

Clustering Structured Web Sources: a Schema-based, Model-Differentiation Approach

Bin He, Tao Tao, and Kevin Chen-Chuan Chang

Computer Science Department,
University of Illinois at Urbana-Champaign,
Urbana IL 61801, USA,
{binhe,taotao}@uiuc.edu, kcchang@cs.uiuc.edu

Abstract. The Web has been rapidly “deepened” with the prevalence of databases online. On this “deep Web,” numerous sources are *structured*, providing schema-rich data—Their schemas define the *object domain* and its *query capabilities*. This paper proposes clustering sources by their *query schemas*, which is critical for enabling both *source selection* and *query mediation*, by organizing sources of with similar query capabilities. In abstraction, this problem is essentially clustering categorical data (by viewing each query schema as a transaction). Our approach hypothesizes that “homogeneous sources” are characterized by the same hidden generative models for their schemas. To find clusters governed by such statistical distributions, we propose a novel objective function, *model-differentiation*, which employs principled hypothesis testing to maximize statistical heterogeneity among clusters. Our evaluation shows that, on clustering the Web query schemas, the model-differentiation function outperforms existing ones with the hierarchical agglomerative clustering algorithm.

1 Introduction

Recently, the Web has been rapidly “deepened” with the prevalence of databases online and thus presents challenges for *large-scale* information integration. On this “deep Web” (database-backed web sources), numerous online databases provide dynamic *query*-based data access through their *query interfaces*, instead of static URL links. Our recent survey [1] in December 2002 estimated between 127,000 to 330,000 deep Web sources. The deep Web thus presents challenges for *large-scale* information integration: While there are myriad useful databases, how can a user *find* the correct sources and *query* them in a correct way?

While tantalized by the need for effectively accessing the deep Web, such *metaquery* over large-scale structured sources has largely remained unexplored. As a first step toward metaquerying, this paper studies clustering sources by their *query schemas*, i.e., attributes in their query interfaces. For instance, for the advanced query interface of *amazon.com*, the query schema is {*author*, . . . , *publisher*}. Specifically, given a set of query schemas representing structured sources, our task is thus to construct a *hierarchy* of clusters, each representing an object domain of “structurally-homogeneous” sources.

In abstraction, this problem is essentially clustering *categorical data*. We can view a schema as a *transaction* and thus a special type of categorical data. As such data is typically sparse in a high-dimensional space, conventional clustering based on similarity measures does not work well. Several recent efforts have thus developed new *objective functions*, e.g., context-linkages [2] and entropy [3].

In this paper, we pursue model-based clustering with a new objective function, motivated by our observations on the query schemas. In particular, we collected a dataset of deep Web

<i>domain</i>	<i>number of sources</i>	<i>domain</i>	<i>number of sources</i>
Airfares	53	Hotels	38
Automobiles	102	Jobs	55
Books	69	Movies	78
CarRentals	24	MusicRecords	75

Fig. 1. Our dataset of sample deep web sources: 494 sources in 8 domains.

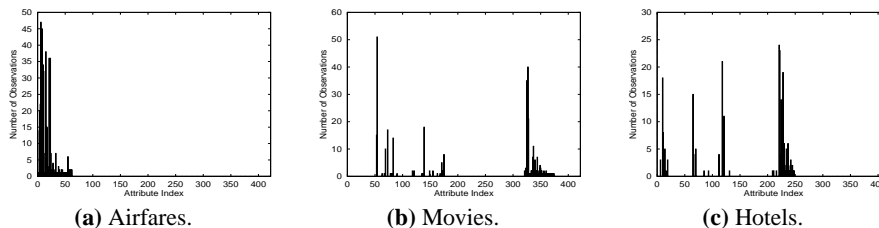


Fig. 2. Attribute frequencies of different domains.

sources using Web directories (e.g., InvisibleWeb.com, BrightPlanet.com) and search engines (e.g., Google.com). As Figure 1 summarizes, the dataset consists of 494 sources in 8 domains.

