An Empirical Study of Categorical Dataset Visualization Using a Simulated Bee Colony Clustering Algorithm

James D. McCaffrey

Microsoft MSDN / Volt VTE
One Microsoft Way
Redmond, WA 98052 USA
v-jammc@microsoft.com

Abstract. This study investigates the use of a biologically inspired meta-heuristic algorithm to cluster categorical datasets so that the data can be presented in a useful visual form. A computer program which implemented the algorithm was executed against a benchmark dataset of voting records and produced better results, in terms of cluster accuracy, than all known published studies. Compared to alternative clustering and visualization approaches, the categorical dataset clustering with a simulated bee colony (CDC-SBC) algorithm has the advantage of allowing arbitrarily large datasets to be analyzed. The primary disadvantages of the CDC-SBC algorithm for dataset clustering and visualization are that the approach requires a relatively large number of input parameters, and that the approach does not guarantee convergence to an optimal solution. The results of this study suggest that using the CDC-SBC approach for categorical data visualization may be both practical and useful in certain scenarios.

Keywords: Categorical data, category utility, cluster analysis, data visualization, simulated bee colony algorithm.

1 Introduction

This paper presents a study of the use of a biologically inspired meta-heuristic algorithm for processing large datasets composed of categorical data in order to present the data in a useful visual form. The analysis and visualization of datasets which contain categorical data has great practical importance. Examples include examining sales data to forecast consumer purchasing behavior, examining telecommunications data for possible terror-related activity, and examining medical information for various clinical diagnoses. For the sake of concreteness, consider the artificial dataset presented below. The nine tuples in the dataset are based on three attributes: color, size, and temperature. Each of the three attributes can take on a single categorical value: red, blue, green or yellow; small, medium, or large; and hot or cold, respectively. With even this unrealistically small dataset, it is quite difficult for human observers to categorize or group the raw dataset in a meaningful way so that the categorized data can then be presented in some visually descriptive form.

G. Bebis et al. (Eds.): ISVC 2009, Part I, LNCS 5875, pp. 179–188, 2009.

```
001:  Green   Medium  Hot
002:  Blue    Small   Hot
003:  Red     Large   Cold
004:  Red     Medium  Cold
005:  Yellow  Medium  Hot
006:  Green   Medium  Hot
007:  Red     Small   Hot
008:  Red     Large   Cold
009:  Blue    Medium  Hot
```

The primary source of the difficulty of clustering the data presented above is the fact that the attribute values are categorical rather than numerical. It is not so obvious how to compute a meaningful difference or a representative value for categorical tuples such as (red, small, hot) and (blue, large, cold). Data clustering is a widely studied problem domain. A search of the IEEE digital library Web site for the keyword "clustering" returned a list of references to over 36,000 documents

In order to measure the quality of a particular clustering algorithm compared to alternative approaches, some measure of clustering effectiveness must be employed. One approach for evaluating the quality of a clustering algorithm which works on categorical data is to generate a synthetic dataset which is based on some hidden, underlying rule set, run the proposed clustering algorithm against the synthetic dataset, and then gauge the quality of the resulting clusters using some form of similarity or likelihood measure. Examples of similarity measures which can be used to evaluate clustering effectiveness include the Simple Matching coefficient, Jaccard's coefficient, Dice's coefficient, the Cosine coefficient, and the Overlap coefficient [1]. Examples of likelihood measures which can be used to evaluate clustering effectiveness include various forms of entropy functions and the category utility function [2]. The category utility (CU) function is generally attributed to a 1985 paper by Gluck and Corter [3]. The CU function is defined in terms of the bivariate distributions produced by a clustering. Suppose a dataset is composed of t tuples where each tuple is based on A_i attributes (i = 1... m) and where each attribute value, V_{ij}, is a categorical value. If a dataset under analysis is partitioned into a cluster set C = $\{C_k\}$ (k = 1... n), then the category utility function for the clustering scheme is given by the equation:

$$CU(C) = \frac{1}{n}\sum_{k=1}^{n} P(C_k)\left[\sum_i \sum_j P\left(A_i = V_{ij} \mid C_k\right)^2 - \sum_i \sum_j P(A_i = V_{ij})^2\right] \quad (1)$$

The left-hand term in the brackets of equation (1) represents conditional probabilities that each attribute takes on a particular categorical value, given the distribution of that value within a cluster. The right-hand term is similar except that it represents unconditional probabilities of attribute values for the entire dataset. Therefore, the entire term in the square brackets in equation (1) measures the difference of the probabilities of finding attribute values in a cluster purely by chance and the probabilities of finding those values given the clustering scheme.

