

# The Relevant-Set Correlation Model for Data Clustering

Michael E. Houle\*

## Abstract

This paper introduces a model for clustering, the *Relevant-Set Correlation* (RSC) model, that requires no direct knowledge of the nature or representation of the data. Instead, the RSC model relies solely on the existence of an oracle that accepts a query in the form of a reference to a data item, and returns a ranked set of references to items that are most relevant to the query. The quality of cluster candidates, the degree of association between pairs of cluster candidates, and the degree of association between clusters and data items are all assessed according to the statistical significance of a form of correlation among pairs of relevant sets and/or candidate cluster sets. The RSC significance measures can be used to evaluate the relative importance of cluster candidates of various sizes, avoiding the problems of bias found with other shared-neighbor methods that use fixed neighborhood sizes.

## 1 Introduction

The performance and applicability of many classical data clustering approaches often force particular choices of data representation and similarity measure. Some methods, such as  $k$ -means and its variants [14], require the use of  $L_p$  metrics or other specific measures of data similarity; others, such as the hierarchical methods BIRCH [16] and CURE [8], pay a prohibitive computational cost when the representational dimension is high, due to their reliance on data structures that depend heavily upon the data representation. Still others place assumptions on the distribution of the data that may or may not hold in practice. Most methods require at least an initial guess as to the appropriate number of clusters or classes. Such assumptions are particularly problematic for the knowledge discovery process.

Most methods for data clustering use similarity values for two kinds of testing: *comparative*, where the measure is used to decide which of two items  $a$  or  $b$  is more similar to a query item  $q$ ; or *quantitative*, where the value is deemed to be meaningful in its own right — this type of usage includes thresholding or pruning via a triangle inequality. However, quantitative testing is open to bias of several different kinds. For example,

when an  $L_p$  metric such as the Euclidean distance is used as the similarity measure, clusters that form around a small number of key attributes tend to have smaller distances to the cluster mean than for clusters that form around a large number of key attributes, since the variation among key attribute values is typically less when the number of key attributes is small. Other examples of bias (for transaction data) can be found in [9]. Another problem arises when the attribute set is not numerical, due to the need for relative weightings of the different categorical or ordinal attributes. Density-based solutions that rely on absolute thresholding, such as the agglomerative method DBSCAN [6], are particularly sensitive to this form of bias. Quantitative tests may also lead to difficulties when the use of the similarity measure is tentative or experimental, as is often the case when exploring data sets whose nature is not fully understood.

An important approach to clustering that requires only comparative tests of similarity values is the use of so-called *shared-neighbor* information. Here, two items are considered to be well-associated not by virtue of their pairwise similarity value, but by the degree to which their neighborhoods resemble one another. Even in contexts in which similarity values do not have a straightforward interpretation, if two items have a high proportion of neighbors in common (as determined by the similarity measure), it is reasonable to assign the items to the same group. The origins of the use of neighborhood information for clustering can be traced to the shared-neighbor merge criterion of Jarvis and Patrick [13] used in agglomerative clustering. The criterion states that two clusters can be merged if they contain equal-sized subclusters  $A$  and  $B$  such that  $|A \cap B| \geq mk$ , where  $k$  is the size of  $A$  and  $B$ , and  $0 < m \leq 1$  is a fixed merge threshold parameter. The Jarvis-Patrick method does not in itself perform any quantitative tests of similarity values — the similarity measure is used only in the generation of the neighborhood sets, typically by means of queries supported by appropriate data structures. Quantitative tests of similarity can be avoided entirely if the search structure does not depend on them. Such structures do exist: practical examples include some metric data structures [3], as well as the SASH hierarchy for approximate search [12].

\*National Institute of Informatics, Tokyo, Japan, meh@nii.ac.jp

As is often the case with agglomerative clustering methods, the original Jarvis-Patrick merge criterion can result in clusters composed of long chains of subclusters, in which the items at one end of the chain bear little or no resemblance to those at the other end. One influential shared-neighbor clustering method that largely avoids the pitfalls of chaining is ROCK [9]. ROCK is a hierarchical clustering algorithm in which the merge criterion depends on the degree of overlap between neighborhood sets of cluster items, where the neighborhood is defined according to a user-supplied threshold on minimum inter-object similarity (under the assumption that full similarity corresponds to a value of 1, and total dissimilarity to a value of 0). Cluster formation is encouraged when pairs of member items have a high *linkage*, defined as the number of items contained in the common intersection of their neighborhoods. ROCK avoids the chaining problem by assessing linkages over all pairs of cluster items.

Another shared-neighbor clustering method of note is SNN (Shared Nearest Neighbor) [5], derived from the well-known density-based clustering heuristic DBSCAN [6]. In the original DBSCAN, ‘core’ points are identified in regions of high density —  $v$  is declared to be a core point if the number of items within a user-supplied distance  $\epsilon$  from  $v$  meets a user-supplied minimum density threshold  $\delta$ . Any non-core point which has no other core point within the distance  $\epsilon$  is declared to be noise, and disregarded. Any non-noise, non-core points are grouped together with the core points within distance  $\epsilon$ . The clusters formed may have arbitrary shape through chaining; however, the use of core points tends to limit the length of the chains produced. In SNN, the distance threshold  $\epsilon$  is replaced by a neighborhood size threshold  $k$ , and the density threshold  $\delta$  is replaced by a threshold on the sum of the intersections between the  $k$ -neighborhood of  $v$  with the  $k$ -neighborhoods of each of the neighbors of  $v$ . Whereas DBSCAN has the drawback that it cannot find clusters of greatly-differing densities, SNN improves upon DBSCAN in that the shared-neighbor density measure allows for the formation of locally-dense clusters rather than globally-dense ones. However, like DBSCAN, SNN requires a user-supplied neighborhood size  $k$  and minimum shared-neighbor density threshold to determine the ‘core points’ around which a cluster can form — the choices are important, since they heavily bias towards clusters that have a minimum degree of local connectivity.

For data mining, SNN has the very desirable feature that the number of clusters can be automatically determined. However, the user of SNN is required to guess an appropriate neighborhood size  $k$  and minimum shared-

neighbor density threshold. The original Jarvis-Patrick heuristic, as well as ROCK, also require that an appropriate neighborhood size  $k$  be chosen. The value of  $k$  can introduce a very significant bias on the sizes and other characteristics of clusters that can be produced by the methods.

Only relatively recently have shared-neighbor clustering methods been proposed that avoid quantitative testing of the similarity measure while allowing for variation in the sizes of the neighborhoods. The generic Patch Model (PM) and associated PatClust clustering heuristic assumes only the existence of a pairwise similarity measure in order to produce data clusterings [11]. Given a cluster candidate  $C$  of size  $k$ , the Patch Model employs a measure of internal association of cluster candidates equivalent to that of the density measure of SNN. Unlike SNN, however, PM allows the value of  $k$  to vary in a search for critical values at which the density of a neighborhood of size  $k$  exceeds that of a neighborhood of size  $2k$  by a user-supplied minimum threshold value. The set of cluster candidates consists of those neighborhoods satisfying this relative density criterion. Overly-similar candidates are then eliminated, resulting in a collection of candidates with the potential of covering all regions of the data set regardless of their local densities. The method has the advantage that the final number of clusters is naturally determined without the need for intervention on the part of the user. However, PM has several very significant deficiencies:

- Only neighborhood sets centered at data items are considered as candidate clusters, resulting in clusters that are spherical with respect to the similarity measure upon which the rankings are based.
- Differentiation between similar cluster candidates, and elimination of duplicates, is performed in a very conservative and arbitrary fashion. Although the relative density threshold greatly affects the number of clusters produced, there is no satisfactory way for the user to choose an appropriate value for this parameter.
- An inability to cope with large variations of candidate sizes prevents the discovery of clusters of size smaller than roughly 25 items, a serious drawback for many data mining applications.
- The PM density measure is biased towards the formation of smaller clusters over larger ones, as it is less likely in practice for large neighborhoods of items belonging to large clusters to attain the same degree of homogeneity as small neighborhoods of items belonging to smaller clusters.

