word, we cluster to maximize a quantity M, where M is defined as follows:

$$M(C_1, C_2, \ldots, C_k) = Difference(C_1, C_2, \ldots, C_k) + \sum_{i=1}^{k} Similarity(C_i)$$

More specifically, the difference is defined based on the idea that frequent itemsets generated from each cluster C_i are "very different" from the frequent itemsets generated from other clusters. "Very different" is defined based on the difference in support values of the itemsets generated in each cluster.

Note that this suggests that itemsets are generated for each cluster separately (i.e. based on the transactions within that cluster only). However, to avoid repeated executions of the itemset discovery routine, in practice we can generate the set of all itemsets once (from the entire transaction data), and re-calibrate the support values for the itemsets in any cluster as needed. For example, assume that the total data size is 100,000 transactions and the minimum number of transactions in a cluster is set to 1000. For a minimum support of 5%, the smallest frequent itemset in any cluster has a support of 50 transactions. Hence we set the minimum support to (50/100,000) for the entire database to learn the set of all itemsets that could be frequent in any cluster. Subsequently for any group of transactions we update the set of frequent itemsets by determining the actual support values for all the itemsets within this set of transactions.

Let *FIS* be the set of all the frequent itemsets (we call it the candidate itemsets) based on the entire transaction data (before any clustering). After clustering, we calculate the support for each itemset in *FIS* for each cluster. Note that a candidate itemset is frequent in the entire set of transactions before clustering, but not necessarily frequent for each cluster. Specifically, $FIS = \{is_1, is_2, \ldots, is_p\}$ where is_i is an itemset in *FIS* and s_{ij} is the support of itemset i in cluster j (see Table 2).

Table 2. Matrix of itemsets' supports

	C_1	C_2	...	C_k
is_1	s_{11}	s_{12}	...	s_{1k}
is_2	s_{21}	s_{22}	...	s_{2k}
...	...	...	...	...
is_p	s_{p1}	s_{p2}	...	s_{pk}

Home Depot
Ex. 1006 - Page 14

Next, we define the difference between two clusters and the similarity of the transactions within a cluster using the support of the itemsets.

Definition 1 (Difference of two clusters)

$$Difference\ (C_i, C_j) = \sum_{a=1}^{p} \frac{\left| \frac{S_{ai}}{|C_i|} - \frac{S_{aj}}{|C_j|} \right|}{\frac{1}{2}\left(\frac{S_{ai}}{|C_i|} + \frac{S_{aj}}{|C_j|} \right)}$$

$|C_i|$ is the number of transactions in cluster C_i and p is the number of itemsets in *FIS*. This is a good measure since for every itemset, it calculates the difference between the pattern in each cluster by computing the difference between the support values (adjusted by the actual support values). For example, if the itemset {*starting_time=afternoon, most_visited_category=sports*} occurs in 3% of the transactions in cluster-1 while it occurs 9% of time in cluster-2, the difference is 1, while if it occurs 50% and 75% of time respectively the difference is 0.4.

Note that we have defined the difference between two clusters only. This is sufficient since hierarchical clustering techniques can be used to repeatedly cluster the transactions into two groups in such a way that the process results in clustering the transactions into an arbitrary number of clusters (which is generally desirable since the number of clusters do not have to be specified upfront). Clearly, alternative definitions of difference are possible and the above function, though intuitive, is one among possibly several reasonable difference definitions between sets of itemsets.

Definition 2 (Intra-cluster similarity):
Here, our goal is to define a metric that captures what it means for patterns in one cluster to be "similar" to each other. An itemset is considered "good" if it is frequent and hence we define similarity based on the number of frequent itemsets within a cluster.
Similarity $S(C_i)$ = the number of frequent itemsets in cluster C_i

In this section we described the use of itemsets for clustering Web transactions and presented an objective function to maximize in order to generate the pattern-based clusters. In the next section we describe YACA, an algorithm for pattern-based clustering.

3. The Clustering Algorithm

The ideal algorithm will be one which maximizes M (defined in previous section). However, for the objective function defined above, if there are n transactions and two clusters that we are interested in learning, the number of possible clustering schemes to examine is 2^n. Hence we need a heuristic approach.

