

Cost-Sensitive-Data Preprocessing for Mining Customer Relationship Management Databases

Junfeng Pan and Qiang Yang, *Hong Kong University of Science and Technology*

Yiming Yang and Lei Li, *Sun Yat-sen University*

Frances Tianyi Li and George Wenmin Li, *Guangzhou E-DM Tech Corporation*

A staged preprocessing framework for cost-sensitive data processing can help service providers identify customers who might switch to a competitor.

With worldwide industry deregulation, telecommunications and financial services customers face an ever-growing number of choices. So, more customers are switching their service providers. This phenomenon is called customer churning or attrition and is a costly problem for these industries. For example, *The Walker Loyalty*

Report for Software and Hardware (www.walkerinfo.com/what/loyaltyreports) estimates that the US cellular telephone market experiences a 30-percent churn rate. Lost customers are difficult to get back. The average cost to win back a customer is hundreds of dollars, if not more, whereas retaining a customer costs only tens of dollars. These companies and institutions must therefore identify customers likely to churn and formulate plans to combat the problem.

Customer retention data is cost sensitive. For example, if you predict a valuable customer as loyal but that customer churns, the cost is usually higher than if you classify a loyal customer as one who will churn. The situation in direct marketing is just the opposite: it costs more to classify a willing customer as a reluctant one. These cases imply that the classification problem is cost-sensitive in nature. Customer retention data sets are also often imbalanced, in that

the customers who churn are a small fraction of all customers. Similarly, customers who buy a certain product after a marketing campaign are usually a small fraction of all the customers. So, this type of data set contains a small fraction of positive data and a large proportion of negative data.

Many researchers have tackled the direct-marketing problem as a classification problem.¹⁻⁴ Researchers have also recognized the importance of cost-sensitive data in the direct-marketing and customer-retention domains.^{2,5} However, they often account for cost information only in the classification-result evaluation stage, not in the data-preprocessing stage. We've developed a staged framework for data preprocessing to support data mining. Our framework pushes the cost sensitivity and data imbalance of customer retention data into the data preprocessing itself.

When we tested our framework on the data set from the ACM KDD Cup 1998, it outperformed the

winner of that data mining and knowledge discovery competition. We've also incorporated our framework into a software system, ED-Money. To demonstrate our framework's ability to predict customer attrition with high accuracy, we've applied it to some benchmark data and to a real customer attrition data set from a large Chinese mobile telecommunications company.

Cost-sensitive-data preprocessing

The ACM KDD Cup 1998 (www.kdnuggets.com/datasets/kddcup.html) focused on a direct-marketing problem. In this problem, we regard the fund-raising center as a *company* and the donors as *customers*. We aim to build a classifier to predict whether a customer will respond and how much money the company might earn from a responding customer. The company wishes to maximize the net profit by evaluating a customer's value and determining whether it should contact the customer by mail. We calculate the net profit as the difference between earned money and the cost of mailing: $\Sigma(\text{earned money} - \$0.68)$. The earned money is the money collected from a customer, and \$0.68 is the mailing cost.

We face three challenges in working with the ACM KDD Cup 1998 data set:

- It's difficult to know ahead of time what type of data set we're dealing with. Are we working with a classification problem that's inherently nonlinear? If so, we must design a corresponding classifier for the prediction task.
- Because the problem is cost sensitive, in that the data set contains both classification information and profit (amount) information, we must tackle it head-on.
- The classification results might be sensitive to the classifier's parameters, so we need methods to stabilize the models.

In data preprocessing's first phase, we let the user visualize the data set to gain intuitive insights on the data's distribution. The data set contains imbalanced data, with the positive data corresponding to only a small fraction of the total. So, we'd like to know whether a linear classifier such as a perceptron or linear-regression method would be sufficient to separate the two classes of data.

