CS724: Topics in Algorithms
Clustering of Categorical Data

Prof. Dan A. Simovici

UMASS BOSTON

Categorical Data

Categorical data consists of symbols; the domains of categorical attributes are sets of categorical data.
For an attribute A we denote by $\mathrm{Dom}(A)$ its domain. In general, we assume that $\mathrm{Dom}(A)$ contains at least two elements.

Example

The attributes **color** and **dow** having the domains
$\mathrm{Dom}(\texttt{color}) = \{red, yellow, grey, blue, black, green\}$ and
$\mathrm{Dom}(\texttt{dow}) = \{Mon, Tue, Wed, Thu, Fri, Sat, Sun\}$ are categorical

Categorical attributes may be

- nominal (no natural order is defined on their domain), or
- ordinal (a natural order exists on the domain)

Operations that make sense:

- equality test for nominal attributes;
- comparisons using relational operators for nominal attributes.

UMASS BOSTON

Let $I = \{i_1, \ldots, i_n\}$ be a set of items. Any subset X of I is an itemset. If $|X| = k$, then X is a k-itemset. The set of all k-itemsets is denoted by $I^{(k)}$.

Example

I can be:

- the set of products sold at a supermarket;
- the set of web pages at a web site,

Definition

A *transaction* is a pair $t = (d, X)$, where $d \in \mathbb{N}$ is a transaction identifier and X is an itemset referred to as the content of t.

Database Representation of Collection of Itemsets

A set of transactions $\mathcal{T}$ on a set of items $I = \{i_1, \ldots, i_n\}$ can be represented by a table T with binary attributes $i_1, \ldots, i_n$.
A transaction $t = (d, X)$ is represented by a tuple $(a_{d1}, a_{d2}, \ldots, a_{dn})$, where $a_{dp} \in \{0, 1\}$ is defined as

$$a_{dp} = \begin{cases} 1 & \text{if } a_p \in X, \\ 0 & \text{otherwise.} \end{cases}$$

A set of transactions is represented by a binary table:

T

	i_1	i_2	$\cdots$	i_n
1	a_{11}	a_{12}	$\cdots$	a_{1n}
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
m	a_{m1}	a_{m2}	$\cdots$	a_{mn}

Example

We present a transaction database as a list of transactions and as a binary table:

T

d	X
1	{a,b,d,e}
2	{b,e}
3	{c,d,e}
4	{a,b,c}
5	{a,d,e}
6	{a,c,d}

T

	A	B	C	D	E
1	1	1	0	1	1
2	0	1	0	0	1
3	0	0	1	1	1
4	1	1	1	0	0
5	1	0	0	1	1
6	1	0	1	1	0

Definition

The support of an itemset U in a dataset D is the number $supp_D(U)$ defined as

$$supp_D(U) = \frac{|\{d \mid (d, X) \in \mathcal{T} \text{ and } U \subseteq X\}|}{|D|}.$$

Example

The support of the itemset $\{a, b\}$ in the previous data set is $\frac{2}{6}$.
Note that if $U \supseteq V$, then $supp_D(V) \leqslant supp_D(U)$ (the anti-monotonicity of the support.

We are interested in item datasets from the point of view of clustering transactions represented as subsets of the set of items, or as binary vectors. The study of itemsets is important and very interesting from a data mining point of view.

Let A, B be two subsets of a finite set S. It is easy to see that the mapping $d : \mathcal{P}(S) \longrightarrow \mathbb{R}_{\geqslant 0}$ defined by $d(A, B) = |A \oplus B|$ is a metric on $\mathcal{P}(S)$. Next, we present an alternative dissimilarity on sets. The *Jaccard similarity coefficent* is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

when at least one of the conditions $A \neq \emptyset$ or $B \neq \emptyset$ is satisfied. We supplement this definition with $J(\emptyset, \emptyset) = 1$.

It is immediate that $0 \leqslant J(A,B) \leqslant 1$.
If $J(A,B) = 0$, the sets A and B are disjoint; if $J(A,B) = 1$, then $A \cap B = A \cup B$ which implies

$$A = A \cup (A \cap B) = A \cup (A \cup B) = A \cup B, B = B \cup (A \cap B) = B \cup (A \cup B) = A \cup B,$$

hence $A = B$.

