CLICKS: Mining Subspace Clusters in Categorical Data via K-partite Maximal Cliques

Mohammed J. Zaki * Markus Peters

Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY

Abstract

We present a novel algorithm called CLICKS, *that finds clusters in categorical datasets based on a search for k-partite maximal cliques. Unlike previous methods,* CLICKS *mines subspace clusters. It uses a* selective vertical method *to guarantee complete search.* CLICKS *outperforms previous approaches by over an order of magnitude and scales better than any of the existing method for high-dimensional datasets. We demonstrate this improvement in an excerpt from our comprehensive performance studies.*

1 Introduction

Clustering of numeric data has been widely studied, but categorical data has received relatively less attention. The main challenges are: i) the lack of natural order on the individual domains, which renders traditional similarity measures ineffective, ii) the high dimensionality of typical categorical datasets, and iii) the fact that full-space clustering is insufficient as subspace clusters often dominate in real-world data.

In this paper, we present CLICKS [1], a novel algorithm for mining categorical (subspace) clusters. The main contributions are: i) A novel formalization of categorical datasets as k-partite graphs, where clusters correspond to k-partite cliques after post-processing. ii) A *selective vertical expansion* approach to guarantee a complete search; overlapping cliques are merged to report more meaningful clusters. iii) CLICKS outperforms existing approaches by over an order of magnitude. It can mine subspace clusters and scales extremely well for high dimensions.

2 The CLICKS Algorithm

Let $A_1, \ldots, A_n$ be a set of *categorical attributes* and $D_1, \ldots, D_n$ a set of *domains*, where $D_i = \{v_{i_1}, \ldots, v_{i_m}\}$ is the domain for attribute A_i, and $D_i \cap D_j = \emptyset$ for $i \neq j$. A *dataset* is given as $\mathcal{D} \subseteq D_1 \times \ldots \times D_n$.

Let $S_j \subseteq D_{i_j}$ be a subset of values for attribute A_{i_j}. A *k-subspace* (with $k \leq n$) is defined as the cross-product $S = S_1 \times \ldots \times S_k$. Each S_j is called a *projection* of S on attribute A_{i_j}. Given any collection $\mathcal{S}$ of subspaces, $M \in \mathcal{S}$ is a *maximal subspace* iff there does not exist $M' \in \mathcal{S}$, such that $M \subset M'$. The support $\sigma(S)$ of S is the count of all records r in $\mathcal{D}$ where the j-th attribute value $r.A_j \in S_{i_j}$ for $j \in \{1, \ldots, k\}$. S is called *frequent* if $\sigma(S) \geq \sigma^{\min}$, for some user-defined threshold $\sigma^{\min}$. Under attribute independence, the *expected support* of S in $\mathcal{D}$ is given as $E[\sigma(S)] = |\mathcal{D}| \cdot \prod_{j=1}^{k} \frac{|S_j|}{|D_{i_j}|}$.

For $\alpha \in \mathbb{R}^+$ we define a *density indicator* function $\delta_\alpha(S) = 1$ iff $\sigma(S) \geq \alpha \cdot E[\sigma(S)]$, otherwise $\delta_\alpha(S) = 0$. S is called a *dense subspace* iff $\delta_\alpha(S) = 1$, that is, if its expected support exceeds its actual support by a user-defined factor α. S_i and S_j, are called *strongly connected* iff $\forall v_a \in S_i$ and $\forall v_b \in S_j$, the 2-subspace $\{v_a\} \times \{v_b\}$ is dense. $S = S_1 \times \ldots \times S_k$ is called a *strongly connected subspace* iff S_i is strongly connected to S_j for all $1 \leq i < j \leq k$.

Definition 2.1 (Categorical Cluster) *Let $\mathcal{D}$ be a categorical dataset and $\alpha > \mathbb{R}^+$. The k-subspace $C = (C_1 \times \ldots \times C_k)$ is a* (subspace) cluster *over attributes $A_{i_1}, \ldots, A_{i_k}$ iff it is a maximal, dense, and strongly connected subspace in $\mathcal{D}$.*

CLICKS models $\mathcal{D}$ as graph where the vertices (attribute values) form k disjoint sets (one per attribute); edges exist between vertices in different partitions, indicating dense relationships. CLICKS then maps the categorical clustering problem to the problem of enumerating maximal k-partite cliques in the k-partite graph.