To answer this question, we apply the *self-organizing-mapping* model to automatically cluster the data from the set of positive and

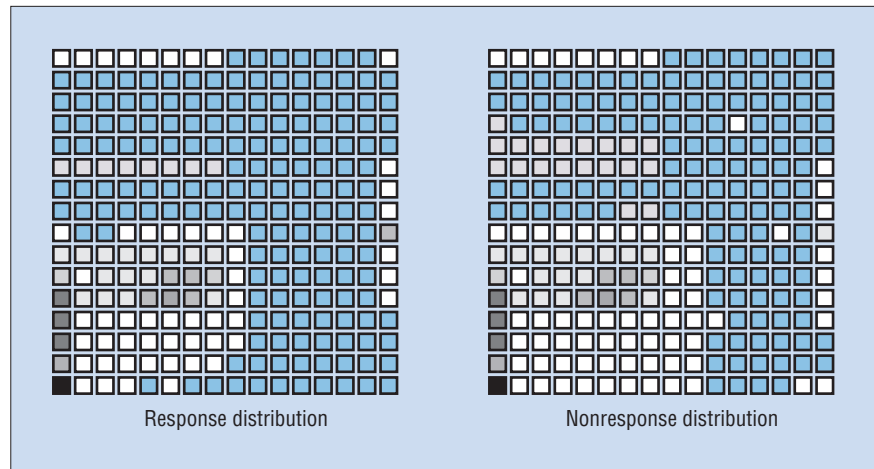


Figure 1. An illustration of mixed positive-class and negative-class distributions. The cell shade represents its density. The two distributions (response and nonresponse) overlap, indicating the problem's complexity.

negative data in the training data set separately. The SOM is a data-visualization tool that converts high-dimensional data items into simple geometric relationships on a low-dimensional display. In our case, we convert all training data to a 2D map. First, we define the similarity between data items according to their class information, and then we learn the weight vectors of mapping each data item onto a 16×16 -cell 2D grid. For each cell in the 2D map, the color represents the number of data items belonging to the cell. So, a heavy color indicates many data items in the cell. We repeat this map building for both positive data (the 5-percent response class) and negative data (the 95-percent nonresponse class), as figure 1 shows. The two subsets of the data have similar data distributions, with large areas of overlap, which tells us that the data are difficult to separate with a linear separating function. On the basis of this information, we decided to use a nonlinear classification method such as multilayer neural networks.

From attribute values to response ratios

The ACM KDD Cup 1998 training set has 95,412 records, with 479 nontarget attributes and two target attributes (respond or not, and amount of donation). As we mentioned before, the data distribution is imbalanced: only 5 percent of customers belong to the response class, while 95 percent don't respond. To solve this imbalance, we use the data distribution information to normalize the range of all attributes.

Our intuition is to push the response-

likelihood information into the data itself. To do this, we apply an equidepth algorithm to convert each attribute's values into ranges, where the maximum number of ranges is a tunable user-defined parameter. In each range, we count the number of response-class and nonresponse-class customers and calculate the *response ratio*. The ratio reflects how likely a customer will respond given this value. Finally, our preprocessing module normalizes the attribute's values by converting all attribute values so that they fall into the range of 0 to 1.

Table 1 shows the response ratio for date-of-birth (DOB) attribute. The table lists 10 intervals, corresponding to each index value in the first column. The second column lists the upper bound in each interval. Next are the total number of instances in each interval and the total number of response and nonresponse instances in the interval. Finally, the last column shows the computed response ratio, which corresponds to the transformed values for the new attribute in the place of the DOB attribute. For example, for that attribute, the response ratio for transformed values in the 0 to 1 range corresponding to value ≤ 0.000 is 0.048772. This corresponds to the first record in table 1.

More precisely, according to the table, we could work out that the minimal response ratio is 0.039054 and the maximal response ratio is 0.060215. Knowing these ratios, we stretch linearly all the ratios into range [0, 1] before inputting them into a the backpropagation (BP) neural network, using the following formula to compute the stretched

Table 1. Division and response ratio of the date-of-birth attribute.

Index	Value upper bound	Total count	Response count	Nonresponse count	Response ratio
0	0.000000	23,661	1,154	22,507	0.048772
1	1308.000000	5,710	223	5,487	0.039054
2	1909.000000	7,363	405	6,958	0.055005
3	2401.000000	7,440	448	6,992	0.060215
4	2905.000000	7,232	403	6,829	0.055725
5	3501.000000	7,494	408	7,086	0.054444
6	4101.000000	7,181	373	6,808	0.051943
7	4701.000000	7,807	437	7,370	0.055975
8	5201.000000	7,233	366	6,867	0.050601
9	5809.000000	6,951	332	6,619	0.047763
10	9710.000000	7,340	294	7,046	0.040054
Total		95,412	4,843	90,569	0.050759