Starting from the Jaccard similarity we can define a dissimilarity $\delta : \mathcal{P}(S) \longrightarrow [0,1]$ as $\delta(A,B) = 1 - J(A,B)$. We shall prove that δ is a metric on the set $\mathcal{P}(S)$.

Note that for $A \neq \emptyset$ we have $\delta(A,\emptyset) = 1$; of course, $\delta(\emptyset,\emptyset) = 0$.

The difficult part of the proof that δ is a metric is to show that δ satisfies the triangular inequality. We present an alternative, more elegant argument based on the notion of *modular function*.

Definition

Let S be a finite non-empty set. A function $f : \mathcal{P}(S) \longrightarrow \mathbb{R}$ is *submodular* if $f(A \cup B) + f(A \cap B) \leqslant f(A) + f(B)$ for all $A, B \in \mathcal{P}(S)$.
If $f(A \cup B) + f(A \cap B) = f(A) + f(B)$ for all $A, B \in \mathcal{P}(S)$ the function f is *modular*.

Clearly, the constant function $f(A) = 0$ for all $A \in \mathcal{P}(S)$ is modular.

Example

Let S be a finite set and let $f : \mathcal{P}(S) \longrightarrow \mathbb{R}$ be $f(A) = |A|$. Then, f is a modular function.

Definition

For a non-negative, monotone, and modular function $f : \mathcal{P}(S) \longrightarrow \mathbb{R}$ such that $f(\emptyset) = 0$ the function $\delta_f : \mathcal{P}(S)^2 \longrightarrow \mathbb{R}$ is given by

$$\delta_f(A, B) = \frac{f(A \cap B)}{f(A \cup B)}$$

for all $A, B \in \mathcal{P}(S)$.

Theorem

Let $f : \mathcal{P}(S) \longrightarrow \mathbb{R}$ *be a non-negative, monotone, and submodular function. For all* $A, B, C \in \mathcal{P}(S)$ *we have*

$$f(A \cap C) \cdot f(B \cup C) + f(A \cup C) \cdot f(B \cap C) \leqslant f(C) \cdot (f(A) + f(B)).$$

UMASS BOSTON

Proof

By the monotonicity of f we have

$$f(A \cap C)f(B) - f(A \cap C)f(B \cap C) \leqslant f(C)f(B) - f(C)f(B \cap C). \quad (1)$$

This allows us to write

$$\begin{aligned}
f(A \cap C) \cdot f(B \cup C) &\leqslant f(A \cap C) \cdot (f(B) + f(C) - f(B \cap C)) \\
&\quad \text{(by the submodularity of } f\text{)} \\
&= f(A \cap C) \cdot f(B) + f(A \cap C) \cdot f(C) \\
&\quad - f(A \cap C) \cdot f(B \cap C)) \\
&\leqslant f(A \cap C) \cdot f(C) + f(C)f(B) - f(C)f(B \cap C) \\
&\quad \text{(by Inquality (1))} \\
&= f(C) \cdot (f(A \cap C) + f(B) - f(B \cap C)).
\end{aligned}$$

Proof cont'd

By swapping A and B we obtain the similar inequality:

$$f(B \cap C) \cdot f(A \cup C) \leqslant f(C) \cdot (f(B \cap C) + f(A) - f(A \cap C)).$$

The above inequalities imply

$$\begin{aligned} & f(A \cap C) \cdot f(B \cup C) + f(B \cap C) \cdot f(A \cup C) \\ \leqslant \quad & f(C) \cdot (f(B) + f(A)). \end{aligned}$$

Corollary

Let $f : \mathcal{P}(S) \longrightarrow \mathbb{R}$ be a non-negative, monotone, and submodular function. For all sets $U, V \in \mathcal{P}(S)$ we have

$$f(U \cap V) \cdot f(U \cup V) \leqslant f(U) \cdot f(V).$$

Proof.