The fact that quality of categorical data clustering algorithms can be evaluated using the CU function raises the possibility of using the CU function as the basis of a clustering generation mechanism. This is the foundation of the approach used by the algorithm introduced in this study. Except in situations with very small datasets, the

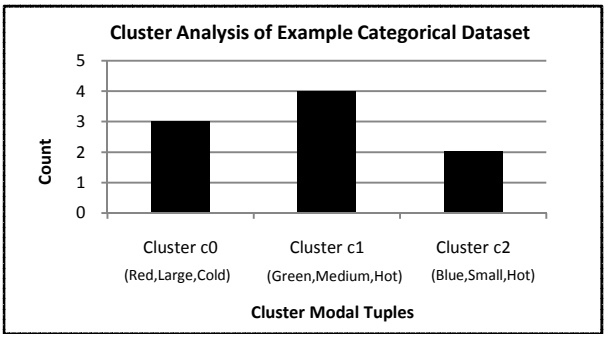

Fig. 1. Visualization of categorical data after clustering

CU function cannot be used to directly generate an optimal clustering of categorical data. Complete analysis of the effectiveness of any clustering algorithm is an NP-complete problem and requires a full enumeration of all possible partitions of the dataset under analysis [2]. Therefore an indirect approach must be employed. The approach introduced by this study is to use a simulated bee colony algorithm (SBC) in conjunction with the category utility function. As will be explained in the following sections of this paper, in essence, the simulated bee colony algorithm intelligently searches the entire solution space of all possible dataset partitions, seeking the partition which has a global maximum category utility value. This algorithm is called CDC-SBC (Categorical Dataset Clustering with a Simulated Bee Colony) to distinguish it from other algorithms in the literature. The resulting clustered categorical data can be visually represented in useful ways, such as the one shown in Fig. 1 which uses modal values from each cluster. The integral relationship between the visualization of categorical datasets and clustering is formalized in a widely cited framework for visualization techniques called the Information Visualization Data State Reference Model (IVDSRM) [4]. The IVDSRM framework is summarized in Fig. 2.

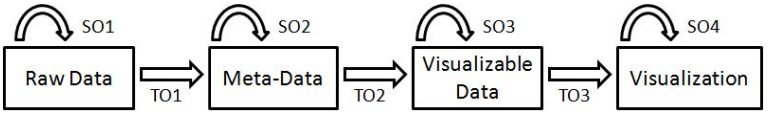

Fig. 2. The Information Visualization Data State Reference Model

The IVDSRM visualization framework models the creation of a visualization as four distinct data stages: raw data (the value stage), meta-data (the analytical abstraction stage), visualizable data (the visualization abstraction stage), and visualization (the view stage). Two types of operators can be applied to each data stage: transformation operators (TO1 through TO3), which create a new data stage, and stage operators (SO1 through SO4) which do not change the underlying structure of data. In the context of the IVDSRM framework, the clustering technique presented in this paper is a Transformation Operator 1 and produces clustering meta-data from the raw categorical dataset data. The clustering meta-data can then be used to produce several different visualizations such as the modal histogram shown in Fig. 1.

2 Algorithms Inspired by Bee Behavior

Algorithms inspired by the behavior of natural systems have been studied for decades. Examples include algorithms inspired by ants, biological immune systems, metallurgic annealing, and genetic recombination. A review of the literature on algorithms inspired by bee behavior suggests that the topic is evolving and that there is no consensus on a single descriptive title for meta-heuristics based on bee behavior. Algorithm names in the literature include Bee System, BeeHive, Virtual Bee Algorithm, Bee Swarm Optimization, Bee Colony Optimization, Artificial Bee Colony, Bees Algorithm, and Simulated Bee Colony.