- The Patch Model does not provide for a partition-style clustering — some clusters may overlap greatly, and many (or even most) items may not be assigned to any cluster.

In this paper, we present the *relevant-set correlation* (RSC) clustering model, a statistical framework for clustering in the absence of explicit knowledge of the nature or representation of the data. Like PM, RSC is a shared-neighbor clustering strategy in which the distances to neighbors are used only for comparative testing, and the sizes of the neighborhoods are allowed to vary. However, unlike PM, the RSC model provides a consistent and comprehensive framework for the assessment of clusterings, based on the statistical significance of a form of set correlation. In particular, the model quantifies the quality of cluster candidates of any arbitrary size, the degree of association between pairs of cluster candidates, and the degree of association between clusters and individual data items. The firm statistical foundation allows RSC to avoid all the drawbacks of PM listed above, while retaining its more attractive features. In particular, the RSC model:

- Allows the discovery of compact clusters with shape more general than can be expressed by a simple neighborhood.
- Can assess the significance of cluster candidates of any size, in such a way as to allow the comparison of any two candidates regardless of their size.
- Requires only two user-supplied parameters, describing the minimum acceptable cluster size, and the size of the maximum acceptable overlap between two clusters. Both of these parameters can be chosen in a natural way with no knowledge of the nature of the data or its distribution. The number of clusters is not specified by the user.
- Can be used to produce either soft (overlapping) or hard (partitional) clusterings, as desired.

In the next section, the details of the RSC model are presented. Heuristic clustering methods based on the model appear in § 3. In § 4, the heuristics are tested on several large, high-dimensional image data sets, and a comparison is provided with several other clustering methods.

## 2 The Relevant-Set Correlation Clustering Model

Let  $S$  be a dataset drawn from some domain  $D$ . For every item  $q \in S$ , we assume the existence of a unique ordering  $(q_1, q_2, \dots, q_{|S|})$  of the items of  $S$ , where  $i < j$

implies that  $q_i$  is deemed more relevant or similar to  $q$  than  $q_j$ . In practical settings, the item most relevant to  $q$  is generally  $q$  itself. Nevertheless, unless otherwise stated, we will not require that  $q_1 = q$ .

The relevancy ranking for  $q$  induces a collection of sets  $Q(q, k) = \{q_1, \dots, q_k\}$  for each choice of set size  $1 \leq k \leq |S|$ . With respect to the ranking, if a dataset query-by-example operation were to be based at item  $q$ ,  $Q(q, k)$  would represent the top- $k$  relevant set.  $Q(q, k)$  can also represent the result of a  $k$ -nearest neighbor ( $k$ -NN) query for  $q$  with respect to some distance measure  $dist : D \times D \rightarrow \mathbb{R}^{\geq 0}$ .

**2.1 Measuring association** Consider any two subsets  $A$  and  $B$  drawn from data set  $S$ , each associated with some underlying concept relevant to the domain. Every item of  $S$  can be associated with a coordinate of a vector space whose dimension is equal to the size of  $S$ . A subset  $A$  of  $S$  can be represented by a zero-one characteristic vector in this space, where a coordinate value of 1 indicates that the corresponding item is a member of  $A$ , and a value of 0 indicates that the item does not belong to  $A$ . Even if no additional information is available regarding the nature of  $A$  and  $B$ , the relationship between  $A$  and  $B$  (and their underlying concepts) can be quantified in terms of the correlation between corresponding coordinates of their characteristic vectors.

For sequences of variables  $(x_1, x_2, \dots, x_n)$  and  $(y_1, y_2, \dots, y_n)$  with means  $\bar{x}$  and  $\bar{y}$ , respectively, the Pearson correlation is given by the following formula [10]:

$$r = \frac{\sum_{i=1}^n x_i y_i - n \bar{x} \bar{y}}{\sqrt{(\sum_{i=1}^n x_i^2 - n \bar{x}^2)(\sum_{i=1}^n y_i^2 - n \bar{y}^2)}}.$$

Applying the formula to the characteristic vectors of sets  $A$  and  $B$ , and noting that  $\sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i = n \bar{x}$  whenever  $x_i \in \{0, 1\}$ , we obtain the following *inter-set correlation* formula:

$$R(A, B) = \frac{|S| \left( \text{CM}(A, B) - \frac{\sqrt{|A||B|}}{|S|} \right)}{\sqrt{(|S| - |A|)(|S| - |B|)}},$$

where

$$\text{CM}(A, B) = \frac{|A \cap B|}{\sqrt{|A||B|}}$$

is the popular *cosine* similarity measure between  $A$  and  $B$  [10]. Note that when the sizes of  $A$  and  $B$  are fixed, the inter-set correlation value tends to the cosine measure as the data set size  $|S|$  increases.

Intuitively speaking, if an item  $v \in A$  is strongly associated with the remaining items of  $A$ , it is likely that the items of  $S$  that are highly relevant to  $v$  also belong to

set  $A$ . Alternatively, if  $A$  as a whole were to have a high degree of internal cohesion, one would expect many if not most of its items to have relevant sets that correlate significantly with one another. These intuitions form the motivation for two intra-set association measures under the RSC model.

The *first-order intra-set correlation* measure quantifies intra-set association as the mathematical expectation of the inter-set correlation between  $A$  and a relevant set of the form  $V = Q(v, |A|)$  based at an item  $v$  selected uniformly at random from  $A$ :

$$\begin{aligned} \text{SR}_1(A) &\triangleq \mathbf{E}[\mathbf{R}(A, V)] \\ &= \frac{1}{|A|} \sum_{v \in A} \mathbf{R}(A, Q(v, |A|)). \end{aligned}$$

An intra-set correlation value of 1 indicates perfect association among the members of  $A$ , whereas a value approaching 0 indicates little or no internal association within  $A$ .

The *second-order intra-set correlation* measure quantifies intra-set association as the expectation of the inter-set correlation between two relevant sets of the form  $V = Q(v, |A|)$  and  $W = Q(w, |A|)$  selected independently and uniformly at random from  $A$ . Although a formulation is possible based only at unordered pairs of distinct items, the following definition will be seen to have useful properties in the context of cluster item ranking:

$$\begin{aligned} \text{SR}_2(A) &\triangleq \mathbf{E}[\mathbf{R}(V, W)] \\ &= \frac{1}{|A|^2} \sum_{v \in A} \sum_{w \in A} \mathbf{R}(Q(v, |A|), Q(w, |A|)). \end{aligned}$$

Again, a value of 1 indicates perfect association among the members of  $A$ , whereas a value approaching 0 indicates little or no internal association within  $A$ .