After we get *FIS*, as described in Section 2, we convert the initial transactions into the format in Table 3, where the rows represent transactions to be clustered and the columns represents itemsets. A "1" in a cell indicates that a certain transaction contains a certain itemset.

Table 3. Transaction represented in itemsets

	is_1	is_2	...	is_p
T_1	1	0	...	0
T_2	1	0	...	1
...	...	...	...	...
T_n	0	1	...	0

We use $T'=\{ T'_1, T'_2, \ldots, T'_n \}$ to represent this new transaction set. Instead of clustering the original transactions, we convert the problem into clustering these binary vectors and present a divisive hierarchical algorithm. The entire transaction set is first divided into two clusters. Each cluster is further divided into two clusters if it has more than a predefined number of transactions. This process is repeated until no cluster is big enough to be divided further. In addition to cluster size, we assume the stopping conditions are user-specified and can contain additional criteria such that if a cluster's size is bigger than the threshold but its quality is very good (contains very similar transactions), we can stop dividing that cluster.

In order to generate balanced clusters, we introduce another component to M. Also because each division creates two clusters, the revised M is specified as follows:

$$M(C_1,C_2) = K_1 \cdot D(C_1,C_2) + K_2 \cdot S(C_1) + K_3 \cdot S(C_2) - K_4 \cdot |N_1 - N_2|$$

K_1, K_2, K_3, K_4 are user-specified weights (can be decided based on different applications),
$D(C_1, C_2)$ represents the inter-cluster difference,
$S(C_1)$ and $S(C_2)$ are the intra-cluster similarity for cluster-1 and cluster-2 respectively, and
N_1 and N_2 are the number of transactions in cluster-1 and cluster-2 respectively.

The candidate itemsets are generated from association rule discovery algorithms such as Apriori [1]. Figure 1 describes the algorithm YACA.

For each division, a randomly chosen first transaction and the 'most different' transaction (in terms of Euclidean distance) from the first transaction are chosen to form two initial clusters, C_1 and C_2. Then, every other transaction is either assigned to C_1 or C_2 whichever maximizes M.

Home Depot
Ex. 1006 - Page 15

Input: $C = \{C_0\}$
$C_0 = \{T'_1, T'_2, \ldots, T'_n\}$ – the converted transactions to be clustered (see Table 3)
Output: Clusters $C = \{C_1, C_2, \ldots, C_f\}$

Repeat {
Choose any cluster $X = \{T_1, T_2, \ldots, Ts\}$ from C such that the stopping condition is not satisfied for X;
$C_1 = \{\}, C_2 = \{\}$;
Assign T_1 in X to C_1;
Assign T_m to C_2 | T_m is 'most different' from T_1;
for i=1 to s {
Allocate T_i to C_1 or C_2 to maximize M;
}
$C = (C - X) \cup C_1 \cup C_2$;
} While the stopping condition is not satisfied for every X in C

Figure 1 *Algorithm YACA*

4. Experiments: Clustering user-centric Web transactions

Evaluating whether our grouping of transactions using patterns is "good" is a hard problem since it, like any other clustering technique, is an unsupervised learning technique. However, as mentioned in the introduction, in order to test the efficacy of pattern-based clustering it is natural to combine transactions with a known category – Web transactions from different users (without maintaining the user ID) - and examine how the method does in separating transactions that belong to the individual users as compared to traditional clustering techniques. This is the approach used in our experiments. We pick out the sessions belonging to a certain number of users to form a sub-dataset which is used as the input to YACA. The users we picked out are those who have the most sessions. The most active users have approximately 2500 sessions over that year. In order to demonstrate the robustness of our approach, we constructed 30 such sub-datasets. Among these 30 datasets, 10 datasets contain sessions from 2 users, 10 contain sessions from 3 users, and 10 contain sessions from 4 users. There are approximately 5000 sessions in each of the 2-user datasets, 7500 sessions in each 3-user set, and 10000 in each 4-user set. The transactions of each user capture his/her online activity over a period of a year.