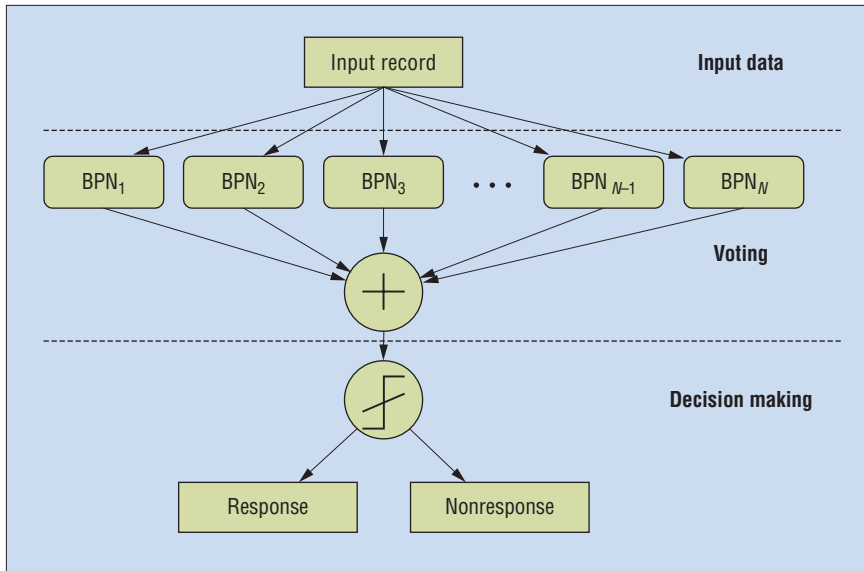


Figure 2. Decision making by voting, where the majority vote is taken as the final prediction.

value, which is the response ratio. Let the input response ratio be $R1$, the minimal response ratio be $R2$, the maximal response ratio be $R3$, and the stretched value be S . We then have $S = (R1 - R2)/(R3 - R2)$.

In general, the larger the response ratio of an attribute’s value, the more likely the corresponding customer will respond. Almost all the ratios range from 4 to 6 percent (near 5 percent). So, no obvious feature exists that an individual attribute could use alone to determine a customer’s response. However, the data set is now cleaner and, as we show, can produce better classification results.

Cost-sensitive-data sampling

We next use the cost matrix to sample the data. Our goal is to produce a more balanced data set with more positive data than is in the original data set. Our sampling criterion is based on each record’s profit. Each mail incurs \$0.68 in cost. So, if we don’t win over a customer, we lose this amount. However, if we fail to respond to a potential customer, we lose even more. We can use this cost information to decide how to redistribute the positive and negative data.

In data sampling, a data set consists of three subsets:

- a training data subset for training the model,
- a validation data subset for evaluating different parameters of the model and selecting the best parameters, which can be used to provide a termination condition for the training process, and
- a testing subset for testing the model and producing a final evaluation result using the best parameter set.

The testing data subset is part of the ACM KDD Cup 1998 data set for calculating the final profit values and comparing the methods. We first randomly split the input data set into model training and validation subsets with a 2:1 ratio, giving us 67 percent for the training subset and 33 percent for the validation subset. While training, we save the weights learned for the BP network if the profit from the validation data set is higher. Otherwise, we restore the original weights. The training process stops if it fails to increase the net profit for several steps in a row.

To solve the cost-sensitive-data problem, we process the training subset by considering not only the samples as positive or negative but also each customer’s actual profit. From the initial training data subset, we duplicate the positive samples according to their profit: $(\text{earned money} - \$0.68)/\0.68 . Because many negative samples exist, we don’t duplicate them. Their profit is $-\$0.68$.

We regard $(\text{earned money} - \$0.68)/\0.68 as each positive sample’s weight. For example, for a positive-response customer with a \$10 profit, we would duplicate the sample $(\$10 - \$0.68)/\$0.68 = 14$ times. By doing this, we would assume that if we successfully find a responsive customer, we collect \$0.68 on average.

Building a neural network ensemble

We use a BP neural network as our base classifier. In training the classifier, we set several parameters such as the number of layers, number of units in each layer, and learning-rate value, η . We determine these parameters by evaluating the experimental results in the training subset against the testing subset. We observed that the profit values change greatly with the different samples. So, to permit stable models, we apply an ensemble of classifiers.⁶ We train the neural network N times, obtaining N groups of weights. Each group corresponds to a model called $BPN_1, BPN_2, BPN_3, \dots, BPN_N$.