**2.2 Significance testing** In general, when making inferences involving Pearson correlation, a high correlation value alone is not considered sufficient to judge the significance of the relationship between two variables. When the number of variable pairs is small, it is much easier to achieve a high value by chance than when the number of pairs is large.

During the clustering process, instead of verifying whether or not the intra-set correlation of a candidate set meets a minimum significance threshold, we will more often need to test whether one candidate has a more ‘significant’ intra-set correlation than another. For this, we test against the assumption that each relevant set contributing to the correlation score is independently generated by means of uniform random selection from

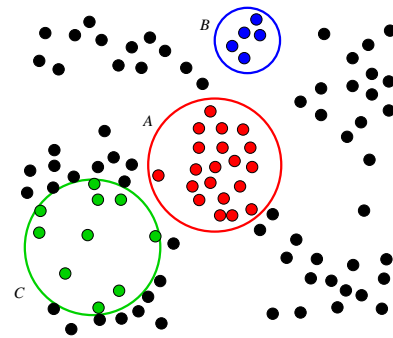


Figure 1: Set  $A$  has smaller first-order intra-set correlation than  $B$ , but is a more significant aggregation.

among the available items of  $S$ . In practice, of course, the relevant sets are far from random. However, this situation serves as a convenient reference point from which the significance of observed correlation values can be assessed. Under the randomness hypothesis, the mean and standard deviation of the correlation score can be calculated (as will be shown below). Standard scores (also known as  $Z$ -scores) [10] can then be generated and compared with one another. The more significant relationship would be the one whose standard score is highest — that is, the one whose correlation exceeds its expected value by the greatest number of standard deviations.

We first analyze the significance of the inter-set correlation for the case where one of the two sets is random. Assume that we are given an arbitrary set  $U \subseteq S$  and a second set  $V$  chosen uniformly at random (without replacement) from the items of  $S$ . Then  $X = |U \cap V|$  is known to be a hypergeometrically-distributed random variable with expectation

$$\mathbf{E}[X] = \frac{|U||V|}{|S|}$$

and variance

$$\mathbf{Var}[X] = \frac{|U||V|(|S| - |U|)(|S| - |V|)}{|S|^2(|S| - 1)}.$$

Noting that  $\mathbf{E}[cX + d] = c\mathbf{E}[X] + d$  and  $\mathbf{Var}[cX + d] = c^2\mathbf{Var}[X]$  for any constants  $c$  and  $d$ , we have the random variable  $\mathbf{R}(U, V)$  has expectation

$$\mathbf{E}[\mathbf{R}(U, V)] = \frac{|S| \left( \frac{\mathbf{E}[|U \cap V|]}{\sqrt{|U||V|}} - \frac{\sqrt{|U||V|}}{|S|} \right)}{\sqrt{(|S| - |U|)(|S| - |V|)}} = 0$$

and variance

$$\mathbf{Var}[\mathbf{R}(U, V)] = \frac{|S|^2 \mathbf{Var}[|U \cap V|]}{|U||V|(|S| - |U|)(|S| - |V|)}$$

$$= \frac{1}{|S|-1}.$$

The expectation and variance of  $R(U, V)$  do not depend on the choice of  $U$  or  $V$  at all, provided that either  $U$  or  $V$  or both are selected uniformly at random from  $S$  (without replacement).

Given any two sets  $A, B \subseteq S$ , we can assess the significance of the correlation value  $R(A, B)$  by normalizing against the assumption that at least one of  $A$  and  $B$  was generated via random selection as above. The significance of the relationship between  $A$  and  $B$  is given by the standard score using mean  $\mu = 0$  and variance  $\sigma^2 = \frac{1}{|S|-1}$ :

$$Z(A, B) \triangleq \frac{R(A, B) - \mu}{\sigma} = \sqrt{|S|-1} R(A, B).$$

Interestingly, since the factor  $\sqrt{|S|-1}$  does not depend on  $A$  or  $B$ , this analysis supports the use of the inter-set correlation alone as a measure of the significance of the relationship between two subsets of  $|S|$ .

Consider next the first-order intra-set correlation value  $SR_1(A)$  of some non-empty subset  $A \subseteq S$ . Let  $\underline{SR}_1(A)$  and  $\underline{SR}_2(A)$  denote the first- and second-order intra-set correlation values for  $A$  under the assumption that for each  $v \in A$ , the relevant set  $Q(v, |A|)$  is independently replaced by a set  $\underline{Q}(v, |A|)$  consisting of  $|A|$  distinct items selected uniformly at random from  $S$ . Then  $\underline{SR}_1(A)$  is a random variable with expectation

$$\mathbf{E}[\underline{SR}_1(A)] = \frac{1}{|A|} \sum_{v \in A} \mathbf{E}[R(A, \underline{Q}(v, |A|))] = 0$$

and variance

$$\begin{aligned} \mathbf{Var}[\underline{SR}_1(A)] &= \frac{1}{|A|^2} \sum_{v \in A} \mathbf{Var}[R(A, \underline{Q}(v, |A|))] \\ &= \frac{1}{|A|(|S|-1)}. \end{aligned}$$

Similarly, one can show that the random variable  $\underline{SR}_2(A)$  has expectation and variance

$$\mathbf{E}[\underline{SR}_2(A)] = 0 \text{ and } \mathbf{Var}[\underline{SR}_2(A)] = \frac{1}{|A|^2(|S|-1)}.$$

The *first-order significance* of  $A$  is defined as the standard score for  $SR_1(A)$  under the randomness hypothesis:

$$\begin{aligned} Z_1(A) &= \frac{SR_1(A) - \mathbf{E}[\underline{SR}_1(A)]}{\sqrt{\mathbf{Var}[\underline{SR}_1(A)]}} \\ &= \sqrt{|A|(|S|-1)} SR_1(A). \end{aligned}$$

Similarly, the *second-order significance* of  $A$  is defined as the standard score for  $SR_2(A)$  under the randomness hypothesis, and equals

$$Z_2(A) = |A| \sqrt{|S|-1} SR_2(A).$$

In the example in Figure 1, the first-order significances of the three sets are  $Z_1(A) = \frac{783}{160} \sqrt{55} \approx 36.29$ ,  $Z_1(B) = 3\sqrt{55} \approx 22.25$ , and  $Z_1(C) = \frac{7}{6} \sqrt{110} \approx 12.24$ . These values conform with our intuition regarding the relative significance of  $A$ ,  $B$  and  $C$ .

The randomness hypothesis, as stated earlier, does not take into account the possibility that the relevant set  $Q(v, |A|)$  may be guaranteed to contain  $v$ . If such a guarantee were provided, the randomness hypothesis could be varied so that  $\underline{Q}(v, |A|)$  comprised  $v$  together with  $|A|-1$  items selected uniformly at random from among the items of  $S \setminus \{v\}$ . Moreover, if the set  $A$  were itself known to be a relevant set of some item  $a \in S$ , then one may opt to select random relevant sets only for the  $|A|-1$  summation terms where  $v \neq a$ . These choices lead to slightly different (and less elegant) formulations of the first- and second-order significance measures, the details of which are omitted here.