First, we observe that query schemas are *discriminative* representative of structured sources. Specifically, we count attribute frequencies for each domain (i.e., the aggregate occurrences of an attribute across all sources in the same domain). Figure 2 lists the attribute frequencies (y-axis) of 3 domains (Airfares, Movies and Hotels) over all the attributes (x-axis) in the 8 domains. We observe that each domain contains a dominant range of attributes, distinctive from other domains. For example, Airfares only covers the first 53 attributes and does not overlap with Movies. Hotels has its dominant range of attributes from index 200 to 250 (while overlapping with Airfares in some of the first 53 attributes).

Second, we observe that the aggregate schema vocabulary of sources in the same domain tends to converge at a relatively small size with respect to the growth of sources. As detailed in [1], for each domain, the vocabulary growth rates (i.e., the slopes of these curves) decrease rapidly with respect to the increase of sources. This observation indicates that homogeneous sources (in the same domain) share some *concerted* vocabulary of attributes.

These two observations together motivate our approach: The “discriminative” observation suggests using query schemas as “representatives” of sources in their clustering. Further, the “concerted” observation leads us to hypothesize the existence of a hidden schema model (for each domain), which probabilistically generates schemas from a finite vocabulary of attributes. This hypothesis naturally implies *model-based* clustering: to form clusters according to their underlining models. Further, the “discriminative” observation hints a novel objective function, *model-differentiation* or MD, which seeks to maximize *statistical heterogeneity* among clusters. Rather than relying on ad-hoc cluster-similarity measures, MD takes principled statistical hypothesis testing, called *test of homogeneity* [4], to evaluate if multiple clusters are generated from homogeneous distributions.

Specifically, we develop Algorithm MD_{hac} for clustering query schemas to build a domain hierarchy. First, we develop the statistical model of a cluster as a *multinomial distribution* of attributes observed in the cluster. Second, we adopt χ^2 testing for evaluating the homogeneity among clusters. Third, for hierarchy construction, we use the general hierarchical agglomerative clustering approach [5–7].

We experimented with about 500 real sources in 8 domains (e.g., Airfares, Automobiles, Books). Our goals are two-fold: (1) to evaluate the effectiveness of schema-based clustering for organizing structured sources into domain hierarchies, and (2) to evaluate the performance

of the MD objective by comparing to the existing approaches using context linkages, log-likelihood, and entropy. The results show effectiveness in both aspects.

2 Related Work

We relate our work to the literature in two aspects. *First*, in terms of the *problem*, this paper studies clustering structured sources on the Web. Our goal of clustering sources to facilitate large-scale integration or “metaquery” has largely been unexplored. On one hand, for structured sources, *information integration* has mainly assumed relatively small-scaled, pre-configured systems (e.g., Information Manifold [8], TSIMMIS [9]). On the other hand, research efforts on large-scale search has mostly focused on *text* sources [10–12]. Our focus mixes both of the above: We aim to enable *large-scale* metaquery over *structured* databases.

Second, in terms of the *techniques*, this paper proposes model-differentiation for clustering schema data. Clustering of categorical data has recently been more actively studied, e.g., STIRR [13], CACTUS [14], ROCK [2], and COOLCAT [3]. STIRR treats clustering as a partitioning problem of hypergraph and solves it based on non-linear dynamical systems. CACTUS considers a cluster as a set of pairwise strong connected attributes by measuring attribute occurrences. ROCK, COOLCAT and this paper are pursuing the same direction of defining a new similarity measure involving the *global context* (such as properties of a entire cluster) instead of local pairwise measure. ROCK uses context linkages between data points, and COOLCAT uses entropy of clusters. As an alternative, we develop the model-differentiation measure, which maximizes the statistical heterogeneity among clusters. Section 4 compares these related approaches.

Our statistical approach belongs to the general idea of model-based clustering [5, 6]. In general, such clustering assumes that data is generated from a mixture of distributions, each of which defines a cluster. This general approach is traditionally not specific to categorical data— More recently, reference [7] proposes a multivariate multinomial distribution (in which each feature is an independent multinomial distribution) for categorical data. In comparison, the model we propose for schema data (or transactional data) is “joint” multinomial, where all features are from one multinomial distribution (Section 3.1).