When applying the ensemble of classifiers

to each record in the validating subset, each of the N models decides whether to mail out to this customer. So, each decision corresponds to a yes or no vote. We consider the majority vote as the final vote, as figure 2 shows.

Figure 3 shows how we've incorporated the preprocessing and classification framework into a system called ED-Money. The graph's three curves show the profit changing during the training. The red curve gives a quick profit evaluation of the training subset. It doesn't compute the whole subset but randomly samples 1,000 records. The evaluation results demonstrate the error between the training and testing subsets. We don't use the validation subset in this stage, because we have no way of knowing the actual data class and final amount of profit in that subset. In the ACM KDD Cup 1998 data set, the competition organizers used the testing subset later to determine the winner. We use this same methodology.

The blue curve indicates a full profit evaluation of the entire testing subset. We estimate the BP network's performance for the testing subset. The green curve evaluates profit of the validation subset by sampling 1,000 records. We use this estimate only to show the error between the test and validation subsets. We don't use this value for training. The edit box on the right of figure 3 shows the actual profit gained in the validation subset as \$14,247.00 after training.

Application to the KDD Cup 1998 data set

Figure 4 shows some details and results for our experiments on the KDD Cup 1998 data set. These experiments used a two-layer neural network. We experimented with BP networks with zero, one, and two hidden layers with a different learning rate, η . The best result is \$15,154.10, with an average of \$14,867.50 over a total of seven experiments. Table 2 compares results for our algorithm with KDD Cup 1998 participant results. Our average result tops those of all of the KDD Cup 1998 participants. Even though training is time consuming, once the system learns the model, the application process is very efficient.

Application to a mobile telecommunication data set

With China's recent successful entrance into the World Trade Organization, foreign

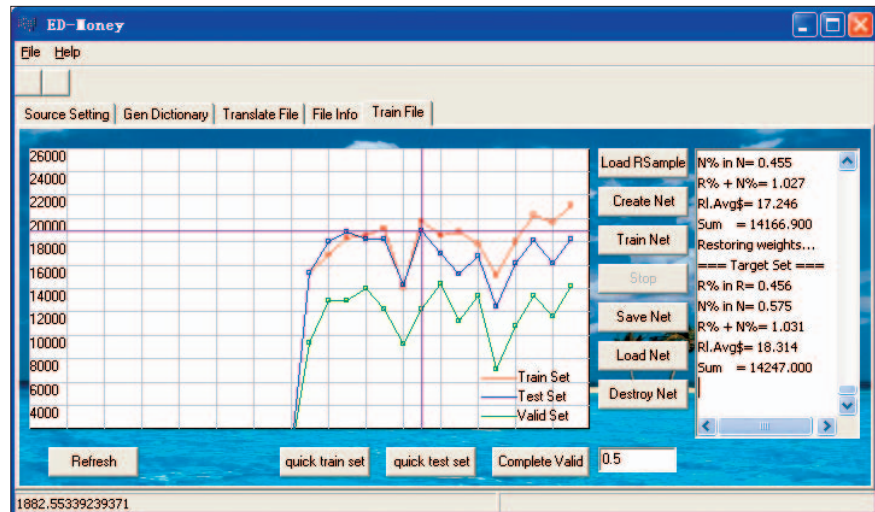


Figure 3. The test program's user interface.

Ratio of training data to testing data	2:1
Backpropagation network	
Input layer	450 units
Hidden layer	Layer 1: 16 units Layer 2: 8 units
Output layer	1 unit
Learning rate	0.10 minutes
Number in training set	6 (voters)
Average training time	14 minutes
Total training time	14 * 6 = 84 minutes
Results	
Net profit for each experiment	1. \$14,876.00 2. \$14,964.90 3. \$15,154.10 4. \$15,004.20 5. \$15,068.40 6. \$14,720.60 7. \$14,284.10
Average profit	\$14,867.50
Best profit	\$15,154.10
Worst profit	\$14,284.10
Standard deviation	\$292.30

Figure 4. Details and results for seven experiments on the framework.

enterprises have had more opportunities for trade and investment. At the same time, competition is severe between domestic corporations that haven't been exposed to an international business environment. However, as a result of the increasingly cutthroat competition, local companies are providing higher-quality products and services at lower prices. As this trend continues, a significant side effect also takes root: the loss of customer loyalty. This and the hostility among competitors concern many enterprises in China today. They're realizing that they need intelligent marketing tools to study customer

behavior, increase their customer base, exploit customer values, and minimize customer churning.