### 2.3 Partial significance and cluster reshaping

Within any highly-significant set  $A$ , the contributions of some relevant sets to the intra-set correlation scores may be substantially greater than others. Items whose relevant sets contribute highly can be viewed as better associated with the concept underlying aggregation  $A$  than those whose contributions are small. However, to compare the contributions of a single item with respect to several different sets, or the contributions of several different item-set pairs, a test of significance is again needed.

The contribution to  $SR_1(A)$  attributable to item  $v \in A$  is given by

$$t_1(v|A) \triangleq \frac{1}{|A|} R(A, Q(v, |A|)).$$

The *first-order significance* of the relationship between  $v$  and  $A$  is defined as the standard score for  $t_1(v|A)$  under the randomness hypothesis:

$$Z_1(v|A) = \sqrt{|S|-1} R(A, Q(v, |A|)).$$

The details of the derivation are omitted, as the analysis is essentially the same as that for  $Z_1(A, B)$  in § 2.2, with  $B = Q(v, |A|)$ .

Similarly to the first-order case, the second-order intra-set correlation can be expressed as the sum of contributions attributable to the items of  $A$ :

$$t_2(v|A) \triangleq \frac{1}{|A|^2} \sum_{w \in A} R(Q(v, |A|), Q(w, |A|)).$$

The *second-order significance* of the relationship between  $v$  and  $A$  is defined as the standard score for  $t_2(v|A)$  under the randomness hypothesis. Again, following arguments similar to that of the first-order case, one can derive:

$$Z_2(v|A) = \sqrt{\frac{|S|-1}{|A|}} \sum_{w \in A} R(Q(v, |A|), Q(w, |A|)).$$

Both the first- and second-order significances can be concisely expressed in terms of the sum of their respective partial significances, as follows:

$$(2.1) \quad Z_i(A) = \frac{1}{\sqrt{|A|}} \sum_{v \in A} Z_i(v|A), \quad i \in \{1, 2\}.$$

Partial significances, whether first-order or second-order, can be directly used to rank the items of  $A$  according to their level of association with  $A$ , much like the items of a relevant set are ranked with respect to an individual query item. Moreover, the ranking can be extended to all items of  $S$ , as the definitions of partial significance are meaningful regardless of whether  $v$  is actually a member of  $A$ . In this case,  $A$  can be regarded as a form of *cluster query* that returns a set of items ranked according to  $Z_1(v|A)$  or  $Z_2(v|A)$ . Although in principle  $A$  could be any set of items, the equations (2.1) indicate that the relevancy scores are high only when  $A$  is itself a significant aggregation of items — that is, when  $A$  is itself a ‘reasonably good’ cluster candidate. From the definition of first-order partial significance, ranking according to  $Z_1(v|A)$  is easily seen to be equivalent to ranking according to  $R(A, Q(v, |A|))$  or  $CM(A, Q(v, |A|))$ .

Figure 2 illustrates the first-order cluster query ranking for the point set  $A$  from Figure 1. In this example, the partial significance ranking manages a rough approximation of the original Euclidean distance ranking as measured from a central location within the cluster, despite the lack of knowledge of the individual Euclidean distance values themselves.

It is worth noting that two items lying outside  $A$  ( $y$  and  $z$ ) have higher partial significances than one item contained in  $A$  (item  $x$ ). This suggests that partial significances may be used to ‘reshape’ a candidate cluster set, by replacing poorly-associated members with other, more strongly-associated items, thereby improving the overall cluster quality. Let us consider the situation where  $A$  has been reshaped to yield a new candidate set  $B$ . To assess the quality of  $B$ , the average association can be computed between set  $A$  and relevant sets based at items of the new set  $B$ , instead of at items of  $A$ . The result is a measure of the significance of  $B$  conditioned on the acceptance of  $A$  as

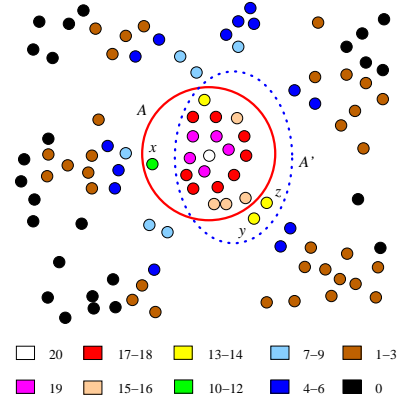


Figure 2: Rankings of points according to first-order partial significance with respect to  $A$ . The value ranges shown are of  $|A \cap Q(v, |A|)|$ , which determines the same ranking as  $Z_1(v|A)$  for fixed  $A$ .

a suitable *pattern*:

$$SR_1(B|A) \triangleq \frac{1}{|B|} \sum_{v \in B} R(A, Q(v, |A|)).$$

The quality of  $B$  can also be assessed according to a second-order intra-set correlation formulation, where the expected correlation value is calculated over pairs of relevant sets, with one relevant set based at an item of  $B$ , and the other based at an item selected from  $A$ :

$$SR_2(B|A) \triangleq \frac{1}{|B|} \sum_{v \in B} \sum_{w \in A} R(Q(v, |A|), Q(w, |A|)).$$

Starting from the intra-set correlation measures, and based on the randomness hypothesis, one can derive the following significance measures for the reshaped set  $B$ . The details of the derivation are omitted, as they are very similar to those of equation (2.1).

$$(2.2) \quad Z_i(B|A) = \frac{1}{\sqrt{|B|}} \sum_{v \in B} Z_i(v|A), \quad i \in \{1, 2\}.$$

An important implication of equation (2.2) is that for any fixed candidate size  $|B| = k$ , the highest possible significance is attained by letting  $B$  consist of those  $k$  items of  $S$  having the highest partial significance values with respect to  $A$ .

Returning to the example of Figure 2, the reshaped candidate set  $A' = (A \cup \{y, z\}) \setminus \{x\}$  has first-order significance value  $Z_1(A'|A) = \frac{137}{56} \sqrt{33} \approx 37.18$ , which is an improvement over the original significance score  $Z_1(A|A) = Z_1(A) \approx 36.29$ . It can be verified that  $A'$  attains the maximum significance score over all possible reshapings of  $A$ .

**2.4 Partition-style clustering** As a result of a clustering process based on the RSC model, individual items may find themselves assigned to more than one cluster, or to none. However, for some applications, in which a ‘hard’ or partition-style clustering is desired, it may be more appropriate to force each item to join only one cluster, and to encourage unassigned items to join the cluster to which it is best associated.

Ideally, the set to which an individual item should be assigned is one which simultaneously achieves a high cluster significance, and a high item-to-cluster significance. Let  $v$  be an item with positive significance scores with respect to clusters  $A$  and  $B$ . The first-order partial significance scores  $Z_1(v|A)$  and  $Z_1(v|B)$  alone are not sufficient to determine which of  $A$  or  $B$  would be a more suitable assignment for  $v$ , since neither score takes into account of which of  $A$  or  $B$  is the more significant grouping. The second-order partial significance scores would provide a better indication, since they take into account the correlations between the relevant set of  $v$  on the one hand, and the relevant sets of all items of the cluster on the other — thus simultaneously expressing both the item-to-cluster relationship and inter-cluster relationship. However, the computation of second-order correlations can be quite expensive.