Common honey bees such as Apis mellifera take on different roles within their colony over time [5]. A typical hive may have 5,000 to 20,000 individuals. Young bees (2 to 20 days old) nurse larvae, construct and repair the hive, guard the entrance to the hive, and so on. Mature bees (20 to 40 days old) typically become foragers. Foraging bees typically occupy one of three roles: active forgers, scout foragers, and inactive foragers. Active foraging bees travel to a food source, gather food, and return to the hive. Roughly 10% of foraging bees in a hive are employed as scouts.

A 1997 study by Sato and Hagiwara used a model of honey bee behavior named Bee System to create a variation of the genetic algorithm meta-heuristic [6]. The algorithm essentially added a model of the behavior of scout bees to introduce new potential solutions and avoid premature convergence to local minima solutions. A 2002 study by Lucic and Teodorvic used a variation of the Bee System model to investigate solving complex traffic and transportation problems [7]. The study successfully used Bee System to solve eight benchmark versions of the traveling salesman problem. A 2004 paper by Nakrani and Tovey presented a honey bee inspired algorithm for dynamic allocation of Internet services [8]. The study concluded that bee inspired algorithms outperformed deterministic greedy algorithms in some situations. A 2005 study by Drias et al. used a meta-heuristic named Bee Swarm Optimization to study instances of the Maximum Satisfiability problem [9]. The study concluded that Bee Swarm Optimization outperformed other evolutionary algorithms, in particular an ant colony algorithm. A 2006 paper by Basturk and Karaboga investigated a bee-inspired algorithm named Artificial Bee Colony to solve five multi-dimensional numerical problems [10]. The paper concluded that the performance of the bee algorithm was roughly comparable to solutions by differential evolution, particle swarm optimization, and evolutionary algorithms. A 2009 study by McCaffrey demonstrated that an algorithm named Simulated Bee Colony outperformed existing deterministic algorithms for generating pairwise test sets, for six out of seven benchmark problems [11].

3 Simulated Bee Colony Algorithm Implementation

There are many ways to map honey bee foraging behavior to a specific algorithm which clusters categorical data in order to create a useful visual presentation. The three primary design features which must be addressed are 1.) design of a problem-specific data structure that simulates a foraging bee's memory and which represents the location of a food source, which in turn represents a dataset clustering scheme, 2.) formulation of a problem-specific function which measures the goodness, or quality,

of a candidate partitioning, and 3.) specification of generic algorithm parameters such as the numbers of foraging, scout, and inactive bees in the colony, and the maximum number of times a bee will visit a particular food source. Suppose the dataset to be analyzed contains the data described in the Introduction section of this paper, with attributes of color (red, blue, green, yellow), size (small, medium, large), and temperature (hot, cold), and a cluster size of n = 3 is specified. The screenshots shown in Fig. 3 and Fig. 4 show the result of a sample program run and illustrate many of the implementation details.

```
C:\CDC-SBC\Run\bin\Debug> Run.exe

Begin cluster analysis of categorical data using SBC

Number clusters = 3

Initializing Hive

Number Active bees = 60
Number Inactive bees = 20
Number Scout bees = 20
Maximum number of cycles = 10,000
Maximum cycles without improvement = 10,000
Maximum visits to a food source = 10
Probability waggle dance will convince = 0.9000
Probability a bee accepts a worse source = 0.0100

Hive initialized
```

Fig. 3. Screenshot of initialization phase of the CDC-SBC implementation

The CDC-SBC algorithm implementation used in this study models a bee as an object with four data members. The primary data member is a two-dimensional integer array named MemoryMatrix which corresponds to a bee's memory of the location of a food source, which in turn represents a dataset clustering. A Status field identifies the bee's role (1 = an active forager). A CategoryUtility field is a value which is a measure of the quality of the memory matrix, as described in the Introduction section of this paper. A NumberVisits field is a counter that tracks the number of times the bee object has visited a particular food source. The honey bee colony as a whole is modeled as an array of bee objects. The CDC-SBC algorithm iterates through each bee in the colony and examines the current bee's Status field. If the current bee is an active forager, the algorithm simulates the action of the bee leaving the hive to go to the current food source in memory. Once there, the bee examines a single neighbor food source. A neighbor food source is one which, relative to the current food source, has a single tuple assigned to a different cluster. If the quality of the neighbor food source is superior to the current food source, the foraging bee's memory is updated with the neighbor location and the NumberVisits counter is reset to 0.