The algorithms we've described have been developed into a system we call ED-Money, which lets companies accurately predict which customers will likely churn, helping them to make business and marketing actions to stop customer churning. Like direct mailing, this is an example of cost-sensitive learning, where the costs of false positives and false negatives differ.^{1,2,4,5} In this application, failing to recognize a churning customer will cost the company the customer's lifetime

Table 2. A comparison of our framework to ACM KDD Cup 1998 competitors.

Method	Net profit
Our framework	\$14,867 (average)
GainSmarts	\$14,712
SAS	\$14,662
Quadstone	\$13,954
CARRL	\$13,825
Amdocs	\$13,794

value. Conversely, if we correctly identify a customer as a churning customer, we can spend a small amount of money to keep the customer, gaining the company a positive profit. However, if we wrongly classify a nonchurning customer as churning, we'll also lose some money on the direct-marketing effort. We can encode this cost information in a cost matrix.

We're working with a large Chinese telecommunications company to identify solutions for its customer attrition problem. The customer data consists of 13,978 records, with two classes: "lost = yes" and "lost = no." About 32 percent of the customers belong to the "yes" class. So, most customers aren't churning. This is similar to the situation in the ACM KDD Cup 1998 data set, with an inverted objective function: instead of maximizing net profit, in attrition detection we wish to minimize net cost.

We split the data set into training, validation, and testing subsets. The ratio between training (which includes validation) and testing is 2:1. There are 25 nontarget attributes and one target attribute. In one of the tests, we set up the cost matrix so that the false-positive cost is -\$1.00, the true-positive cost is -\$5.00, and the rest is zero cost. This matrix corresponds to the false-negative situation (when we predict a churning customer to be nonchurning), in which we lose a great deal in terms of the customer's lifetime value (designated by the \$5.00 loss). However, if we wrongly predict a nonchurning customer to be churning (a false positive), we'll waste \$1.00 in marketing costs. Table 3 shows this matrix. In this application, our goal is to reduce the total cost as much as possible by determining the potentially churning customers.

Like the ACM KDD Cup 1998 data set, this data set isn't linearly separable. We used a collection of six neural networks in the ensemble, each consisting of four layers. The training time is 33 seconds total. Table 4 compares the results for ED-Money and a C4.5 decision

Table 3. A cost matrix for attrition detection.

Did the customer actually respond?	Prediction of customer response	
	Yes	No
Yes	0 * true positive	-5 * false negative
No	-1 * false positive	0 * true negative

Table 4. A comparison of ED-Money and decision tree predictions, showing the average results from seven experiments.

Confusion matrix	ED-Money		C4.5 decision tree	
	Prediction of customer response		Prediction of customer response	
	Yes	No	Yes	No
Customer responded (Yes)	1,216	280	444	1,052
Customer didn't respond (No)	1,147	2,088	73	3,162
Accuracy	69.8%		76.4%	
Net loss	-\$2,547.00		-\$5,333.00	

tree.⁷ We contrast our results with two baseline cases:

- *Case 1* is the baseline result when we classify all customers in the validation data set as nonchurning customers. In this case, the baseline net cost is -\$7,480.00.
- *Case 2* is the second baseline result when we consider all customers as churning customers with a simple rule "lost = yes." The net loss in this case is -\$3,235.00.

As the table shows, ED-Money performs much better than the decision tree and is above both baselines by a large margin.

By applying ED-Money in the telecommunications company, we can identify customers who are potentially valuable but also likely to leave. New marketing campaigns target these customers, with satisfying results. We have also applied ED-Money to China's real estate market to identify potentially valuable customers from a large pool of all customers. Again, this allows companies target these customers using special deals. In the long term, we hope to quantify the real gain of using ED-Money using a control group of customers.

We'd like to extend our preprocessing and classification techniques to other areas, such as deciding what to do once we identify a customer as likely to churn. We're also investigating automatic methods for studying the

nature of data in place of visualizing the SOM. ■

Acknowledgments

Junfeng Pan and Qiang Yang thank Hong Kong RGC grant HKUST 6187/04E and 621606.