A more practical alternative (adopted in the experimentation of this paper) would be to consider the product

$$Z_1(v|C^*) \cdot Z_1(C|C^*),$$

where  $C^*$  is the pattern giving rise to cluster candidate  $C$ . Assigning  $v$  to the cluster maximizing this criterion rewards both high item-to-cluster significance and high cluster significance. Moreover, in the cases where the item-to-cluster significance  $Z_1(v|A)$  and  $Z_1(v|B)$  are roughly equal, precedence would be given to the pattern ( $A$  or  $B$ ) of higher significance, which in most cases would be the larger of the two patterns. Thus, items would tend to be assigned to the largest groupings for which the item-to-cluster significance is high. The partition technique is in some sense similar to nearest-neighbor classification (see [4] for a general reference), with inter-set correlations playing the role of distances.

As a byproduct of the RSC cluster reshaping operation, every item whose relevant set intersects a cluster pattern is considered as a potential member of the cluster; the reshaping operation determines whether or not the item is accepted as a cluster member. Regardless of whether or not the item is accepted, the significance of its relationship with the cluster must be computed. This information can be saved with each item as it is generated, to produce a list of associated clusters for the item together with the significances of the associations. Consequently, RSC-based clustering heuristics can be

easily and efficiently adapted to provide a partition-style clustering when desired. For reasons of space, in the description of the RSC-based heuristics to follow, we will describe only the original clustering strategy without the supplemental partition step.

### 3 Clustering Under the RSC Model

**3.1 Scaling via sampling** The heuristic described in this section, *Greedy Relevant Set Correlation* (GreedyRSC), serves as but one example of a practical application of the RSC clustering model. Due to the inherent complexity of computing second-order significance values, in this paper GreedyRSC is described and implemented only in terms of the first-order correlation formulations.

The overall strategy of GreedyRSC resembles that of the PM-based PatClust heuristic method of [11]: both heuristics employ a greedy strategy for cluster selection whereby candidates with the highest quality are selected first, and any candidates found to be overly-similar to a previously-selected candidate are declared to be redundant, and then eliminated. For the sake of efficiency and scalability, both PM and GreedyRSC incorporate the following additional heuristic design choices:

- The avoidance of quadratic cost in cluster quality evaluation by strictly limiting the size of all relevant sets considered to be at most some constant  $b > 0$ .
- The discovery of large-scale clusters by first computing small tentative clusters with respect to a range of data samples of varying sizes.
- The limitation of the number of candidate clusters by using only relevant sets of sample items as the eligible candidate patches or patterns.

PatClust has a very substantial deficiency in that it uses the aforementioned tentative clusters (called *patches*) as estimates of full-sized clusters, but does not provide the full contents of these clusters. GreedyRSC, on the other hand, treats the tentative clusters as *patterns* for the explicit generation of full-sized clusters, by reshaping the tentative clusters with respect to the full dataset using the techniques of § 2.3. Another important feature of GreedyRSC not shared by PatClust is that the former method also seeks to reduce the total size and number of candidate cluster sets generated, by eliminating redundant cluster candidates at intermediate stages of the clustering process.

The use of sampling for RSC-based clustering can be intuitively justified as follows. Let  $C$  be a true (unknown) cluster of high quality, as evidenced by its meeting a high minimum first-order intra-set correlation

threshold. A high quality score implies that the relevant sets of many cluster members are in mutual agreement, so much so that if the set  $C$  were replaced by one of these relevant sets (call it  $C' = Q(q, |C|)$ ), that the remaining elements would likely still be in agreement with it. Restricting the relevant set items (including  $C'$ ) to a sample of the dataset still has the potential for discovering these agreements if the intersections between the relevant sets and the sample are sufficiently numerous. More precisely, we consider relevant sets of size  $t = \frac{m|C|}{n}$  with respect to a sample of size  $m$  taken from the full dataset (of size  $n$ ), and focus our attention on  $C'' = Q''(q, t)$ , where  $Q''(q, t)$  denotes the  $t$  items most relevant to  $q$  within the sample. The intra-set correlation value of  $C''$ , using relevant sets of size  $t$  drawn from the sample, serves as an estimate of the value of  $C$ , using relevant sets of size  $|C|$  drawn from the full dataset. In this fashion,  $C''$  serves as a pattern from which the members of  $C$  can be estimated, by reshaping  $C''$  with respect to the full set as described in § 2.3.

If we are to obey the restriction that all relevant sets be limited in size to at most some constant  $b > 0$ , then in order to discover  $C$ , the sample sizes should be chosen so that for at least one sample, the value  $t$  falls into a constant-sized range. One way of covering all possible values of  $t$  (and thereby allowing the discovery of clusters of arbitrary size) is to create a hierarchy of subsets  $H = \{S_0, S_1, \dots, S_{h-1}\}$  by means of uniform random sampling, such that:

- $S_0$  is identical to  $S$ , and  $S_i \subset S_{i-1}$  for all  $0 < i \leq h - 1$ ;
- the number of samples  $h$  is chosen to be the largest integer such that  $|S_{h-1}| > c$ , for some constant  $c > 0$ ;
- the size of  $S_i$  is equal to  $\lfloor \frac{|S|}{2^i} \rfloor$  for all  $0 \leq i \leq h - 1$ ;
- the pattern sizes  $t$  covered by sample  $i$  fall in the range  $0 < a < t < b$ , where  $a$  and  $b$  are chosen such that  $b > 2a$ .

This last condition ensures that all cluster sizes between  $a$  and  $b2^{h-1}$  are covered by some pattern size with respect to at least one of the samples. Alternatively, if a limit  $K$  is to be set on the maximum cluster size, the number of samples can be determined as  $h = \lceil \log_2 \frac{K}{b} \rceil + 1$ .

To support the sampling heuristic, for each sample  $S_i$ , we assume the existence of an oracle  $O_i$  that accepts any query item  $q \in S$ , and returns a ranked relevant set consisting of  $b$  items of  $S_i$ . The samples sets can optionally be selected and maintained by the oracles themselves.

As a final observation regarding the benefits of sampling, we note that a reasonable restriction on inter-cluster similarity implies that only one pattern need be retained for any given item-sample combination. For any item  $q$ , and defining  $s = |S_i|$ , the correlation between two relevant sets based at a common item is

$$R(Q(q, a), Q(q, b)) = \sqrt{\frac{s-b}{s-a}} \sqrt{\frac{a}{b}},$$

Assume that a maximum threshold value  $\chi$  is placed on the allowable correlation value between any two clusters (including patterns). If  $a \geq b\chi^2$ , and provided that  $s$  is reasonably large compared to  $a$  and  $b$ , then at most one choice of pattern size can be made for any  $q$  with respect to any given sample. For example, the condition essentially holds for the convenient choices  $b = 4a$  and  $\chi < 0.5$ . In the overview of the GreedyRSC method below, we will assume that these parameters have been chosen so as to justify the retention of no more than one pattern per item-sample combination.