After examining a neighbor food source, an active bee returns to the hive. If the returning bee has reached a threshold for the maximum number of visits to its food source in memory, that bee becomes inactive and a randomly selected inactive bee is

```
All cycles completed

Best clustering matrix found is
0 0 1 1 0 0 0 1 0
1 0 0 0 1 1 0 0 1
0 1 0 0 0 0 1 0 0

Corresponding category utility is 0.3971

Cluster c0 =
003 ( Red     Large   Cold    )
004 ( Red     Medium  Cold    )
008 ( Red     Large   Cold    )
------------------------------
mode: Red     Large   Cold

Cluster c1 =
001 ( Green   Medium  Hot     )
005 ( Yellow  Medium  Hot     )
006 ( Green   Medium  Hot     )
009 ( Blue    Medium  Hot     )
------------------------------
mode: Green   Medium  Hot

Cluster c2 =
002 ( Blue    Small   Hot     )
007 ( Red     Small   Hot     )
------------------------------
mode: Blue    Small   Hot

End SBC visualization run
```

Fig. 4. Screenshot of execution and results of the CDC-SBC implementation

converted to an active forager. Otherwise the returning bee performs a simulated waggle dance to all inactive bees in the hive. This dance conveys the goodness of the current food source / clustering in the dancing bee's memory. Inactive bees with food sources in memory which have lower quality than the returning bee's food source will update their memories to the returning bee's memory with probability = 0.90. Scout bees are not affected by the waggle dances of returning foragers. Instead, scouts leave the hive, examine a randomly selected food source, return to the hive, and perform a waggle dance to the audience of currently inactive bees.

4 Results

Two common metrics for measuring the effectiveness of clustering algorithms are precision and recall [12]. Suppose some dataset contains t tuples and some clustering algorithm assigns each tuple to one of n clusters. Let a_i represent the number of tuples correctly assigned to cluster i. Let b_i represent the number of tuples which have been incorrectly assigned to cluster i. And let c_i represent the number of tuples which have been incorrectly rejected from cluster i (and incorrectly assigned to some cluster j

where $j \neq i$). Then the precision for cluster i is given by $p_i = a_i / (a_i + b_i)$. The recall for cluster i is given by $r_i = a_i / (a_i + c_i)$. The precision for a given cluster can be thought of as a measure of accuracy, and the recall can be thought of as a measure of completeness. The micro-precision of a clustering result is computed as a whole, across all clusters, using overall numbers of correctly assigned tuples, incorrectly assigned tuples, and incorrectly rejected tuples.

4.1 Experiment #1 – Congressional Voting Data

In order to evaluate the effectiveness of the CDC-SBC algorithm and resulting data visualizations compared to alternative clustering algorithms, the CDC-SBC algorithm was executed against the UCI voting dataset. The voting dataset consists of actual congressional votes from the U.S. House of Representatives on 16 issues in 1984. Results of running the CDC-SBC algorithm against the voting dataset (with party affiliation omitted) and the corresponding values for seven other categorical data clustering algorithms are shown in Table 1.

Table 1. Effectiveness of different clustering algorithms on the benchmark UCI voting dataset

Algorithm	Correct	Precision	CU	CU'
CDC-SBC	383	0.8805	1.4711	2.9422
COBWEB	378	0.8690	1.4506	2.9011
Ahmad-K	377	0.8667	1.4465	2.8929
LIMBO	376	0.8644	1.4424	2.8847
K-Means	376	0.8644	1.4424	2.8847
Huang-K	364	0.8368	1.3931	2.7861
COOLCAT	363	0.8345	1.3890	2.7779
ROCK	345	0.7931	1.3150	2.6300

The values in the column labeled Correct in Table 1 are the number of tuples in the voting dataset which were correctly clustered as Democrat or Republican by each algorithm. The Precision column is the micro-precision value as described above, which in this situation is just the number of tuples which are correctly clustered, divided by the total number of tuples ($t = 435$) in the voting dataset. The CU column is the category utility of the clustering produced by each algorithm, as defined by equation (1). The CU' column is a slightly different definition of category utility used by some studies, which is not normalized for number of clusters. Because the number of clusters in this situation is 2, the values in the CU' column are simply twice the values in the CU column, and have been included solely to provide a consistent comparison with the reported results of other studies.