To construct the data, we start with user-centric data provided by a market data vendor. User-centric Web transaction data is data collected at the user level, and thus captures the entire Web surfing behavior for each user. First, we group individual hits into sessions using a common rule of thumb that groups consecutive hits within 30 minutes of each other into a session. As described in Section 2 (see Table 1), we create features to describe the user's behavior for each session and construct 46 features for each session. We create an item for each value of the categorical features and categorize the continuous variables according to their value distribution (uniformly binned), and create an item for each bin. After itemizing, we create a session (transaction) by item matrix (Table 4) with binary values for each sub-dataset. A "1" in a cell indicates that a certain transaction contains a certain item. $T_1, T_2, \ldots, T_n$ correspond to the sessions in a sub-dataset. Depending on the sub-dataset, there are approximately 2000 to 5000 items for each sub-dataset.

Table 4. Transactions represented in items

	i_1	i_2	...	i_p
T_1	1	0	...	1
T_2	1	1	...	0
...	...	...	...	...
T_n	0	1	...	1

From this matrix, we use Apriori to discovery the candidate itemsets and create a matrix like the one in Table 3 used as the input to YACA. In the experiment, the minimum support we set for Apriori to discover frequent itemsets is 3%. There are 2000 to 6500 itemsets (we restricted these to have a maximum of 4 items) for different sub-datasets.

In order to compare the performance of YACA, we implemented three different methods: (i) Hierarchical *k*-means on data represented in items as in Table 4 (the "*k*means-items" approach) (ii) Hierarchical *k*-means on data represented in itemsets as in Table 3 (the "*k*means-itemsets" approach) and (iii) YACA on data represented in itemsets as in Table 3 (the "YACA-itemsets" approach). Hierarchical *k*-means is described in Figure 2.

Input: $C = \{C_0\}$
$C_0 = \{T'_1, T'_2, \ldots, T'_n\}$ - the converted transaction set to be clustered (the transactions can take the forms both as that in Table 4 ("*k*-means-items" approach) and that in Table 3 ("*k*means-itemsets" approach).
Output: Clusters $C = \{C_1, C_2, \ldots, C_f\}$

Repeat {
Choose any cluster $X = \{T_1, T_2, \ldots, Ts\}$ from C such that the stopping condition is not satisfied for X;
Use *k*-means to cluster X into 2 clusters C_1 and C_2;
$C = (C - X) \cup C_1 \cup C_2$;
} While the stopping condition is not satisfied for every X in C

Figure 2. Hierarchical k-means

Home Depot
Ex. 1006 - Page 16

The reason we use hierarchical *k*-means is because the number of natural clusters are not known in advance. This is particularly useful since even for a sub-dataset containing transactions from 2 users, we expect more than 2 clusters, since each user may have more than one behavior pattern. Note that if we use just *k*-means instead of hierarchical *k*-means, we get inferior results. For example, when we applied *k*-means on a 2-user dataset, each resulting cluster contains approximately 50% of the transactions from each user. By using hierarchical clustering, around 70% of the clusters we get are more than 95% pure (a cluster is considered "95% pure" if at least 95% of transactions in the cluster belong entirely to a single user). When we cluster using YACA, we set the value of K_1 to be 1, K_2 and K_3 to be 1.1, K_4 to be 100, and the support threshold that we used to define large itemsets to be 2%, and have the same value for these parameters through out the entire experiment. These values are chosen to bring the four components of M to relatively compatible scales. We only consider itemsets which contain less than 4 items and the stopping criterion is that the size of the cluster is smaller than 5% of the size of the dataset to be clustered.

Figures 3~5 present the results on the 30 datasets. The number of clusters generated for each sub-dataset is between 20 and 100 and the three methods generate relatively balanced clusters. Hence we use the percentage of 95% pure clusters generated to measure the goodness of the clustering algorithms. This is a strong measure given that all the datasets contain approximately equal number of transactions from each user.