### 3.2 The GreedyRSC heuristic

1. For each sample set  $S_i$ , do the following:

- (a) *Relevant sets.*

For each item  $q \in S$ , use oracle  $O_i$  to generate a relevant set  $R_{q,i}$  for  $q$  with respect to the set  $S_i$ , such that  $|R_{q,i}| = b$  for some constant  $0 < b < c$ .

- (b) *Inverted relevant sets.*

Produce a collection of inverted relevant sets  $I_{v,i}$ , where  $q \in I_{v,i}$  if and only if  $v \in R_{q,i}$ .

- (c) *Pattern generation.*

Let  $R_{q,i,t} \subseteq R_{q,i}$  denote the relevant set consisting of the  $t$  highest-ranked items of  $R_{q,i}$ , for any  $0 < t \leq b$ . Compute the value of  $t$  that maximizes the significance score  $Z_1(R_{q,i,t})$  over all  $a \leq t \leq b$ . Let  $P_{q,i}$  be the set at which the maximum is attained. If  $a < |P_{q,i}| < b$  and if the significance score meets the minimum threshold value, then designate  $P_{q,i}$  as the pattern of  $q$  with respect to sample  $S_i$  (otherwise,  $q$  is not assigned a pattern with respect to  $S_i$ ).

- (d) *Redundant pattern elimination.*

Iterate through the patterns of  $S_i$  in decreasing order of significance. For pattern  $P_{v,i}$ , use the inverted relevant sets  $I_{*,i}$  to determine all other lower-ranked patterns sharing items with  $P_{v,i}$  (pattern  $P_{w,i}$  shares an item  $x \in P_{v,i}$  only if  $w \in I_{x,i}$ ). If the inter-set significance score  $Z_1(P_{v,i}, P_{w,i})$  exceeds the maximum threshold, then delete  $P_{w,i}$ .

(e) *Pattern reshaping.*

For every surviving pattern  $P_{v,i}$ , use the inverted relevant sets  $I_{*,i}$  to determine those items  $w$  from the full dataset for which  $R_{w,i,p_v}$  shares members with  $P_{v,i}$ , where  $p_v$  denotes the size of  $P_{v,i}$ . Sort these items in decreasing order of their item-to-set significance with  $P_{v,i}$ , namely  $Z_1(R_{w,i,p_v}|P_{v,i})$ . Let  $C_{v,i,t}$  denote the set consisting of the  $t$  highest-ranked items in this ordering. Reshape the pattern into a cluster candidate set by computing the value of  $t$  that maximizes the significance score  $Z_1(C_{v,i,t}|P_{v,i})$ ; let  $C_{v,i}$  denote this cluster candidate.

(f) *Redundant cluster candidate elimination.*

Iterate through the cluster candidates in decreasing order of the significance  $Z_1(C_{v,i}|P_{v,i})$ . For candidate  $C_{v,i}$ , use inverted cluster membership lists to determine all other lower-ranked candidates sharing items with  $C_{v,i}$ . If the inter-set significance score  $Z_1(C_{v,i}, C_{w,i})$  exceeds the maximum threshold value, then delete  $C_{w,i}$ .

2. *Integration across samples.*

Sort all surviving cluster candidates produced across all samples  $S_i$ , in decreasing order of the significance scores  $Z_1(C_{v,i}|P_{v,i})$ . For candidate  $C_{v,i}$ , use inverted cluster membership lists to determine all other lower-ranked candidates sharing items with  $C_{v,i}$ . If the inter-set significance score  $Z_1(C_{v,i}, C_{w,i})$  exceeds the maximum threshold value, then delete  $C_{w,i}$ .

**3.3 Complexity analysis** Over all executions of step 1(a), a query to the oracle is made for each item of  $S$  with respect to each sample set  $S_i$ . In general, if  $\phi(i)$  represents the average cost of the queries taken over set  $S_i$ , the total cost is proportional to  $n \sum_{i=0}^{h-1} \phi(i)$ . If the oracle is implemented as a distance-based ranking using sequential search, the total number of distances computed would be no more than  $O(n^2)$ . However, if fast approximate search structures are used to limit the cost of an individual query, a lower complexity can be realized. For example, limiting the average query time  $\phi(i)$  to be  $O(b + h - i)$  results in an overall cost of  $O(bn \log \frac{K}{b} + n \log n \log \frac{K}{b})$  distances computed.

Producing the inverted relevant sets in step 1(b) requires a total of  $O(bn \log n \log \frac{K}{b})$  operations. For each item, with respect to each sample, determining the candidate pattern size in step 1(c) requires  $O(b^2)$  operations, for a total of  $O(b^2 n \log \frac{K}{b})$ .

The elimination of redundant patterns in step 1(d)

requires the intersection to be computed between  $P_{v,i}$  and every other pattern containing at least one member of  $P_{v,i}$ , as determined using the inverted lists for the members of  $P_{v,i}$ . If  $\psi_{w,i}$  is the size of the inverted member list for item  $w \in S_i$ , then the total number of contributions to intersections that can be ascribed to  $w$  is no more than  $\psi_{w,i}^2$ . Summing these contributions over all items of  $S_i$ , and noting that the average inverted list size is bounded by  $b$ , we obtain  $\sum_{w \in S} \psi_{w,i}^2 \leq (b^2 + \sigma_i^2)n$ , where  $\sigma_i^2$  is the variance of the sizes of the inverted member lists of members of  $S_i$ . Letting  $\sigma^2 = \frac{1}{h} \sum_{0 \leq i < h} \sigma_i^2$  be the average of these variances, we can bound the total cost of this step by  $O((b^2 + \sigma^2)n \log \frac{K}{b})$ .

The cluster reshaping step 1(e) is performed by finding all patterns  $P_{w,i}$  intersecting  $P_{v,i}$ , computing their correlations with  $P_{v,i}$ , and then sorting the correlations. The bound on the cost of eliminating redundant patterns in step 1(e) also applies to this step, except for the additional work of sorting the accumulated correlations. The total number of items to be sorted for each sample  $S_i$  is at most  $bn$ , the total size of all member lists. The total cost of sorting correlations over all samples is thus  $O(bn \log(bn) \log \frac{K}{b})$ . Since  $\log b$  is of order  $o(\log n)$ , this simplifies to  $O(bn \log n \log \frac{K}{b})$ .

The cost of eliminating redundant cluster candidates in step 1(f) can be accounted for in a similar manner as for patterns in step 1(d), with clusters  $C_{v,i}$  in place of patterns  $P_{v,i}$ . Here, let  $\xi_{v,i}$  be the size of the inverted cluster membership list associated with  $v$  at the time of execution of step 1(f) for sample  $S_i$ . Letting  $\tau_i^2$  be the variance of the values of  $\xi_{v,i}$  over all  $v \in S_i$ , and noting that the average inverted list size remains bounded by  $b$ , we observe that the cost for sample  $S_i$  is of order  $O((b^2 + \tau_i^2)n)$ . Letting  $\tau^2 = \frac{1}{h} \sum_{0 \leq i < h} \tau_i^2$  be the average of the variances over all samples, we obtain a bound for the total cost of this step in  $O((b^2 + \tau^2)n \log \frac{K}{b})$ . The bounds for steps 1(e) and 1(f) also apply to the final candidate pruning performed in step 2.