Take $A = B = U$ and $C = V$ in Theorem 10. □

Theorem

Let f be a non-negative, monotone, and modular function $f : \mathcal{P}(S) \longrightarrow \mathbb{R}$ such that $f(\emptyset) = 0$. The Jaccard dissimilarity $\delta_f : \mathcal{P}(S) \longrightarrow \mathbb{R}$ satisfies the triangular inequality that is

$$\delta_f(A, B) \leqslant \delta_f(A, C) + \delta_f(C, B)$$

for every $A, B, C \in \mathcal{P}(S)$.

Proof

Note that any of the sets A, B, C is $\emptyset$ the triangular inequality is satisfied.
Therefore, we may assume that none of these sets is empty.
The inequality of the theorem is equivalent to

$$1 - \frac{f(A \cap B)}{f(A \cup B)} \leqslant 1 - \frac{f(A \cap C)}{f(A \cup C)} + 1 - \frac{f(C \cap B)}{f(C \cup B)},$$

or equivalently

$$\frac{f(A \cap C)}{f(A \cup C)} + \frac{f(C \cap B)}{f(C \cup B)} \leqslant 1 + \frac{f(A \cap B)}{f(A \cup B)} = \frac{f(A) + f(B)}{f(A \cup B)}$$

for all non-empty subsets A, B, C of S.

Proof cont'd

This inequality can be obtained as follows:

$$
\begin{aligned}
&\frac{f(A\cap C)}{f(A\cup C)}+\frac{f(C\cap B)}{f(C\cup B)}\\
&= \frac{f(A\cap C)\cdot f(C\cup B)+f(C\cap B)\cdot f(A\cup C)}{f(A\cup C)\cdot f(B\cup C)}\\
&\leqslant \frac{f(C)\cdot(f(A)+f(B)}{f(A\cup C)\cdot f(B\cup C)}\\
&\quad \text{(by Theorem 10)}\\
&\leqslant \frac{f(C)\cdot(f(A)+f(B)}{f((A\cup C)\cap(B\cup C))\cdot f(A\cup B\cup C)}\\
&\quad \text{(by Corrolary 11)}\\
&\leqslant \frac{f(C)}{f((A\cap B)\cup C)}\cdot\frac{f(A)+f(B)}{f(A\cup B)}\\
&\quad \text{(by the monotonicity of } f\text{)}\\
&\leqslant \frac{f(A)+f(B)}{f(A\cup B)}.
\end{aligned}
$$

Corollary

The standard Jaccard dissimilarity $\delta : \mathcal{P}(S) \longrightarrow \mathbb{R}$ *satisfies the triangular inequality and is, therefore, a metric on* $\mathcal{P}(S)$.

Proof.

This is immediate consequence of the previous Theorem. □

Example

If sets are represented by their characteristic vectors, the Jaccard similarity is computed by the function

```
jaccard <- function(u,v) {
  if (length(u) != length(v)){
    print(''Vectors must have equal sizes!'')
  }
  int <- pmin(u,v)
  uni <- pmax(u,v)
  return(round(sum(int)/sum(uni),2))
}
```

Note that the precision is limited here at 2 decimal digits.

Consider a market basket database containing the following four transactions over the set of items $\{1, 2, 3, 4, 5, 6\}$:

Trans.	Items
(a)	{1, 2, 3, 5}
(b)	{2, 3, 4, 5}
(c)	{1, 4}
(d)	{6}

The transactions can be regarded as points with binary attributes

$$\mathbf{t}_1 = (1, 1, 1, 0, 1, 0),$$
$$\mathbf{t}_2 = (0, 1, 1, 1, 1, 0),$$
$$\mathbf{t}_3 = (1, 0, 0, 1, 0, 0),$$
$$\mathbf{t}_4 = (0, 0, 0, 0, 0, 1).$$

The centroid method of clustering

- Using Euclidean distance to measure the closeness between points, the distance between the first two points is $\sqrt{2}$ and this is the smallest distance between pairs of points. As a result, they are merged by a centroid-based hierarchical algorithm. The centroid of the new merged cluster is $(0.5, 1, 1, 0.5, 1, 0)$.
- In the next, step, the third and fourth points are merged since the distance between them is $\sqrt{3}$ which is less than the distance between the centroid of the merged cluster and each of them ($\sqrt{3.5}$ and $\sqrt{4.5}$, respectively).
- This corresponds to merging (c) and (d) that do not have a single item in common. This, using distances between the centroids of clusters when making decisions about the clusters to merge could be erroneous.