Overall the results are striking and all three figures demonstrate the same pattern. YACA significantly outperforms *k*-means-itemsets which in turn outperforms *k*means-items. On average, for the two-user datasets, YACA outperforms *k*means-itemsets by 25% and *k*means-items by 81%. For the three-user datasets, YACA outperforms *k*means-itemsets by 40% and *k*means-items by 125% and for the four-user datasets, , YACA outperforms *k*means-itemsets by 133% and *k*means-items by 274%. These numbers are highly significant and suggest that pattern-based clustering techniques may be a natural approach to cluster consumer transactions such as Web transactions considered in this paper. We also ran experiments for alternate definitions of "pure" clusters (e.g. "80% pure"), and the results are similar. For lack of space, we do not report all the results here.

In addition to the quantitative results, there are several examples of interesting clusters discovered in the data. For example:

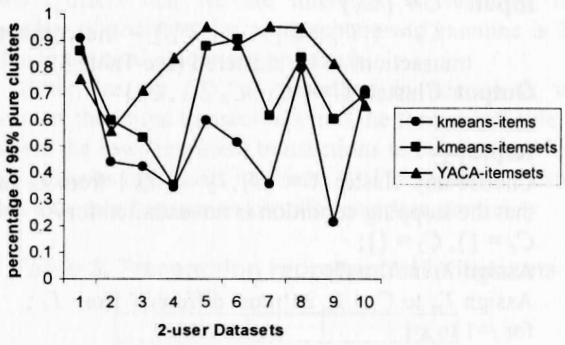


Figure 3. Results for 10 two-user datasets

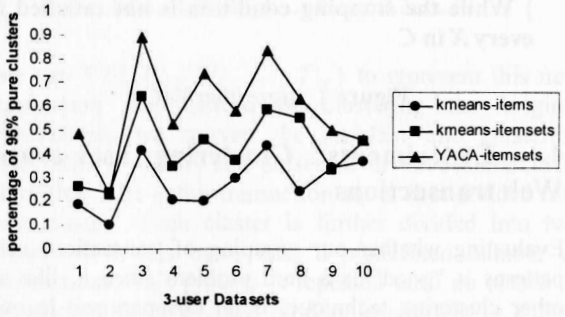


Figure 4. Results for 10 three-user datasets

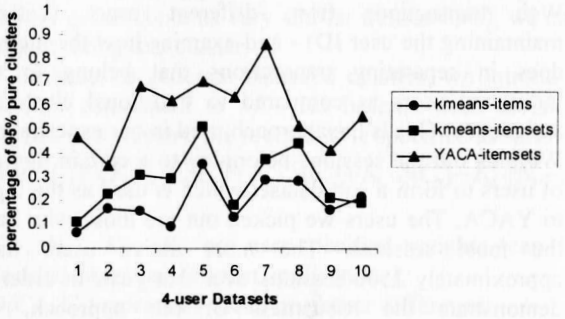


Figure 5. Results for 10 four-user datasets

- Two significant itemsets in a cluster were {*starting_day = Saturday, most_visited_category = sports*} and {*starting_day = Sunday, most_visited_category = services such as chat rooms*} – reflecting a weekend behavior pattern of one of the users in the data.
- A highly significant itemset in a cluster was {*start category = search, most visited category = retail*} indicating shopping patterns driven by Web searches rather than going to favorite sites to make purchases.

Home Depot
Ex. 1006 - Page 17

- Two most significant itemsets in one of the clusters were: (1) {*first_category = ISP, news_category = 1*} with support 71.3% - in contrast, the support for this itemset over the entire dataset was just 3.18%; (2) {*average # of sites per category = 1, first_site = site187, retail category = 1*} with support 51.7% - again, in contrast the support over the entire dataset was just 5%. Both these itemsets are reflective of a behavior that is more complex - indicating a more focused browsing behavior perhaps - than the first two examples above.

In general, the itemsets in some of the clusters suggest highly explainable behavior patterns (as in the first two examples above), while in other cases it is more difficult to place a single behavior type (third example) based on the itemsets in the cluster. This may be in part due to the fact that consumer behavior in the real-world is highly complex. In such cases, the clusters generated can still be useful by providing a starting point for a deeper understanding of consumer behavior.

