ROCK: A Robust Clustering Algorithm for Categorical Attributes

S. Guha, R. Rastogi and K. Shim

School of informatics

Introduction

- Clustering, traditional approaches.
- The ROCK algorithm.
- Experiments.
  - Artificial dataset.
  - Real-world datasets.

Aim: Cluster Items with non-Numerical Attributes

- Clustering: Group similar items together, keep disimilar items apart.
- We are interested in clustering based on non-numerical data—catagorical/boolean attributes.

  Catagorical: { black, white, red, green, blue }
  Boolean: { true, false }

- Boolean attributes are mearly a special case of catagorical attributes.

An Example Problem

- Supermarket transactions.
- Each datapoint represents the set of items bought by a single customer.
- We wish to group customers so that those buying similiar types of items appear in the same group, e.g:
  Group A— baby-related: diapers, baby-food, toys.
  Group B— expensive imported foodstuffs.
  etc...
- Reperesent each transaction as a binary vector in which each attribute reperesents the presence or absence of a particular item in the transaction (boolean).

Partitional Clustering

- Attempt to divide the points into k clusters so as to optimise some function, E.

- A common approach is to mimimise the total (Euclidian) distance between each point and its clusters center:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} |x - \mu_i|$$

- e.g. k-Means.

(Agglomerative) Hierarchical Clustering

- Start with all items in their own clusters.
- Repeatedly merge the two clusters that are the 'closest', based on some similarity measure.
- Commonly examples are centroid-based methods— merge the two clusters whos centers are the closest.

Clustering with Boolean Attributes

- This all works fine for numerical data, but how do we apply it to, for example, our transaction data?

- Simple approach: Let true $= 1$, false $= 0$ and treat the data as numeric.

- An example with hierarchical clustering:

$A = (1, 0, 0, 0, 0)$ $\quad$ $B = (0, 0, 0, 0, 1)$ $\quad$ $C = (1, 1, 1, 1, 0)$

$|A - B| = \sqrt{2}$ $\quad$ $|A - C| = \sqrt{3}$ $\quad$ $|B - C| = \sqrt{5}$

A and B will merge but they share no items, whilst A and C do.

Clustering with Boolean Attributes

- This all works fine for numerical data, but how do we apply it to, for example, our transaction data?

- Simple approach: Let true $= 1$, false $= 0$ and treat the data as numeric.

- Dosen't work very well. Other problems:
  - We will end up with long vectors that have only a few non-zero coordinates.
  - Two transactions A and B may be similar in that they contain many items of the same type, but have no individual items in common. Gets worse with large clusters.

Clustering with Boolean Attributes

- Need a better similarity measure, one suggestion is the Jaccard coefficient:

$$J(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

- Merge clusters with the most similar pair of points/highest average similarity.
- Considers only the similarity of two points in isolation, does not consider the neighbourhood of the points.
- Can fail when clusters are not well-seperated, sensitive to outliers.

Neighbours and Links

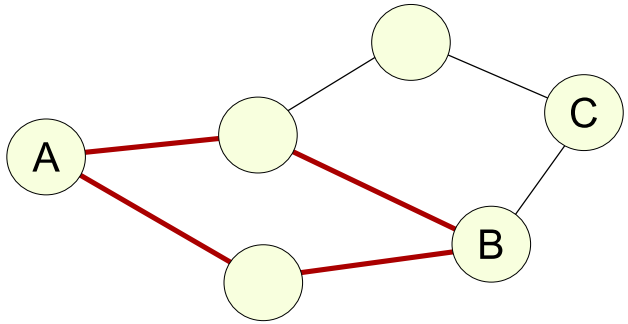
- Need a more global approach that considers the links between points.
- Use common neighbours to define links.
- If point A neighbours point C, and point B neighbours point C then the points A and B are linked, even if they are not themselves neighbours.
- If two points belong to the same cluster they should have many common neighbours.
- If they belong to different clusters they will have few common neighbours.