All the existing model-based works essentially use likelihood as the objective function to maximize— In contrast, we propose model differentiation (Section 3.2) by maximizing the statistical heterogeneity among clusters. In our extended report [15], we show that these two objective functions are *equivalent* in assessing the global clustering results. However, they indeed imply *different* greedy “local” similarity measures (which Section 3.3 will develop). Our experiments are also compatible with the likelihood-based HAC clustering.

3 MD-Based Clustering

As motivated in Section 1, we are pursuing a MD-based approach to cluster schema data. In the literature, model-based clustering has been widely discussed. The general idea can be stated as: The population of interest consists of G different clusters, generated by G different models. Given a set of data points (a set of schemas) $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, where each \mathbf{x}_i is independently generated from one of the G models, $\mathcal{M}_1, \dots, \mathcal{M}_G$, the probability of generating \mathbf{x}_i in the k th model is $Pr(\mathbf{x}_i | \mathcal{M}_k)$. A clustering of \mathbf{X} is a partition of \mathbf{X} into G groups: denoted by $(\mathbf{X}; P) = (C_1, \dots, C_G)$, where P partitions \mathbf{X} . The objective of model-based clustering is to identify the partition P that all the \mathbf{x}_i generated from the same model $Pr(\bullet | \mathcal{M}_k)$ are partitioned into a single group.

To realize this model-based clustering for query schemas, we design a model as a multinomial distribution (Section 3.1) and develop model-differentiation as the new *objective function* of clustering based on statistical hypothesis testing. Specifically, guided by this objective

function, we adopt the commonly used χ^2 testing (Section 3.2). Unlike the clustering work in statistics software, which also use χ^2 testing, we apply it for categorical data based on the generative model. Since we are pursuing a hierarchical clustering approach, we apply the widely used HAC (hierarchical agglomerative clustering) algorithm, which needs a measure to quantify the “similarity” between two clusters. In particular, we derive a new similarity measure from the MD objective function for HAC algorithm. (Section 3.3).

3.1 Hypothesis Modeling

To develop the MD-based clustering, we need to define the generative model under one cluster. We first define our model as multinomial distribution. Then we describe how a model generates a schema in statistical way and further how to generate a cluster of schemas. As Section 1 introduced, we view a query schema as a set of attributes for a query interface. For simplicity, in later examples, we denote attributes in letters $\mathbf{A}, \mathbf{B}, \dots$

As detailed in [15], we adopt the *multinoimal model* with the *sampling with replacement* [4] strategy for modeling the schema data. Specifically, we assume attributes are independent each other, which is a commonly used assumption for text data [16]. Thus, a multinomial model \mathcal{M} for some cluster C consists of an exhaustive set of N mutually exclusive events (or attributes) A_1, \dots, A_N (which covers all the attributes observed in C) with associated probabilities p_1, \dots, p_N , $\sum_{j=1}^N p_j = 1$. We denote \mathcal{M} as $\mathcal{M} = \{A_1:p_1, \dots, A_N:p_N\}$. Each trial of \mathcal{M} generates one of the N events. The probability of generating an attribute \mathbf{A} from \mathcal{M} in a single trial is

$$Pr(\mathbf{A}|\mathcal{M}) = \begin{cases} p_i, & \exists i : \mathbf{A} = A_i \\ 0, & otherwise \end{cases} \quad (1)$$

Under this multinomial model, a schema Q is characterized by its observed attributes frequencies. We thus view Q (of length n) as $Q = \{A_1:y_1, \dots, A_N:y_k\}$, $\sum_{i=1}^N y_i = n$, where y_i is the frequency (number of occurrences) of attribute A_i in Q . That is, Q (of length n) is generated from \mathcal{M} as the result of n independent trials with the following probability, by definition of standard multinomial distribution [4]:

$$Pr(Q|\mathcal{M}, n) = n! \prod_{i=1}^N \frac{Pr(A_i|\mathcal{M})^{y_i}}{y_i!}. \quad (2)$$