5. Related Work

Traditional clustering algorithms usually employ a distance metric based (e.g., euclidean) similarity measure in order to partition the data. As pointed out in [7], while the traditional distance-based clustering approaches can yield satisfactory results for numeric attributes, it is not appropriate for data sets with categorical attributes. In the recent past, there has been much work done on clustering large categorical databases (ROCK[7], [8], STIRR[6], LargeItem[12], CLOPE[13]). [7] presents a concept of clustering based on links between data points. They consider a pair of points to be neighbors if their similarity exceeds a certain threshold. The similarity value for pairs of points can be based on any non-metric similarity function. The number of links between a pair of points is then the number of common neighbors for the points. During clustering, they merge clusters/points with the most number of links. The drawback of this method is the heavy computational cost, and sampling has to be used when scaling to large datasets. Also, there may exist undesirable mergers because of the large number of pairs whose similarities barely exceed the threshold.

The idea of pattern-based clustering is not new. [11] considers two objects similar if they exhibit a coherent pattern on a subset of dimensions. The definition of a pattern is based on the correlation between attributes of objects to be clustered. The specific definition of a pattern used in [11] makes it more suitable for numerical data. This paper uses a set of itemsets as the representation of patterns in each cluster. Hence within the clustering methods dealing with categorical data, those using the concept of large items/itemsets and association rules are particularly relevant to this paper. [8] addresses the problem of clustering related customer transactions in a market basket database. Frequent itemsets used to generate association rules are used to construct a weighted hypergraph. Each frequent itemset is a hyperedge in the weighted hypergraph and the weight of the hyperedge is computed as the average of the confidences for all possible association rules that can be generated from the itemset. Then, a hypergraph partitioning algorithm from [9] is used to partition the items such that the sum of the weights of hyperedges that are cut due to the partitioning is minimized. The result is a clustering of items (not transactions) that occur together in the transactions. Finally, the item clusters are used as the description of the cluster and a scoring metric is used to assign customer transactions to the best item cluster. The approach in [8] makes the assumption that itemsets that define clusters are disjoint and have no overlap among them. This may not be true in practice since transactions in different clusters may have a few common items.

[12] introduces a clustering criterion suggesting that there should be many large items within a cluster and little overlapping of such items across clusters. They then use this criterion to search for a good clustering solution. [12] also points out that, for transaction data, methods using pairwise similarity, such as *k*-means, have problems in forming a meaningful cluster. For transactions that come naturally in collection of items, it is more meaningful to use item/rule based methods. [13] addresses a similar problem and our work is most similar to LargeItem[12] and CLOPE[13] in that we do not use any pairwise distance function. We define a global goal and use this goal to guide the clustering process. Compared to LargeItem[12] and CLOPE[13], our method takes a new perspective of associating itemsets with behavior patterns, and using that concept to guide the clustering process. Using this approach, we are able to identify the distinguishing itemsets representing a cluster of transactions. As noted previously, in this paper behavioral patterns describing a cluster are represented by a *set* of itemsets (for example, a set of two itemsets {*weekend, second site = eonline.com*} and {*weekday, second site = cnbc.com*}. We allow the possibility to find a *set of itemsets* to describe a cluster instead of just a set of items which is the focus of other item/itemsets related work. This is important for clustering Web transaction data. We also illustrate in the paper how we cluster Web transaction data with both numeric data and categorical data. On a different note, document clustering methods also use the concept of *terms* (similar to items) for clustering. However, document clustering tends to have their special treatment for documents and we do not discuss them here. See [14] for some common

Home Depot
Ex. 1006 - Page 18

approaches in document clustering, and see [3] for a recent approach utilizing the concept of term set.

Finally, another important category of clustering algorithms is model-based clustering [4, 5]. The data is viewed as coming from a mixture of probability distributions, each representing a different cluster. If we view the patterns representing a cluster as a type of model, the spirit of our approach is similar to that of model-based clustering. Model-based methods can provide good cluster results if the data does approximate the assumed distribution, but it can simplify the differences between clusters and ignore other attributes not in the hypothesized distribution and also ignores patterns across attributes.