The COBWEB clustering algorithm incrementally builds a probabilistic hierarchy tree of clusters from a dataset using category utility to measure clustering effectiveness [13]. The Ahmad-K clustering algorithm is a variation of the Huang-K algorithm, which in turn is based on the simple k-means algorithm [14]. The LIMBO algorithm is based on a concept called the information bottleneck, which is essentially a measure of entropy [12]. The COOLCAT clustering algorithm is an iterative technique that uses a greedy algorithm based combined with an entropy measure [2]. The

ROCK algorithm uses a hierarchical approach in conjunction with a distance measure modeled on graph theory [15].

The data in Table 1 were derived from several sources and should be interpreted somewhat cautiously. Most of the studies represented in Table 1 reported result values for the UCI voting dataset in terms of category utility. In the situations where the number of correct tuples was not reported (COOLCAT, COBWEB, LIMBO), an auxiliary program, developed as part of this study, which computes the number of correct values for a given category utility was employed to produce the values shown in the Correct column of Table 1. The results for the K-Means algorithm were determined by executing the WEKA data analysis tool [16]. Further, published results differ slightly for the ROCK, COOLCAT, and COBWEB algorithms, presumably because of differences in input parameters to the algorithms. In situations where the differences in reported values for these algorithms could not be resolved, the data in Table 1 represent arithmetic means of reported results.

The results of the CDC-SBC algorithm and the seven other algorithms listed in Table 1 represent the best clustering results of the benchmark UCI voting dataset discovered by a comprehensive review of the literature. The data indicates that the CDC-SBC algorithm produced more accurate results than all previously published algorithms for clustering the UCI voting dataset.

4.2 Experiment #2 – Synthetic Datasets

In order to evaluate the efficiency of the CDC-SBC algorithm and its resulting data visualizations, the algorithm was executed against six synthetic datasets. The results are shown in Table 2.

Table 2. Accuracy of the CDC-SBC algorithm on synthetic datasets

Dataset	Attributes	Attribute Values	Tuples	Clusters	Partitions	Precision
DS0	3	(4,3,2)	9	2	$3.02 * 10^{3}$	1.00
DS1	4	(5,5,5,5)	20	3	$5.81 * 10^{8}$	1.00
DS2	5	(2,3,4,3,2)	36	4	$1.97 * 10^{20}$	1.00
DS3	6	(3,3,..,3)	50	5	$7.40 * 10^{32}$	1.00
DS4	10	(2,2,…,2)	200	2	$8.03 * 10^{59}$	0.98
DS5	16	(2,2,…,2)	435	2	$4.44 * 10^{130}$	0.95

After the synthetic datasets had been generated, a program which implemented the CDC-SBC algorithm was executed using each synthetic dataset (without cluster values) as input. The micro-precision was computed for each resulting clustering, and is listed in Table 2. For all synthetic dataset inputs, the maximum number of iterations of the main SBC algorithm loop was limited to a count of 10^8 or until a partitioning result with precision of 1.00 was discovered. The column in Table 2 which is labelled Partitions holds the total number of possible partitions for the associated synthetic dataset, computed using Stirling numbers of the second kind, and is a measure of dataset complexity.

The results in Table 2 suggest that the CDC-SBC algorithm is highly effective at clustering datasets which contain self-consistent data. The results also suggest that the CDC-SBC algorithm is at least reasonably effective at clustering datasets which have huge search spaces. The results for dataset D05 are particularly noteworthy; the CDC-SBC algorithm correctly placed 413 out of 435 tuples from a problem domain with over 10^{130} possible partitions.