Consider a cluster of schemas $C = \{Q_1, Q_2, \dots, Q_m\}$, where each schema Q_j (with length n_j) is generated by the same model $\mathcal{M} = \{A_1:p_1, \dots, A_N:p_N\}$. Since each Q_j is a multinomial experiment of n_j trials, we can view C as an experiment with $\sum_{j=1}^m n_j$ trials by concatenating the trials in all schemas. That is, we consider C is a series of sampling from the same multinomial distribution \mathcal{M} (i.e., the same p_1, \dots, p_N), with all these independent trials. The theoretical explanation is as follows: Let all $Q_j = \{A_1:y_{j1}, \dots, A_N:y_{jN}\}$, where y_{ji} 's are random variables denoting the frequencies of A_i , share the same multinomial distribution $\mathcal{M} = \{A_1:p_1, \dots, A_N:p_N\}$. For the entire C , we define new random variables $\mathbf{z}_1, \dots, \mathbf{z}_N$ as aggregate attribute frequencies. That is, $\mathbf{z}_i = \sum_{j=1}^m y_{ji}$. In [15], we show that $\mathbf{z}_1, \dots, \mathbf{z}_N$ also form the same multinomial distribution \mathcal{M} with $\sum_{j=1}^m n_j$ trials. Therefore, under this multinomial view, we can express C as aggregate attribute frequencies, i.e., $C = \{A_1:z_1, \dots, A_N:z_N\}$.

More discussion about this multinomial modeling and its comparison with the model in MGS [17] can be found in our extended report [15].

3.2 Model-Differentiation: A New Objective Function

A clustering must be guided by some *objective function* that specifies the property of the ideal clusters. Regardless of the objective function, the basic idea of clustering is to put similar data

	A_1	A_2	A_3	...	A_n	sum
C_1	O_{11}	O_{12}	O_{13}	...	O_{1n}	X_1
C_2	O_{21}	O_{22}	O_{23}	...	O_{2n}	X_2
...
C_m	O_{m1}	O_{m2}	O_{m3}	...	O_{mn}	X_m
sum	Y_1	Y_2	Y_3	...	Y_n	S

Fig. 3. Contingency table for testing.

together and dissimilar data apart. For model-based clustering, similar data might be generated from the same underlying models, while dissimilar data are from different models. Thus, we achieve better clustering result when the underlying models are more distinguishable.

Therefore, we define the objective function of clustering as some function \mathcal{H} that characterizes the heterogeneity of models under a partition P , denoted by $\mathcal{H}(\mathbf{X}; P)$. The goal of clustering is to find the partition P maximizing function \mathcal{H} , i.e., $\arg \max_P \mathcal{H}(\mathbf{X}; P)$. In statistics, the homogeneity of distributions can be measured by *test of homogeneity* using statistical hypothesis testing. More specifically, if we have a partition function P partitioning \mathbf{X} into clusters $C_k (1 \leq k \leq G)$, we can test the hypothesis “ $C_k (1 \leq k \leq G)$ are generated by same distribution” with standard testing approaches. The result of testing is a probabilistic variable λ to indicate the confidence that we accept the hypothesis. Thus the heterogeneity of models is $1 - \lambda$. Formally, the MD-based clustering is to find

$$\begin{aligned}
\arg \max_P \mathcal{H}(\mathbf{X}; P) &= \arg \max_P \mathcal{H}(C_1, \dots, C_G) \\
&= \arg \max_P \{1 - \lambda(C_1, \dots, C_G)\} \\
&= \arg \min_P \lambda(C_1, \dots, C_G),
\end{aligned} \tag{3}$$

where $\lambda(C_1, \dots, C_G)$ is the hypothesis testing on a partition P with G clusters.

Specifically, given a partition P on the observed data \mathbf{X} , we apply χ^2 hypothesis testing to compute $\lambda(C_1, \dots, C_G)$. In statistics, χ^2 testing can be used to test the homogeneity among multiple clusters with multinomial distributions by constructing a *contingency table*. Since we show that a cluster of schemas is also from a multinomial distribution, we can directly apply the test of homogeneity by fitting the attribute frequencies in the cluster into the contingency table, which reflects the fact that our modeling simplifies the testing.