Neighbours and Links

- We need a way of deciding which points are 'neighbours'.
- Define a similarity function, $sim(p_1, p_2)$, that encodes the level of similarity ('closeness') between two points.
- Normalise so that $sim(p_1, p_2)$ is one when p_1 equals p_2 and zero when they are completely dissimilar.
- We then consider p_1 and p_2 to be 'neighbours' if $sim(p_1, p_2) \geq \theta$, where θ is a user-provided paramater.

Neighbours and Links

- Then define $link(p_1, p_2)$ to be the number of common neighbours between p_1 and p_2.



- The similarity function can be anything— Euclidian distance, the Jaccard coefficient, a similarity table provided by an expert, etc . . .

- For supermarket transactions use the Jaccard coefficient.

The Criterion Function

- We characterise the best set of clusters through the use of a criterion function, E_l— the best set of clusters is that which maximises E_l.
- First approach— maximise the number of links between pairs of points in each cluster:

$$E_l = \sum_{i=1}^{k} \sum_{p_q, p_r \in C_i} link(p_q, p_r)$$

- Keeps points that share many links in the same cluster . . .
- . . . but does not force points with few links to split into different clusters.
- May end up with all points in one big cluster.

The Criterion Function

- Improved approach— divide the actual number of links by the *expected number of links*.
- Prevents points with few links being placed in the same cluster.
- If we add a new point the number of expected links increases, so if the new point has few links E_l will decrease.
- Define a function $f(\theta)$, such that a point belonging to a cluster of size n has approximatelys $n^{f(\theta)}$ neighbours in the cluster.
- Depends on the dataset/problem, and has to be provided by the user.

The Criterion Function

- The final criterion function:

$$E_l = \sum_{i=1}^{k} n_i \sum_{p_q, p_r \in C_i} \frac{link(p_q, p_r)}{n_i^{1+2f(\theta)}}$$

- Can be hard to find $f(\theta)$, but authors found even fairly inaccurate, but reasonable, functions can provide good results.

- For supermarket transactions use $\frac{1-\theta}{1+\theta}$.

ROCK: RObust Clustering using linKs

- A hierarchical clustering algorith that uses links.
- Define a goodness measure based on the above criterion function:

$$g(C_i, C_j) = \frac{link[C_i, C_j]}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}}$$

- A each step of the algorithm merge the pair of clusters that maximise this function.

Dealing with Catagorical Attributes

- How do we handle catagorical attributes with the possibility of missing data?
- One possible method is to convert them into transactions.
- For each attribute A and each value it can take v construct an item $A.v$ and include it in the transaction if the attribute takes that value.
- If we have a missing value no item will be present.

Outliers

- Outliers will probably have very few or no neighbours, and as such will take little or no part in clustering and can be discarded early on.

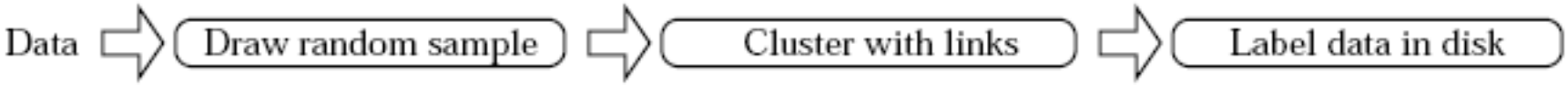
- Small clusters of outliers will persist in isolation until near the end of clustering, so when we are close to reaching the required number of clusters we can stop and weed out any small isolated clusters with little support.

Random sampling

- If we have a huge number of points we can select a random sample with which to do the clustering.

- Once clustering is complete we assign the remaining datapoints from disk by determining which cluster contains the most neighbours to each point (normalised by the expected number of neighbours).

Summary



1. A random sample is drawn from the database.
2. A hierarchical clustering algorithm employing links is applied to the samples.
3. This means: Iteratively merge clusters C_i, C_j that maximise the goodness function
$$g(p_1, p_2) = \frac{\text{total } \# \text{ crosslinks}}{\text{expected } \# \text{ crosslinks}} \quad (1)$$
and stop merging once there are no more links between clusters or the required number of clusters has been reached.
4. Clusters involving only the sampled points are used to assign the remaining data points on disk to the appropriate clusters.