5 Conclusions

The results of this study demonstrate the feasibility of using a simulated bee colony meta-heuristic algorithm in conjunction with the category utility function to cluster categorical datasets so that the data can be usefully visualized. Because the scope of this study is limited and is for the most part empirical, it is not possible to draw definitive conclusions from the results. However, when taken as a whole the results suggest that categorical data visualization using the CDC-SBC technique is a promising technique which has the potential to outperform existing algorithms in terms of clustering accuracy and accuracy of any resulting visualization format, and that the technique merits further investigation. One disadvantage of the CDC-SBC algorithm compared to alternative approaches is that CDC-SBC requires a relatively large number of generic algorithm parameters such as the numbers and percentages of each type of bee object, and simulation probabilities such as the probability that an active foraging bee will accept a neighbor solution with a lower category utility value than the current CU value. Because algorithms based on bee behavior are relatively unexplored, there are very few guidelines available for selecting input parameters and trial and error is often required to tune the algorithm for better performance. Additionally, because the CDC-SBC algorithm is probabilistic, there is no guarantee that the algorithm will produce an optimal solution to any clustering problem.

In addition to clustering accuracy, an advantage of the CDC-SBC algorithm compared to existing approaches is that CDC-SBC can in principle be applied to arbitrarily large datasets. A promising potential extension of CDC-SBC is to investigate datasets with mixed categorical and numerical data. According to a mathematical analysis of the category utility function by Mirkin, in spite of a significantly different outward appearance compared to traditional numerical clustering measures, the CU function is in fact closely related to the square-error criterion used in numerical clustering [17]. This raises the possibility of adapting the CDC-SBC algorithm to deal with mixed data using a unified form of CU function.

References

1. Liu, Y., Ouyang, Y., Sheng, H., Xiong, Z.: An Incremental Algorithm for Clustering Search Results. In: Proceedings of the 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems, pp. 112–117 (2008)
2. Barbara, D., Li, Y., Couto, J.: COOLCAT: An Entropy-Based Algorithm for Categorical Clustering. In: Proceedings of the 11th International Conference on Information and Knowledge Management, pp. 582–589 (2002)

3. Gluck, M., Corter, J.: Information, Uncertainty, and the Utility of Categories. In: Program of the 7th Annual Conference of the Cognitive Science Society, pp. 283–287 (1985)
4. Chi, E.: A Taxonomy of Visualization Techniques using the Data State Reference Model. In: Proceedings of the IEEE Symposium on Information Visualization, pp. 69–75 (2000)
5. Seeley, T.D.: The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies. Harvard University Press, Boston (1995)
6. Sato, T., Hagiwara, M.: Bee System: Finding Solution by a Concentrated Search. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, vol. 4, pp. 3954–3959 (1997)
7. Lucic, P., Teodorovic, D.: Transportation Modeling: An Artificial Life Approach. In: Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence, pp. 216–223 (2002)
8. Nakrani, S., Tovey, C.: On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers. Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems 12(3-4), 223–240 (2004)
9. Drias, H., Sadeg, S., Yahi, S.: Cooperative Bees Swarm for Solving the Maximum Weighted Satisfiability Problem. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 318–325. Springer, Heidelberg (2005)
10. Basturk, B., Karaboga, D.: An Artificial Bee Colony (ABC) Algorithm for Numeric Function Optimization. In: Proceedings of the IEEE Swarm Intelligence Symposium, pp. 687–697 (2006)
11. McCaffrey, J.: Generation of Pairwise Test Sets using a Simulated Bee Colony Algorithm. In: Proceedings of the 10th IEEE International Conference on Information Reuse and Integration (2009)
12. Andritsos, P., Tsaparas, P., Miller, R., Sevcik, K.: LIMBO: Scalable Clustering of Categorical Data. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 123–146. Springer, Heidelberg (2004)
13. Fisher, D.: Knowledge Acquisition via Incremental Conceptual Clustering. Machine Learning 2(2), 139–172 (1987)
14. Ahmad, A., Dey, L.: A k-Mean Clustering Algorithm for Mixed Numeric and Categorical Data. Data Knowledge and Engineering 63(2), 503–527 (2007)
15. Hsu, C., Chen, C., Su, Y.: Hierarchical Clustering of Mixed Data Based on Distance Hierarchy. Information Sciences 177(20), 4474–4492 (2007)
16. Holmes, G., Donkin, A., Witten, I.: WEKA: A Machine Learning Workbench. In: Proceedings of the 2nd Austraila and New Zealand Conference on Intelligent Information Systems, pp. 357–361 (1994)
17. Mirkin, B.: Reinterpreting the Category Utility Function. Machine Learning 45(2), 219–228 (2001)