Once points belonging to different clusters are merged, the situation gets progressively worse. Consider, for example, the set of transactions:

$$\begin{aligned}
\mathbf{t}_1 &= (1,0,0,0,1,0),\\
\mathbf{t}_2 &= (0,1,0,0,0,0),\\
\mathbf{t}_3 &= (0,0,1,0,0,0),\\
\mathbf{t}_4 &= (0,0,0,1,0,0),\\
\mathbf{t}_5 &= (0,0,0,0,1,0),\\
\mathbf{t}_6 &= (0,0,0,0,0,1).
\end{aligned}$$

Consider the clusters $C_1 = \{t_1, t_2, t_3\}$ and $C_2 = \{t_4, t_5, t_6\}$ having the centers $\mathbf{c}_1 = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0\right)$ and $\mathbf{c}_2 = \left(0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$, respectively.

- Both clusters contain three points The Euclidean distance between the centers is $d(\mathbf{c}_1, \mathbf{c}_2) = \frac{\sqrt{2}}{\sqrt{3}}$. This distance is lower than the distance $d(\mathbf{t}_1, \mathbf{c}_1) = \frac{2}{\sqrt{3}}$.
- A clustering method based on distance will merge C_1 and C_2 and will generate a new cluster with center $\mathbf{c} = \left(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right)$. The distance from $\mathbf{t}_1$ to the new cluster center is $d(\mathbf{t}_1, \mathbf{c}) = \frac{\sqrt{5}}{\sqrt{6}}$, which is larger than $d(\mathbf{t}_1, \mathbf{c}_1)$.
- The center of the cluster is spreading over more attributes and the information concerning the tuples represented by the cluster centers is increasingly diluted.

ROCK is essentially a hierarchical clustering algorithm applicable to categorical data. The construction of the tree of clusters is done in a manner that maximizes intra-cluster similarity. Similarity is evaluated using the notion of link between objects.

Let (S, s) be a similarity space with the similarity function ranging in $[0, 1]$. The *set θ-neighbors* of $x \in S$ is

$$N_\theta(x) = \{y \in S \mid y \neq x \text{ and } s(x, y) \geqslant \theta\}.$$

Clustering objects based only on similarity is not capable to distinguish between object that are not well-separated; in this situation, even if two objects are not well-separated it is unlikely that they will have a large number of common neighbors.

Definition

Let p, q be two objects in a similarity space (S, s). Then,

$$\text{links}_\theta(p, q) = |N_\theta(p) \cap N_\theta(q)|$$

is the number of common θ-neighbors between p and q.

The graph associated with a similarity space (S, s), a threshold θ and a number k is $G_{\theta,k} = (S, E_{\theta,k})$, where $\{p, q\} \in E_\theta$ if $\text{links}_\theta(p, q) \geqslant k$.

UMASS BOSTON

Example

Consider the sets C_1 and C_2 of market baskets on the set of items $\{i_1, \ldots, i_7\}$:

$$\begin{aligned} C_1 &= \{\{i_1, i_2, i_3\}, \{i_1, i_2, i_4\}, \{i_1, i_2, i_5\}, \{i_1, i_3, i_4\}, \{i_1, i_3, i_5\}, \\ &\quad \{i_1, i_4, i_5\}, \{i_2, i_3, i_4\}, \{i_2, i_3, i_5\}, \{i_2, i_4, i_5\}, \{i_3, i_4, i_5\}\} \\ C_2 &= \{\{i_1, i_2, i_6\}, \{i_1, i_2, i_7\}, \{i_1, i_6, i_7\}, \{i_2, i_6, i_7\}\}. \end{aligned}$$

The transactions of C_1 are numbered from t_1 to t_{10}; the transactions of C_2 are numbered from t_{11} to t_{14}.