Experimental Results

1 artifical, 3 natural data sets got clustered with Rock and compared to the traditional Clustering Algorithm. For Rock:

- In all of the experiments the Jaccard similarity function was used.
- Expected number of links was approximated using $f(\theta) = \frac{1-\theta}{1+\theta}$.

For Hierarchical Clustering:

- Categorical attributes were converted to boolean attributes with $0/1$ values.
- New attribute $= 1$ iff "value for the original categorical attribute" $=$ "value corresponding to the boolean attribute", else 0
- Outlier handling performed by eliminating clusters with only one point when the number of clusters reduces to $\frac{1}{3}$ of the original number

Synthetic Data Set

- Market basket database containing 114586 transactions.
- Of these, 5456 (arround 5%) are outliers, while the others belong to one of 10 clusters with sizes varying between 5000 and 15000.
- How did these transactions get constructed?

# Cluster	1	2	3	4	5	6
# Transactions	9736	13029	14832	10893	13022	7391
# Items	19	20	19	19	22	19

# Cluster	7	8	9	10	Outliers
# Transactions	8564	11973	14279	5411	5456
# Items	19	21	22	19	116

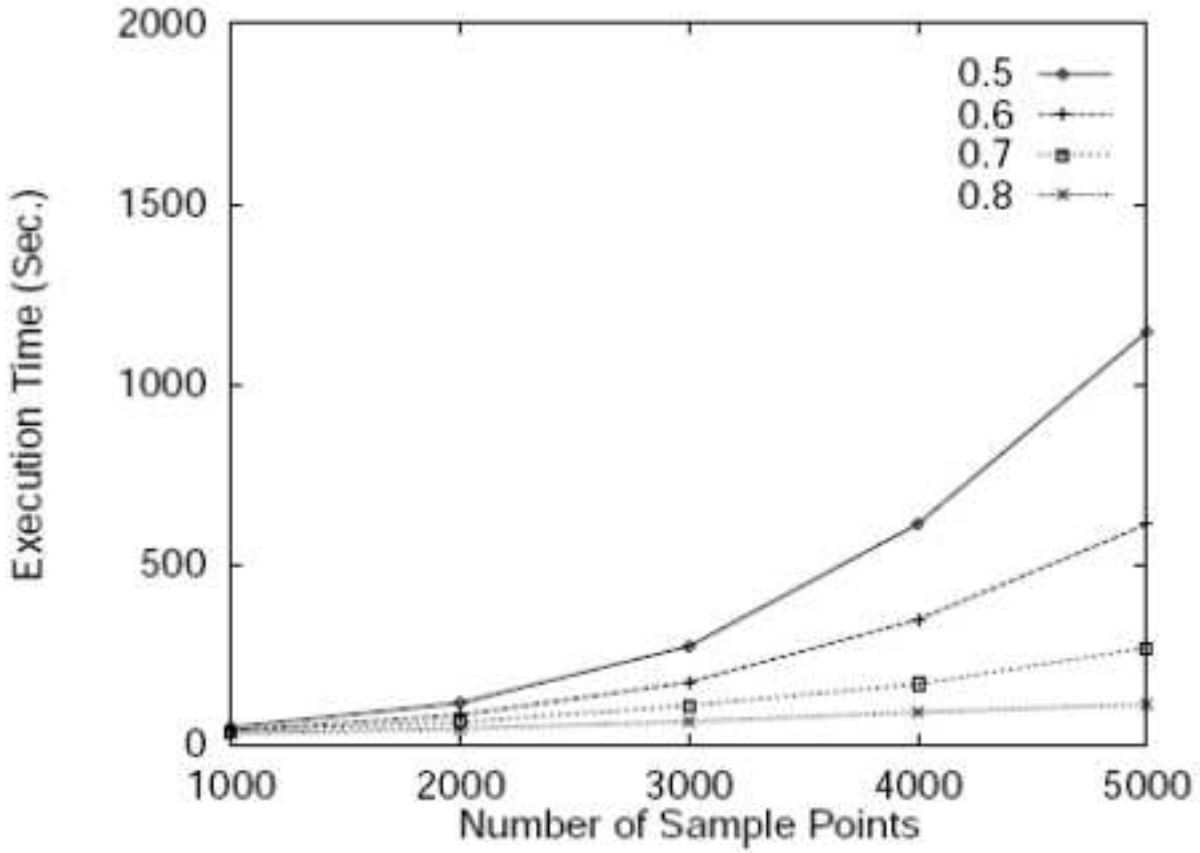
Synthetic Data Set

- Clusers are defined by the items its transactions hold.
- arround 40% of these items in a cluster are common with items for other clusters, arround 60% exclusive to an cluster.
- A transaction for a cluster is generated by randomly selecting items from the set of items that define the cluster.
- Outliers are generated by randomly selecting from among the items for all the clusters.
- The transaction size parameter has a normal distribution with an average value of 15. Due to the normal distribution, 98% of transactions have sizes between 11 and 19.

Scalability

- Using random sampling results a greatly reduced impact of data size on the execution time of ROCK.
- Which impact does the sample size have on the execution time (excl. labelling)?
- Random sample size is varied for four different settings of θ (the "threshold of neighbourhood").

Scalability with Respect to Random Sample Size



Scalability

- The computational complexity of ROCK is roughly quadratic with respect to the sample size.
- For a given sample size, the performance of ROCK improves as θ is increased.
- Why?

Scalability

- The computational complexity of ROCK is roughly quadratic with respect to the sample size.
- For a given sample size, the performance of ROCK improves as θ is increased.