Formally, assume there are m clusters C_1, \dots, C_m , and each of them is generated from its own multinomial distribution (Section 3.1). There are n different attributes altogether, denoted by A_1, \dots, A_n . Figure 3 is the contingency table to show this set of data. In particular, O_{ij} stands for the attribute frequency of A_j in cluster C_i . X_i is the sum of all the O_{ij} in i th row and Y_j is the sum of all the O_{ij} in j th column. That is, $X_i = \sum_{j=1}^n O_{ij}$ and $Y_j = \sum_{i=1}^m O_{ij}$. S is the sum of all O_{ij} in the table. Thus $S = \sum_{i=1}^m X_i = \sum_{j=1}^n Y_j$. We want to test the hypothesis: $\forall j, 1 \leq j \leq n, p_{j1} = p_{j2} = \dots = p_{jm} = \frac{Y_j}{S}$, where p_{ji} is the probability of observing attribute A_j in C_i . This hypothesis is tested by considering the random variable

$$D^2(C_1, \dots, C_m) = \sum_{i=1}^m \sum_{j=1}^n \left[\frac{(O_{ij} - X_i \times \frac{Y_j}{S})^2}{X_i \times \frac{Y_j}{S}} \right]. \tag{4}$$

It can be shown that D^2 has asymptotically a χ^2 distribution with $(n-1)(m-1)$ degree of freedom, denoted by df [18]. Note that we have to use both the values of D^2 and df to decide how similar the m clusters are. D^2 value itself is not a valid indicator for the similarity of clusters without being qualified the degree of freedom. Therefore we need to translate these

```

Require: SchemaSet  $\mathbf{X}$ , ObjectiveFunction  $\mathcal{F}$ , NumberOfClusters  $G$ 
1: /* Form a list of initial  $V$  clusters */
2:  $C_k = X_k, (1 \leq k \leq V)$ 
3: /* Derive similarity measure */
4:  $s$  = a similarity measure derived from  $\mathcal{F}$ 
5: /* HAC main framework */
6: for  $K = V, V - 1, \dots, G$  do
7:   /* Compute pairwise similarities */
8:    $k^*, l^* = \arg \min_{k,l} \mathcal{H}(C_k, C_l), (1 \leq k < l \leq K)$ 
9:   /* Merge the most similar two clusters */
10:   $C_{<k^*, l^*>} = \text{MERGE}(C_{k^*}, C_{l^*})$ 
11: end for

```

Fig. 4. HAC algorithm MD_{hac} .

two values into a single similarity measure. In statistics, we can compute the P -value given D^2 and df , denoted by $PV(D^2, df)$. The P -value is the probability value λ in Equation 3, indicating the confidence that we accept the hypothesis that the m clusters are generated from the same distribution. The objective function \mathcal{H} is then

$$\mathcal{H}(C_1, \dots, C_G) = 1 - PV(D^2, df). \quad (5)$$

The computation of P -value is expensive and requires numerical integration. Therefore, in practice, we develop an alternative measure, $\tilde{\mathcal{H}}$, by applying a normalized D^2 value. In particular, to make the D^2 values of different degrees of freedom (resulted from different clusters) comparable, we use the D^2 values with a commonly adopted significance level 0.5% as the normalization factors, denoted by $D_s^2(df)$ with different degrees of freedom. We consider $\tilde{\mathcal{H}}$ as the ratio between the computed D^2 value and the D_s^2 with the same df :

$$\tilde{\mathcal{H}}(C_1, \dots, C_G) = \frac{D^2}{D_s^2(df)}. \quad (6)$$

3.3 HAC Algorithm And MD-Based Similarity Measure

For constructing domain hierarchy, we adopt the general HAC clustering approach, which is widely used for data clustering [19]. Figure 4 illustrates the general HAC framework [7]. In HAC, we need to measure the similarity of clusters. That is, given a set of clusters, C_1, \dots, C_V , we compute all the pairwise values $s(k, l)$, where s is a similarity function derived from the objective function of clustering. The criterion of defining similarity function $s(k, l)$ is to maximize the objective function in each step. The two clusters with the smallest $s(k, l)$ are merged in each iteration. The algorithm stops when there are G clusters left.