Definition 2.2 (k-Partite Graph and Clique) *Let $\mathcal{D}$ be a categorical dataset over $A_1, \ldots, A_n$ and $V = \bigcup_{i=1}^{n} D_i$. The undirected graph $\Gamma_\mathcal{D} = (V, E)$ where $(v_i, v_j) \in E \iff \delta_\alpha(\{v_i\} \times \{v_j\}) = 1$ is called* k-partite graph *of $\mathcal{D}$. $C \subseteq V$ is a* k-partite clique *in $\Gamma_\mathcal{D}$ iff every vertex pair $v_i \in C \cap D_i$ and $v_j \in C \cap D_j$ with $i \neq j$ is connected by an edge in $\Gamma_\mathcal{D}$. If there is no $C' \supset C$ s.t. C' is a k-partite clique in $\Gamma_\mathcal{D}$, C is called a* maximal k-partite clique. *C is* dense *if $\delta_\alpha(C) = 1$ in $\mathcal{D}$.*

Lemma 2.3 *Given a categorical dataset $\mathcal{D}$ and a k-subspace $C = C_1 \times \ldots \times C_k$ with $C_j \subseteq D_{i_j}$ over attributes $A_{i_1}, \ldots, A_{i_k}$. C is a k-cluster in $\mathcal{D}$ if and only if C is a maximal, dense k-partite clique in $\Gamma_\mathcal{D}$.*

Given a dataset $\mathcal{D}$ and a user-specified threshold $\alpha \in \mathbb{R}^+$, we are interested in mining all full-space and subspace clusters in $\mathcal{D}$. CLICKS uses a three-step approach to this end: In the *pre-processing step* we create the k-partite graph from the input database $\mathcal{D}$, and rank the attribute values v

*Corresponding author: zaki@cs.rpi.edu. This work was supported in part by NSF CAREER Award IIS-0092978, DOE Career Award DE-FG02-02ER25538, and NSF grant EIA-0103708.

[1] CLICKS stands for the bold letters in **S**ubspace **CL**uster**I**ng of **C**ategorical data via maximal **K**-partite cliques.