- Why?
- The reason for this is that as θ is increased, each transaction has fewer neighbours and this makes the computation of links more efficient.

Quality

- Number of transactions misclassified by ROCK for our synthetic data set with θ values of 0.5 and 0.6 and a range of sample sizes:

Sample Size	1000	2000	3000	4000	5000
$\theta = 0.5$	37	0	0	0	0
$\theta = 0.6$	8123	1051	384	104	8

- Note that the quality of clustering is better with $\theta = 0.5$ than with $\theta = 0.6$.

- Why?

Quality

- Random sample sizes we consider range from being less than 1% of the database size to about 4.5%.
- Transaction sizes can be as small as 11, while the number of items defining each cluster is approximately 20.
- A high percentage (roughly 40%) of items in a cluster are also present in other clusters. Thus, a smaller similarity threshold is required to ensure that a larger number of transaction pairs from the same cluster are neighbours.

Real-Life Data Sets

- 3 Real-Life Data Sets:

Data Set	Congressional Votes	Mushroom	U.S. Mutual Fund
# Records	435	8124	795
# Attributes	16	22	548
Missing Values	Yes (very few)	Yes (very few)	Yes
Note	Republicans and 267 Democrats	4208 edible and 3916 poisonous	Jan 4, 1993 - Mar 3, 1995

Table 1: Data Sets

Congressional Votes

"The Congressional voting data set was obtained from the UCI Machine Learning Repository. It is the United States Congressional Voting Records in 1984. Each record corresponds to one Congress man's votes on 16 issues (e.g., education spending, crime). All attributes are boolean values, and very few contain missing values. A classification label of Republican or Democrat is provided with each data record. The data set contains records for 168 Republicans and 267 Democrats."

Congressional Votes on the Rock

- Results ROCK with $\theta = 0.73$ and Hierarchical Clustering Algorithm with centroid-based distance function:

Traditional Hierarchical Clustering Algorithm		
Cluster No	No of Republicans	No of Democrats
1	157	52
2	11	215
ROCK		
Cluster No	No of Republicans	No of Democrats
1	144	22
2	5	201

Table 2: Clustering Result for Congressional Voting Data

Congressional Votes on the Rock

- Both identify two clusters one containing a large number of republicans and the other containing a majority of democrats.
- However, in the cluster for republicans found by the traditional algorithm, around 25% of the members are democrats, while with ROCK, only 12% are democrats.
- Why?