Overall, disregarding the preprocessing time required for computing relevant sets, the execution time for GreedyRSC is bounded by  $O((b^2 + \sigma^2 + \tau^2)n \log \frac{K}{b} + bn \log n \log \frac{K}{b})$ . The standard deviations  $\sigma_i$  and  $\tau_i$  are typically of the order of their means, which themselves are  $O(b)$ . Accordingly,  $\sigma$  and  $\tau$  can be estimated as roughly  $\tilde{O}(b)$ , for an overall cost bound of  $\tilde{O}(b^2 n \log \frac{K}{b} + bn \log n \log \frac{K}{b})$ . The observed cost is dominated by the computation of relevant sets in step 1(a), and the first phase of redundant cluster candidate elimination in step 1(f).

**3.4 Partitional variants of GreedyRSC** The GreedyRSC method as stated above produces a soft

clustering, where the clusters may overlap and yet not necessarily cover the entire data set. However, for some applications, a hard (partitional) clustering may be more desirable. Here, we briefly describe two methods for converting a soft GreedyRSC clustering into a hard clustering. The first method has already been alluded to in § 2.4, which shows how items correlated with two or more clusters can be assigned to a single candidate. In a straightforward manner, the inverted cluster membership lists produced by GreedyRSC can be used to identify the clusters to which an item has positive correlation, and the rule of § 2.4 can then be applied to decide the best assignment (we shall henceforth refer to this hard clustering heuristic as *RSChard*). However, any item that has not previously been assigned to a cluster by GreedyRSC will remain unassigned.

If the original data vectors are available, the original distance measure can be used to complete the assignment of items to clusters:

1. Compute a clustering of the data using GreedyRSC or *RSChard*, as previously described.
2. For each cluster generated, construct a representative by averaging the data vectors associated with its members.
3. For each data item, determine its closest representative. Reassign the item to the cluster associated with this representative.
4. Optionally, if after reassignment of all data items the size of a given cluster falls below a minimum threshold value, then reassign each of its members to the closest cluster whose size meets the threshold.

Steps 2 and 3 of this method (which we will refer to as *RSCmeans*) together constitute a single iteration of the well-known  $k$ -means clustering algorithm [14, 15]. However, whereas  $k$ -means is initialized with a random collection of cluster representatives of arbitrarily chosen size, and then iteratively converges towards a final grouping of the data, *RSCmeans* relies on the *RSC* model and GreedyRSC heuristic to determine a reasonable estimate of  $k$ . Trusting in GreedyRSC to determine a starting clustering of high quality, the difficulties associated with iteration and convergence of  $k$ -means are avoided by performing only a single iteration of averaging and assignment.

## 4 Experimental Results

The GreedyRSC clustering method and its variants were successfully tested on a number of large data sets of various data types, including journal abstracts,

newspaper articles, still images, video images, and protein sequences. For reasons of space, the details of most of these experiments has been omitted in this version of the paper. Instead, we consider several representative cases with image and categorical data.

**4.1 Image data** The image data experiments contrasted *RSChard* and *RSCmeans* with  $k$ -means and SNN, for two sets drawn from the Amsterdam Library of Object Images (ALOI) [7]. The full ALOI data set (*ALOI-full*) consists of 110,250 images of 1000 common objects taken from a number of different angles under different lighting conditions. The notional object classes are of roughly uniform size, 110. Since the appearance of individual objects varies considerably with the vantage points of the images, almost every class would be expected to generate several clusters in any reasonable clustering.

A subset (*ALOI-var*) of 13,943 images was also generated from *ALOI-full* by selecting objects unevenly from among the classes, with objects of the  $i$ -th object class having approximately  $\frac{40000}{400+i}$  image instances selected. The notional class sizes ranged from 4 to 107, with a median of 7. For both sets, images were represented by dense 641-dimensional feature vectors based on color and texture histograms (for a detailed description of how the vectors were produced, see [2]).

For the GreedyRSC variants, the role of the query oracle was played by a SASH approximate similarity search structure, using the euclidean distance as the pairwise similarity measure. The SASH was chosen due to its ability to handle data of extremely high dimensionality directly, without recourse to dimensional reduction techniques. The maximum pattern size was set to  $b = 100$ . The node degree of the SASH was set to 4. The SASH query performance was then tuned to a speedup of roughly 30 times over sequential search, for a recall rate of approximately 96%. For more details on the SASH search structure and its uses, see [12].

For the implementation, a cluster candidate  $C$  was selected by GreedyRSC only if it met minimum thresholds on the *normalized squared intra-set significance* (NSS), obtained from the set significance  $Z_1(C)$  or reshaped significance  $Z_1(C'|C)$  by dividing by  $\sqrt{|S_i| - 1}$  and then squaring the result; here,  $S_i$  is the sample from which the cluster pattern derives. For the purposes of comparing the significance of clusters derived from the same sample, or for cluster reshaping, the outcome when using the NSS is the same as for the original first-order set significance. However, the NSS is interesting in that it equals  $|C|$  whenever the intra-set correlation of  $C$  equals one. Setting the NSS threshold to a value  $z$  is thus able to produce clusters of size as

small as  $z$ , provided that the relevant sets of their items are in perfect agreement. In the experiments, the minimum GreedyRSC cluster size was chosen to be  $z = 3$ . Cluster similarity was assessed by means of normalized inter-set significance (that is, the inter-set correlation). A maximum threshold correlation value of 0.5 was applied, which corresponds to a maximum tolerated overlap of approximately 50% when the two candidate sets are of equal size.

In the implementation,  $k$ -means was run for varying choices of the number of clusters (denoted by KM- $k$ ). The initial representative sets were generated by taking the best of 5 random selection trials. SNN was tested for different values of neighborhood size  $b$  (denoted by SNN- $b$ ). As the performances varied widely with different choices of merge threshold and number of ‘topics’ (clusters), only the best performances for each considered value of  $b$  are reported (as determined by trial-and-error): a merge threshold of 0.175, and a topic ratio of 0.4 (searching for 44100 clusters).

The partition quality produced by the clustering algorithms was assessed using normalized mutual information (NMI). If  $\hat{L}$  is the random variable denoting the partition sets formed by the clusterer, and  $L$  the random variable corresponding to the true object classes, then the NMI value is defined to be

$$\text{NMI} = 2 \frac{H(L) - H(L|\hat{L})}{H(L) + H(\hat{L})}$$

where  $H(L)$  and  $H(\hat{L})$  are the marginal entropies of  $L$  and  $\hat{L}$ , and  $H(L|\hat{L})$  is the conditional entropy. Simply stated, the NMI corresponds to the amount of information that knowing either variable provides about the other.