Example

The characteristic vectors of the baskets are the rows of the $\mathbb{R}^{14\times 7}$-matrix

$$\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1
\end{pmatrix}$$

The matrix of Jaccard similarities is computed by using the `jaccard` **R** function before by using the function

```
matrj <- function(A){
   n <- dim(A)[1]
   M <- matrix(OL,n,n)
   for(i in 1:dim(A)[1]){
     for(j in 1:dim(A)[1]){
     M[i,j] <- jaccard(A[i,],A[j,])
     }
   }
   return(M)
}
```

The function call `matrj(A)` produces a (14×14)-matrix of Jaccard coefficients:

	[, 1]	[, 2]	[, 3]	[, 4]	[, 5]	[, 6]	[, 7]	[, 8]	[, 9]	[, 10]	[, 11]	[, 12]	[, 13]	[, 14]
[1,]	1.0	0.5	0.5	0.5	0.5	0.2	0.5	0.5	0.2	0.2	0.5	0.5	0.2	0.2
[2,]	0.5	1.0	0.5	0.5	0.2	0.5	0.5	0.2	0.5	0.2	0.5	0.5	0.2	0.2
[3,]	0.5	0.5	1.0	0.2	0.5	0.5	0.2	0.5	0.5	0.2	0.5	0.5	0.2	0.2
[4,]	0.5	0.5	0.2	1.0	0.5	0.5	0.5	0.5	0.2	0.5	0.2	0.2	0.2	0.0
[5,]	0.5	0.2	0.5	0.5	1.0	0.5	0.2	1.0	0.2	0.5	0.2	0.2	0.2	0.0
[6,]	0.2	0.5	0.5	0.5	0.5	1.0	0.2	0.5	0.5	0.5	0.2	0.2	0.2	0.0
[7,]	0.5	0.5	0.2	0.5	0.2	0.2	1.0	0.2	0.5	0.5	0.2	0.2	0.0	0.2
[8,]	0.5	0.2	0.5	0.5	1.0	0.5	0.2	1.0	0.2	0.5	0.2	0.2	0.2	0.2
[9,]	0.2	0.5	0.5	0.2	0.2	0.5	0.5	0.2	1.0	0.5	0.2	0.2	0.0	0.2
[10,]	0.2	0.2	0.2	0.5	0.5	0.5	0.5	0.5	0.5	1.0	0.0	0.0	0.0	0.0
[11,]	0.5	0.5	0.5	0.2	0.2	0.2	0.2	0.2	0.2	0.0	1.0	0.5	0.5	0.5
[12,]	0.5	0.5	0.5	0.2	0.2	0.2	0.2	0.2	0.2	0.0	0.5	1.0	0.5	0.5
[13,]	0.2	0.2	0.2	0.2	0.2	0.2	0.0	0.2	0.0	0.0	0.5	0.5	1.0	0.5
[14,]	0.2	0.2	0.2	0.0	0.0	0.0	0.2	0.0	0.2	0.0	0.5	0.5	0.5	1.0

The Jaccard coefficient between two distinct transactions in the first group ranges between 0.2 to 0.5. On other hand, although the Jaccard coefficient of the transactions $\{1, 2, 3\}$ and $1, 2, 7$ is 0.5 they belong to two different clusters, while $\{1, 2, 3\}$ and $\{3, 4, 5\}$ have a low Jaccard coefficient (0.2) but belong to the same cluster.

Note that

$$\begin{aligned} N_{0.5}(t_{11}) &= \{t_1, t_2, t_3, t_{12}, t_{13}, t_{14}\} \\ N_{0.5}(t_{12}) &= \{t_1, t_2, t_3, t_{11}, t_{13}, t_{14}\}. \end{aligned}$$

Thus, the transaction $t_{11} = \{i_1, i_2, i_6\} \in C_2$ has five links with transaction t_{12} in its own cluster C_2 and only three links with $t_1 = \{i_1, i_2, i_3\} \in C_1$. Similarly, transaction $t_{13} = \{i_1, i_6, i_7\}$ has two links at level 0.5 with every transaction in C_2 and no links with any transaction in C_1 at this level.