Congressional Votes on the Rock

- Both identify two clusters one containing a large number of republicans and the other containing a majority of democrats.
- However, in the cluster for republicans found by the traditional algorithm, around 25% of the members are democrats, while with ROCK, only 12% are democrats.
- Why?
- Improvement mainly caused by outlier removal scheme and the usage of links by ROCK.

Congressional Votes on the Rock

Interestingly, the traditional algorithm also discovered the clusters easily. Reasons for this are:

- Only on 3 issues did a majority of Republicans and Democrats cast the same vote.
- On 12 of the remaining 13 issues, the majority of the Democrats voted differently from the majority of the Republicans.
- On each of the 12 issues, the Yes/No vote had sizable support in their respective clusters.
- Therefore the two clusters are quite well-separated.
- Furthermore, there isn't a significant difference in the sizes of the two clusters.

Mushroom

"The mushroom data set was also obtained from the UCI Machine Learning Repository. Each data record contains information that describes the physical characteristics (e.g., color, odor, size, shape) of a single mushroom. A record also contains a poisonous or edible label for the mushroom. All attributes are categorical attributes; for instance, the values that the size attribute takes are narrow and broad, while the values of shape can be bell, at, conical or convex, and odor is one of spicy, almond, foul, fishy, pungent etc. The mushroom database has the largest number of records (that is, 8124) among the real-life data sets we used in our experiments. The number of edible and poisonous mushrooms in the data set are 4208 and 3916, respectively."

Mushroom on the Rock

ROCK with $\theta = 0.8$					
Cluster #	# Edible	# Poisonous	Cluster #	#Edible	#Poisonous
1	96	0	12	48	0
2	0	256	13	0	288
3	704	0	14	192	0
4	96	0	15	32	72
5	768	0	16	0	1728
6	0	192	17	288	0
7	1728	0	18	0	8
8	0	32	19	192	0
9	0	1296	20	16	0
10	0	8	21	0	36

Mushroom on the Rock

- ROCK found 21 clusters instead of 20: no pair of clusters among the 21 clusters had links between them and so ROCK could not proceed further.
- All except one (Cluster 15) of the clusters discovered by ROCK are pure clusters in the sense that mushrooms in every cluster were either all poisonous or all edible.
- There is a wide variance among the sizes of the clusters: 3 clusters have sizes above 1000 while 9 of the 21 clusters have a size less than 100.
- The sizes of the largest and smallest cluster are 1728 and 8, respectively.

Mushroom on the Rock

- In general, records in different clusters could be identical with respect to some attribute values.
- Thus, every pair of clusters generally have some common values for the attributes
- Thus clusters are not well-separated.
- What does this mean for the traditional approach?

Mushroom Traditional

Traditional Hierarchical Clustering Algorithm, cluster # set to 20					
Cluster #	# Edible	# Poisonous	Cluster #	#Edible	#Poisonous
1	666	478	11	120	144
2	283	318	12	128	140
3	201	188	13	144	163
4	164	227	14	198	163
5	194	125	15	131	211
6	207	150	16	201	156
7	233	238	17	151	140
8	181	139	18	190	122
9	135	78	19	175	150
10	172	217	20	168	206

Mushroom on the Rock

Observing these results we find that:

- Points belonging to different clusters are merged into a single cluster and large clusters are split into smaller ones
- None of the clusters generated by the traditional algorithm are pure.
- Every cluster contains a sizable number of both poisonous and edible mushrooms
- Sizes of clusters detected by traditional hierarchical clustering are fairly uniform: More than 90% of the clusters have sizes between 200 and 400, and only 1 cluster has more than 1000 mushrooms.