References

1. Q. Yang et al., "Extracting Actionable Knowledge from Decision Trees," *IEEE Trans. Knowledge and Data Eng.* (IEEE TKDE), vol. 19, no. 1, Jan. 2007, pp. 43-56.
2. C. X. Ling and Chenghui Li, "Data Mining for Direct Marketing: Problems and Solutions," *Proc. 4th Int'l Conf. Knowledge Discovery and Data Mining* (ACM KDD 1998), AAAI Press, 1998, pp. 73-79.
3. K. Wang et al., "Mining Customer Value: From Association Rules to Direct Marketing," *Int'l J. Data Mining and Knowledge Discovery* (DMKD J.), vol. 11, no. 1, July 2005, pp. 57-80
4. B. Zadrozny and C. Elkan, "Learning and Making Decisions When Costs and Probabilities Are Both Unknown," *Proc. 7th ACM Sigkdd Conf. Knowledge Discovery in Data* (KDD 01), ACM Press, 2001, pp. 204-213.
5. P. Domingos, "MetaCost: A General Method for Making Classifiers Cost Sensitive," *Proc. 5th ACM Sigkdd Conf. Knowledge Discovery and Data Mining* (KDD 99), ACM Press, 1999, pp. 155-164.

The Authors



Junfeng Pan is a PhD student in computer science and engineering at Hong Kong University of Science and Technology. He's working on data mining in Web and wireless applications. Contact him at Dept. of Computer Science and Engineering, Hong Kong Univ. of Science and Technology, Clearwater Bay, Kowloon, Hong Kong; panjf@cse.ust.hk; www.cse.ust.hk/~panjf.



Qiang Yang is a faculty member at the Hong Kong University of Science and Technology. His research interests include data mining, planning, case-based reasoning, and machine learning, and their application to Web, bioinformatics, and business intelligence problems. He received his PhD in computer science from the University of Maryland, College Park. He's a member of the AAAI, the ACM, and is a senior member of the IEEE Computer Society. Contact him at the Dept. of Computer Science and Engineering, Hong Kong Univ. of Science and Technology, Clearwater Bay, Kowloon, Hong Kong, China; qyang@cse.ust.hk; www.cse.ust.hk/~qyang



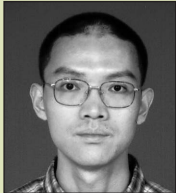
Yiming Yang is a director at the Guangdong Telecom Academy of Science and Technology and a PhD student at the Software Institution at Zhongshan University. His research interests include data warehouses, data mining and its applications, database management systems, and knowledge-based information systems. He received his MSc in automatic control from Huazhong University of Science and Technology. Contact him at the Dept. of Computer Science, Zhongshan Univ., Guangzhou, China; mcp001w@taurus.zsu.edu.cn.



Lei Li is a professor at and the director of the Software Research Institute of Sun Yat-Sen University, where he also serves as the director of graduate studies. His research interests focus on databases, logic programming, and software engineering. He received his PhD in computer science from Claude Bernard Lyon 1 University. Contact him at the Dept. of Computer Science, Zhongshan Univ., Guangzhou, China; lncsri07@cs.zsu.edu.cn.



Frances Tianyi Li is the chief technology officer of Guangzhou E-DM Tech. His research interests include data mining, planning, case-based reasoning, and machine learning, and their application to the Web and telecom industry. He received his MSc from Simon Fraser University. Contact him at Guangzhou E-DM Tech., Guangzhou City, China; ian_lee@gze-dm.com.



George Wenmin Li is the chief executive officer of Guangzhou E-DM Tech. His research interests include data mining and its application to telecom data analysis, logistics, and business intelligence problems. He received his master's degree from Simon Fraser University. Contact him at Guangzhou E-DM Tech., Guangzhou City, China; georgeli@gze-dm.com.

6. R.A. Jacobs et al., "Adaptive Mixtures of Local Experts," *Neural Computation*, vol. 3, no. 1, 1991, pp. 79–87.

7. T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps," *Bio-*

logical Cybernetics, vol. 43, no. 1, 1982, pp. 59–69.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.

THE IEEE'S
1ST ONLINE-ONLY MAGAZINE



IEEE Distributed Systems Online

brings you peer-reviewed articles, detailed tutorials, expert-managed topic areas, and diverse departments covering the latest news and developments in this fast-growing field.

Log on for **free access**

to such topic areas as

- Grid Computing
- Middleware
- Cluster Computing
- Security
- Peer-to-Peer
- Operating Systems
- Web Systems
- Parallel Processing
- Mobile & Pervasive and More!

To receive monthly updates, email

dsonline@computer.org

<http://dsonline.computer.org>