6. Discussion

As mentioned in the introduction, the existence of natural categories of customer behavior is intuitive, and these categories influence the transactions observed. We suggest that pattern-based clustering techniques, such as the one described in this paper, may be effective in learning such natural categories and can, hence, enable firms to understand their customers better and build more accurate customer models. In this paper we presented a new approach, YACA, for pattern-based clustering of Web transactions and demonstrated that the technique performs highly effectively as compared to traditional techniques.

There are limitations of this work, which suggest opportunities for future work. First, the choice of pattern representation for a domain and the subsequent difference and similarity definitions that need to be derived, continue to be subjective. While we argued that itemsets are good for Web transactions, it is harder to claim that they are the "best" representation technique. Similarly the difference and similarity definitions are intuitive but there may be alternatives. The algorithm proposed, YACA, is a heuristic approach that incrementally and hierarchically clusters the transactions and in future work we will investigate means to improve the efficiency. Overall, the results in our experiments suggest that pattern-based clustering can be highly effective, and we hope to investigate approaches to extend the solutions along the lines described above in future work.

References

[1] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. and Verkamo, A. I., "Fast Discovery of Association Rules", Advances in Know. Discovery and Data Mining, Chapter 12, AAAI/MIT Press, 1995.

[2] Allenby, G. M. and Rossi, P. E., "Marketing Models of Consumer Heterogeneity", Journal of Econometrics 89(1999) 57-78.

[3] Beil, F., Eater, M. and Xu, X., "Frequent Term-Based Text Clustering", Proc. 8th Int. Conf. on Knowledge Discovery and Data Mining (KDD '2002), Edmonton, Alberta, Canada, 2002.

[4] Fraley, C. and Raftery, A. E., "How many clusters? Which clustering method? - Answers via Model-Based Cluster Analysis". Technical Report no. 329, Department of Statistics, University of Washington, Computer Journal 41:578-588 (1998).

[5] Fraley, C. and Raftery, A. E., "Model-Based Clustering, Discriminant Analysis, and Density Estimation", Tech Rep. 380, Department of Statistics, University of Washington, Seattle, WA, Oct. 2000. Journal of the American Statistical Association 97:611-631 (2002).

[6] Gibson, D. , Kleinberg, J. , Raghavan, P. , "Clustering Categorical Data: An Approach Based on Dynamical Systems", the VLDB Conference, New York City, New York, August 1998.

[7] Guha, S. , Rastogi, R., Shim K. , "ROCK: A Clustering Algorithm for Categorical Attributes", the 15th International Conference on IEEE Data Engineering, Sydney, Australia, 1999.

[8] Han, E., Karypis, G., Kumar, V. and Mobasher, B., "Clustering based on association rule hypergraphs", in Proceedings of the SIGMOD'97 Workshop on Research Issues in Data Mining and Knowledge Discovery. 1997, ACM.

[9] Karypis, G., Aggarwal, R., Kumar, V. and Shekhar, S., "Multilevel hypergraph partitioning: application in VLSI domain". In Proceedings of the ACM/IEEE Design Automation Conference, Canada (1997).

[10] Kimbrough, S., Padmanabhan, B. and Zheng, Z., "On Usage Metric for Determining Authoritative Sites", Procs. Workshop on Information Technology and Systems WITS 2000, December 2000.

[11] Wang, H., Yang, J., Wang, W., and Yu, P.S., "Clustering by Pattern Similarity in Large Data Sets", Proc. ACM SIGMOD Conference, Madison, WI, June 2002.

[12] Wang, K., Xu, C. and Liu, B. "Clustering Transactions Using Large Items", Proc. 8th Int. Conf. on Information and Knowledge Management (ACM CIKM'99), Kansas City, November, 1999.

[13] Yang, Y., Guan, X., You, J., "CLOPE: A Fast and Effective Clustering Algorithm for Transactional Data", SIGKDD '02, July 23-26, 2002, Edmonton, Alberta, Canada.

[14] Zhao, Y. and Karypis, G., "Criterion functions for document clustering: experiments and analysis", Tech. Report #01-40, Department of Comp. Sci. & Eng., U. Minnesota, 2002.

Home Depot
Ex. 1006 - Page 19