ROCK estimates the number of links in the cluster C_i as $|C_i|^{1+2f(\theta)}$, where $f(\theta)$ is a function that is dependent on the data set as well as the kind of clusters that are of interest. This is based on the assumption that every point in C_i has approximatively $|C_i|^{f(\theta)}$ neighbors in C_i. Since we assume that points outside C_i generate a small number of links to the points in C_i, it follows that each point in C_i contributes $|C_i|^{2f(\theta)}$ links, one for each pairs of its neighbors. This justifies the estimate of

$$|C_i| \cdot |C_i|^{2f(\theta)} = |C_i|^{1+2f(\theta)}$$

for the expected number of links.

ROCKS seeks clusterings $C_1, \ldots, C_n$ that will maximize

$$E = \sum_{i=1}^{k} |C_i| \cdot \sum \left\{ \frac{\text{links}_\theta(p,q)}{|C_i|^{1+2f(\theta)}} \middle| p, q \in C_i \right\}.$$

Dividing $|C_i|$ by the expected number of links in C_i prevents points with very few links between them from being put in the same cluster because assigning them to the same cluster would cause the expected number of links for the cluster to increase more than the actual number of links and the result would be a smaller value for the criterion function.
Since the contribution of every cluster to E is normalized by $|C_i|^{1+2f(\theta)}$ errors in choosing f affect all clusters similarly.

The determination of the function f is a difficult practical problem; In [**?**] authors suggest that

$$f(\theta) = \frac{1-\theta}{1+\theta}$$

is a good choice for market basket data. This can be arrived at by assumming that transactions are of the same approximative size z and are uniformly distributed among m items purchased by customers in cluster C_i. Thus, for some constant c, where $c \leqslant 1$, the number of transactions in the cluster is $\binom{mc}{z}$.

Let t' be a transaction that has $r = |t' \cap t|$ items in common with a transaction t. Since the expected size of a transaction is z, the size $|t' \cup t|$ of the set $t' \cup t$ can be estimated to be approximatively $z + z - r = 2z - r$. To have $J(t', t) = \frac{r}{2z-r} \geqslant \theta$ it suffices to have $r \geqslant \frac{2z\theta}{1+\theta}$.

Thus, transactions that have at least $\frac{2\theta z}{1+\theta}$ items in common with a transaction t are the transactions whose similarity to t exceeds θ. The number of these transactions is approximatively $\binom{mc}{\frac{1-\theta}{1+\theta}s}$. This implies that the number of neighbors for a transaction in C_i is approximatively $|C_i|^{\frac{1-\theta}{1+\theta}}$ and $f(\theta) = \frac{1-\theta}{1+\theta}$. Note that when $\theta = 1$ a transaction has only itself as a neighbor and, since $f(\theta) = 0$, the expected number of links is $|C_i|$.

ROCK has three phases:

- a random sample is drawn from the data;
- a hierarchical clustering algorithm is applied to the sample data;
- the clusters involving the sample points are used to assign the points to the appropriate clusters.

The dissimilarity between clusters is defined as

$$g(C_i, C_j) = \frac{\text{links}(C_i, C_j)}{|C_i + C_j|^{1+2f(\theta)} - |C_i|^{1+2f(\theta)} - |C_j|^{1+2f(\theta)}}$$

The pair of clusters for which this measure is maximal is the best pair to merge which corresponds to the intuitive idea that the pairs of clusters with large number of cross links are, in general good candidates for merging.

The computation of links is the most costly part of ROCK. Let A be an $n \times n$-binary matrix, where

$$a_{ij} = \begin{cases} 1 & \text{if } p_i \text{ is a neighbor of } p_j, \\ 0 & \text{otherwise.} \end{cases}$$

The number of links between a pair of objects p_i and p_j is $\sum_{k=1}^{n} a_{ik} a_{kj}$. Thus, the problem of computing the number of links amounts to computing A^2 and this requires a time of $O(n^3)$, although slightly faster methods exist which yield an answer in $O(n^{2.37})$. This makes ROCK of limited use for large data sets.