Mushroom on the Rock

So the quality of the clusters generated by the traditional algorithm was very poor. Reasons for this are:

- Clusters are not well-separated and there is a wide variance in the sizes of clusters.
- Cluster centers tend to spread out in all the attribute values and lose information about points in the cluster that they represent.
- Thus - as discussed earlier - distances between centroids of clusters become a poor estimate of the similarity between them.

US Mutual Funds

"We ran ROCK on a time-series database of the closing prices of U.S. mutual funds that were collected from the MIT AI Laboratories' Experimental Stock Market Data Servery. The funds represented in this dataset include bond funds, income funds, asset allocation funds, balanced funds, equity income funds, foreign stock funds, growth stock funds, aggressive growth stock funds and small company growth funds. The closing prices for each fund are for business dates only. Some of the mutual funds that were launched later than Jan 4, 1993 do not have a price for the entire range of dates from Jan 4, 1993 until Mar 3, 1995. Thus, there are many missing values for a certain number of mutual funds in our data set. (...) This makes it difficult to use the traditional algorithm since it is unclear as to how to treat the missing values in the context of traditional hierarchical clustering."

US Mutual Funds on the Rock

Mutual Funds Clusters generated with ROCK, $\theta = 0.8$			
Cluster Name	# Funds	Ticker Symbol	Note
Bonds 1	4	BTFTX BTFIX BTTTX BTMTX	Coupon
Bonds 2	10	CPTNX FRGVX VWESX FGOVX PRCIX	-
Bonds 3	24	FMUIX SCTFX PRXCX PRFHX VLHYX	Municipal
Bonds 4	15	FTFIX FRHIX PHTBX FHIGX FMBDX	Municipal
Bonds 6	3	VFLTX SWCAX FFLIX	Municipal
Bonds 7	26	WPGVX DRBDX VUSTX SGZTX PRULX	Income
Financial Service	3	FIDSX FSFSX FSRBX	-
Precious Metals	10	FDPMX LEXMX VGPMX STIVX USERX	Gold
International 2	4	PRASX FSEAX SCOPX	Asia

US Mutual Funds on the Rock

- The Financial Service cluster has 3 funds: Fidelity Select Financial Services (FIDSX), Invesco Strategic Financial Services (FSFSX) and Fidelity Select Regional Banks (FSRBX) that invest primarily in banks, brokerages and financial institutions.

- The cluster named International 2 contains funds that invest in South-east Asia and the Pacific rim region; they are T. Rowe Price New Asia (PRASX), Fidelity Southeast Asia (FSEAX), and Scudder Pacific Opportunities (SCOPX).

- The Precious Metals cluster includes mutual funds that invest mainly in Gold.

US Mutual Funds on the Rock

- It appears that ROCK can also be used to cluster time-series data.
- It can be employed to determine interesting distributions in the underlying data even when there are a large number of outliers that do not belong to any of the clusters, as well as when the data contains a sizable number of missing values.
- A nice and desirable characteristic of this technique: it does not merge a pair of clusters if there are no links between them.
- Thus, the desired number of clusters input to ROCK is just a hint: ROCK may discover more than the specified number of clusters (if there are no links between clusters) or fewer (in case certain clusters are determined to be outliers and eliminated).

Remarks

- A new concept of links to measure the similarity/proximity between a pair of data points with categorical attributes is investiaged.
- The robust hierarchical clustering algorithm ROCK employs links and not distances for merging clusters.
- This method naturally extend to non-metric similarity measures that are relevant in situations where a domain expert/similarity table is the only source of knowledge.
- The results of the experimental study with real-life data sets is encouraging.

Email & End

John-Paul Cunliffe wrote:
(...) If there is any relevant information not covered in your paper, I would appreciate any hint you can give me on it so I can present your work as complete as possible. (...)

Sudipto Guha:

(...) We started out to solve a problem, I believe the problem was solved and we (I) moved on. That's that. ROCK does work quite well in practice, I have even seen being used on environmental data where the categories were anonymized and the algorithm gave correct answers.

As far as I am aware other researchers have tried to take the research to the next step, in terms of optimizations of various factors. (...)