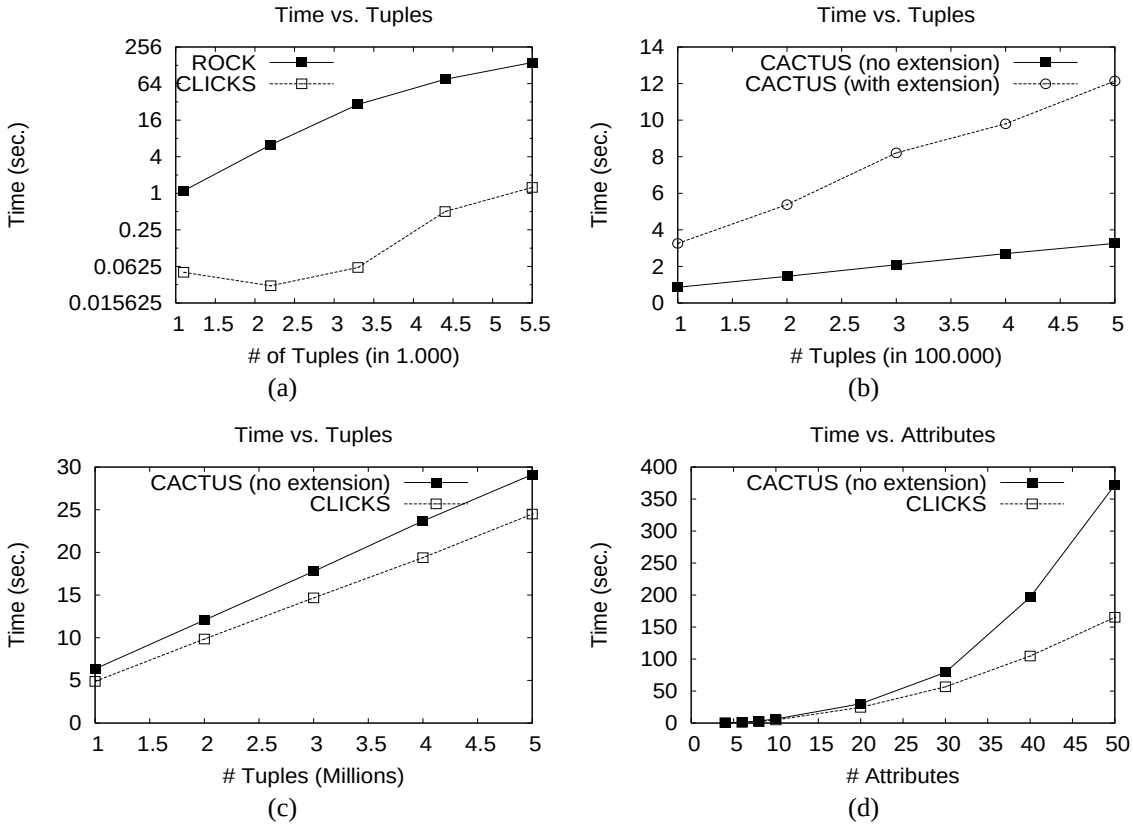

Figure 1. Performance Comparison: CLICKS**, CACTUS, ROCK**

by their connectivity $|\eta(v)|$ for efficiency reasons; $\eta(v)$ is given as the set of neighbors and the remaining values of the same attribute. In the *clique detection* step, we enumerate maximal k-partite cliques in the graph $\Gamma_{\mathcal{D}}$. Our approach is based on a backtracking search that at each step tries to expand the current clique to ensure maximality; it adds only those vertices to a clique that are in the connectivity set $\eta(C)$ of the clique (i.e., those that are connected to all previous vertices in C). In the *post-processing* phase we verify the density property for the detected cliques. A maximal clique may fail the density test, whereas one of its subcliques may be dense. To guarantee completeness, CLICKS allows an optional *selective vertical expansion* approach to explore the sub-cliques induced by a non-dense maximal clique. Further, we optionally merge similar clusters to partially relax the strict cluster notion. For more details of our approach, please see [4].

3 Experimental Study

We present a study of CLICKS versus CACTUS [1], and ROCK [3] (we also tested against STIRR [2]). Testing was done on an Intel Xeon 2.8GHz with 6 GB RAM running Linux. We generated synthetic datasets on 3-50 attributes and with 100 values per attribute. Starting from a uniformly distributed base dataset we embedded two clusters, located on attribute values $[0, 9]$ and $[10, 19]$ for every attribute. Each cluster was created by adding an additional 5% of the original number of records in this subspace region.

We found that CLICKS outperforms ROCK by over an order of magnitude (Fig. 1(a)). The original CACTUS implementation does not perform the extension step to extract the final clusters. When we extended CACTUS to report the final clusters, we found that it is three times slower than the baseline version (Fig. 1(b)).

We varied the dataset size from one to five million tuples (Fig. 1(c), 10 attributes). Both methods scale linearly but CLICKS outperforms CACTUS (with no extension) by an average of 20%. On a dataset with one million records and 100 attribute values per dimension, CLICKS outperforms CACTUS by a factor 2 - 3, when varying the number of attributes from 10 to 50 (Fig. 1(d)). Given that the extension step slows CACTUS down by over a factor of 3, CLICKS can be over an order of magnitude faster than CACTUS. Our full study [4] also shows that the clustering quality is better than in previous methods.

References

[1] V. Ganti, J. Gehrke, and R. Ramakrishnan. CACTUS: Clustering categorical data using summaries. *SIGKDD*, 1999.

[2] D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamical systems. *VLDB*, 1998.

[3] S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes. *ICDE*, 1999.

[4] M. Peters and M. J. Zaki. CLICK: Clustering categorical data using k-partite maximal cliques. TR 04-11, CS Dept., RPI, 2004.