Specifically, for the MD-based clustering, we derive $s(k, l)$ from $\mathcal{H}(\mathbf{X}; P)$ (defined in Section 3.2) as follows: In each iteration of HAC, we merge the clusters with the smallest \mathcal{H} value (i.e., the most similar two models) and therefore we define $s(k, l)$ to be

$$s(k, l) = \mathcal{H}(C_k, C_l). \quad (7)$$

For the space limitation, we illustrate MD_{hac} in more details in the extended report [15].

4 Experiments

To evaluate the MD_{hac} algorithm, we test it with 8 domains of sources we collected on the deep Web. We totally collected 494 sources, covering 422 attributes. For each source, we manually extract attributes from its query interface and then judge its corresponding domain.

		MD _{hac}							
		Af	Am	Bk	Cr	Ht	Jb	Mv	Mr
C ₁	0	101	0	0	2	4	0	0	0
C ₂	0	0	62	0	0	1	9	2	0
C ₃	0	0	0	24	0	0	0	0	0
C ₄	0	0	0	0	35	0	0	1	0
C ₅	0	0	0	0	0	50	1	0	0
C ₆	53	0	0	0	1	0	0	0	0
C ₇	0	0	0	0	0	0	8	67	0
C ₈	0	1	7	0	0	0	62	7	0

(a) Conditional entropy of MD_{hac}: 0.32.

		EP _{hac}							
		Af	Am	Bk	Cr	Ht	Jb	Mv	Mr
C ₁	0	100	0	0	2	4	0	0	0
C ₂	0	0	62	0	0	0	5	2	0
C ₃	0	0	0	24	0	0	0	0	0
C ₄	0	0	0	0	35	6	0	1	0
C ₅	0	0	0	0	0	0	57	5	0
C ₆	0	0	0	0	0	0	8	67	0
C ₇	53	0	0	0	1	0	0	0	0
C ₈	0	2	7	0	0	45	10	2	0

(b) Conditional entropy of EP_{hac}: 0.38.

Fig. 5. Comparison of different measures in HAC.

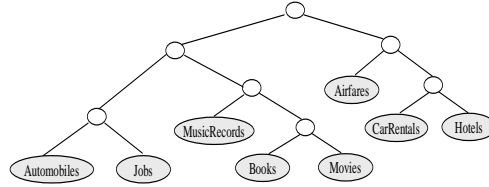


Fig. 6. Domain hierarchy built by MD_{hac}.

This is our ground truth of “correct” clustering. In the experiment, we compare the automatic algorithms with our ground truth, to evaluate the accuracy of clustering.

We compare our MD-based approach with likelihood [7], entropy (COOLCAT) [3] and context linkage (ROCK) [2] based approaches using HAC algorithm. Also, we show the domain hierarchy built by MD_{hac}.

To measure the result of clustering, we adopt the *conditional entropy*, introduced in [20]. For a given number of clusters G , the value of the conditional entropy is within the range from 0 to $\log G$, where 0 denotes the 100% correct clustering, $\log G$ denotes the totally messing up result. Thus, the closer the conditional entropy value is to 0, the better the result is.

First, we compare MD_{hac} with the three existing approaches: likelihood based approach (LK_{hac}), entropy based approach (EP_{hac}) and context linkage based approach (CL_{hac}) for clustering the sources of 8 domains. Figure 5 shows the comparison result. Due to the space limitation, we only show the best two measures, MD_{hac} and EP_{hac}. (Complete comparison can be found at [15]). We use the abbreviations Af, Am, Bk, Cr, Ht, Jb, Mv and Mr to denote the 8 domains Airfares, Automobiles, CarRentals, Hotels, Jobs, Movies and MusicRecords respectively. The results show that: 1) It is feasible to address the clustering of structured sources as the clustering of query schemas. The matrix of MD_{hac}, LK_{hac} and EP_{hac} do show correct clustering for most data. The result of CL_{hac} is not good perhaps because its measure may not fit the schema data well; 2) MD_{hac} achieves the best performance among the four measures on clustering web schemas. In particular, compared with the second best measure, EP_{hac}, MD_{hac} have better results for Jobs and Movies.

Second, we show the effectiveness of MD_{hac} to build the domain hierarchy. After clustering 8 domains, we continue to build the domain hierarchy in the same way as the HAC approach. The result in Figure 6 illustrates that Automobiles and Jobs are merged in the same subtree, MusicRecords, Books and Movies in another subtree, and Airfares, CarRentals and Hotels in a third subtree. This hierarchy is consistent with our observation in the real world.

5 Conclusion

This paper studies the clustering of deep Web sources. Motivated by our observations, we propose to cluster sources by their query schemas and develop a new model-differentiation objective function for clustering. Guided by the MD objective, we develop a new similarity measure for the HAC algorithm. Our experiments show the effectiveness of our approach by comparison with some related existing techniques.

References

1. Chang, K.C.C., He, B., Li, C., Zhang, Z.: Structured databases on the web: Observations and implications. Technical Report UIUCDCS-R-2003-2321, Dept. of Computer Science, UIUC (2003)
2. Guha, S., Rastogi, R., Shim, K.: ROCK: A robust clustering algorithm for categorical attributes. *Information Systems* **25** (2000) 345–366
3. Barbara, D., Li, Y., Couto, J.: Coolcat: An entropy-based algorithm for categorical clustering. In: *CIKM Conference*. (2002)
4. Brunk, H.D.: *An Introduction to Mathematical Statistics*. Blaisdell Pub. Co (1965)
5. Banfield, J.D., Raftery, A.E.: Model-based gaussian and non-gaussian clustering. *Biometrics* **49** (1993) 803–821
6. Fraley, C.: Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing* **20** (1999) 270–281
7. Meila, M., Heckerman, D.: An experimental comparison of several clustering and initialization methods. Technical report, Microsoft Research, MSR-TR-98-06 (1998)
8. Levy, A.Y., Rajaraman, A., Ordille, J.J.: Querying heterogeneous information sources using source descriptions. In: *VLDB Conference*. (1996)
9. Papakonstantinou, Y., García-Molina, H., Ullman, J.: Medmaker: A mediation system based on declarative specifications. In: *ICDE Conference*. (1996)
10. Callan, J.P., Connell, M., Du, A.: Automatic discovery of language models for text databases. In: *SIGMOD Conference*. (1999)
11. Panagiotis G. Ipeirotis, Luis Gravano, M.S.: Probe, count, and classify: Categorizing hidden web databases. In: *SIGMOD Conference*. (2001)
12. Meng, W., Liu, K.L., Yu, C.T., Wang, X., Chang, Y., Rishe, N.: Determining text databases to search in the internet. In: *VLDB Conference*. (1998)
13. Gibson, D., Kleinberg, J.M., Raghavan, P.: Clustering categorical data: An approach based on dynamical systems. *VLDB Journal* **8** (1998) 222–236
14. Ganti, V., Gehrke, J., Ramakrishnan, R.: CACTUS - clustering categorical data using summaries. In: *Knowledge Discovery and Data Mining*. (1999) 73–83
15. He, B., Tao, T., Chang, K.C.C.: Clustering structured web sources: A schema-based, model-differentiation approach. Technical Report UIUCDCS-R-2003-2322, Dept. of Computer Science, UIUC (2003)
16. Ponte, J., Croft, W.: A language modelling approach to information retrieval. In: *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval*. (1998)
17. He, B., Chang, K.C.C.: Statistical schema matching across web query interfaces. In: *Proceedings of the 2003 ACM SIGMOD Conference*. (2003)
18. Agresti, A.: *Categorical Data Analysis*. John Wiley & Sons, Inc. New Jersey (2002)
19. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. *ACM Computing Surveys* **31** (1999) 264–323
20. Berkhin, P.: Survey of clustering data mining techniques. Technical report, Accrue Software (2002)