The clustering results are shown in Figure 3. The RSChard implementation partitioned ALOI-full into 3520 clusters, with minimum size 3, median size 9, and maximum size 377; RSCmeans reduced the number of clusters to 3517, with median size 18 and maximum size 222. RSCmeans achieved an NMI score significantly better than the best of the three SNN variants — the top-performing SNN variant having its neighborhood size approximately the same as the average class size. For ALOI-var, RSChard produced 859 clusters with minimum size 3, median size 8, and maximum size 270; RSCmeans produced the same number of clusters, but with median size 11 and maximum size 190. Its NMI score was significantly better than that of the top-performing SNN variant (SNN-100). Note that the small average cluster size led SNN to perform very poorly for large neighborhood sizes. The good performance of RSCmeans followed from that of RSChard, which (unlike SNN) was able to assign almost all items

ALOI-full	Time (s)	NMI	Uncl.%
SNN-20	10620	0.737	27.1
SNN-100	11504	0.840	14.1
SNN-200	11938	0.817	9.4
KM-100	1371	0.621	0.0
KM-200	2461	0.687	0.0
KM-400	6393	0.753	0.0
KM-800	6757	0.817	0.0
KM-1600	10378	0.859	0.0
RSChard	5032	0.843	1.5
RSCmeans	6541	0.879	0.0
ALOI-var	Time (s)	NMI	Uncl.%
SNN-20	190	0.658	37.6
SNN-100	203	0.696	18.4
SNN-150	187	0.555	12.4
SNN-170	200	0.314	9.8
SNN-200	214	0.184	8.5
KM-100	122	0.710	0.0
KM-200	234	0.780	0.0
KM-400	262	0.841	0.0
KM-800	478	0.880	0.0
KM-1600	821	0.895	0.0
RSChard	342	0.785	1.2
RSCmeans	384	0.896	0.0

Figure 3: Clustering results for the ALOI data sets. *NMI* denotes the normalized mutual information score; *Uncl.%* denotes the percentage of items not assigned to any cluster.

to a cluster while still achieving good classification rates.

Overall, the results demonstrate both the inability of ‘fixed-sized’ shared-neighbor methods (as represented by SNN) to perform consistently well for sets with variable cluster sizes, and the difficulty of estimating parameters such as neighborhood sizes (SNN) and numbers of clusters (KM). In contrast, RSChard was able to automatically produce high-quality clusterings that were further improved upon by RSCmeans.

**4.2 Categorical data** In their paper, the authors of ROCK reported testing their method on the *Mushroom* categorical data set from the UCI Machine Learning Repository. The data consists of entries for 8124 varieties of mushroom, each record with values for 24 different physical attributes (such as color, shape, stalk type, etc.). Every mushroom in the data set is classified as either ‘edible’ (4208 records) or ‘poisonous’ (3916 records). We repeated the experiments of [9] with RSChard; the distance measure used for both data sets was a straightforward mismatch count, and attributes for which values were missing were treated as a mismatch in the similarity assessment.

The results of the classification are shown in Fig-

Class	GreedyRSC		ROCK	
	Size	Errors	Size	Errors
edible	1728	0	1728	0
poisonous	1728	0	1728	0
poisonous	1296	0	1296	0
edible	768	0	768	0
edible	512	0		
edible	192	0	704	0
poisonous	288	0	288	0
edible	288	0	288	0
poisonous	256	0	256	0
poisonous	192	0	192	0
edible	192	0	192	0
edible	192	0	192	0
edible	97	1		
poisonous	7	0	96	0
edible			8	0
poisonous				
poisonous	97	25		
edible	7	0		
poisonous			104	32
edible	96	0	96	0
edible	88	40		
edible			48	0
poisonous			32	0
poisonous			8	0
edible	48	0	48	0
poisonous	36	0	36	0
edible	16	0	16	0
totals	8124	66	8124	32

Figure 4: Cluster set sizes and classification results for the Mushroom set.

ure 4. Despite the genericity of the method, GreedyRSC achieved a classification rate almost equal to that of ROCK, with striking correspondances among the cluster sizes and compositions. Both greatly outperformed the traditional heirarchical algorithm implemented in [9], which produced 20 clusters within which 3432 out of 8124 items were misclassified. It should be noted that whereas ROCK required an estimate of the number of clusters to be provided, GreedyRSC was able to automatically determine this number.

## 5 Conclusion

The RSC model has many important and distinctive features, all recognized as important requirements of clustering for data mining applications [10]:

- The ability to scale to large data sets, in terms of the numbers of both items and attributes.
- Genericity, in its ability to deal with different types of attributes (categorical, ordinal, spatial) assuming that an appropriate similarity measure is provided.
- Other than the provision of an appropriate similarity measure, no special knowledge of the data is required in order to determine input parameters. In particular, the number of output clusters is determined automatically.

As evidenced by the RSCmeans clustering variant, RSC is well-suited for hybridization with other clustering methods. RSC clustering heuristics can also serve as a good initial estimators of parameters for more traditional mining and analysis techniques.

## References

- [1] R. Agrawal and R. Srikant, *Fast algorithms for minning association rules*, Proc. 20th VLDB Conf., Santiago, Chile, 1994, pp. 487–499.
- [2] N. Boujemaa, J. Fauqueur, M. Ferecatu, F. Fleuret, V. Gouet, B. Le Saux and H. Sahbi, *IKONA: Interactive Generic and Specific Image Retrieval*, Proc. Intern. Workshop on Multimedia Content-Based Indexing and Retrieval (MMCBIR), Rocquencourt, France, 2001.
- [3] E. Chávez, G. Navarro, R. Baeza-Yates and J. L. Marroquín, *Searching in metric spaces*, ACM Comput. Surv., 33 (2001), pp. 273–321.
- [4] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, Wiley, New York, NY, USA, 2001.
- [5] L. Ertöz, M. Steinbach and V. Kumar, *Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data*, Proc. 3rd SIAM Intern. Conf. on Data Mining (SDM), San Francisco, CA, USA, 2003.
- [6] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise*, Proc. 2nd Int. Conf. on Knowl. Discovery and Data Mining (KDD), Portland, OR, USA, 1996, pp. 226–231.
- [7] J. M. Geusebroek, G. J. Burghouts and A. W. M. Smeulders, *The Amsterdam library of object images*, Int. J. Comput. Vision 61 (2005), pp. 103–112.
- [8] S. Guha, R. Rastogi and K. Shim, *CURE: an efficient cluster algorithm for large databases*, Proc. ACM SIGMOD Conf. on Management of Data, New York, USA, 1998, pp. 73–84.
- [9] S. Guha, R. Rastogi and K. Shim, *ROCK: a robust clustering algorithm for categorical attributes*, Inform. Sys. 25 (2000), pp. 345–366.
- [10] J. Han and M. Kamber, *Data Mining: Concepts and Techniques* (2nd ed.), Morgan Kaufmann, San Francisco, CA, USA, 2006.
- [11] M. E. Houle, *Navigating massive data sets via local clustering*, Proc. 9th ACM SIGKDD Conf. on Knowl. Disc. and Data Mining (KDD), Washington DC, USA, 2003, pp. 547–552.
- [12] M. E. Houle and J. Sakuma, *Fast approximate similarity search in extremely high-dimensional data sets*, Proc. 21st IEEE Int. Conf. on Data Eng. (ICDE), Tokyo, Japan, 2005, pp. 619–630.
- [13] R. A. Jarvis and E. A. Patrick, *Clustering using a similarity measure based on shared nearest neighbors*, IEEE Trans. Comput. C-22 (1973), pp. 1025–1034.
- [14] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: an Introduction to Cluster Analysis*, John Wiley & Sons, New York, USA, 1990.
- [15] J. McQueen, *Some methods for classification and analysis of multivariate observations*, Proc. 5th Berkeley Symp. on Math. Statistics and Probability, 1967, pp. 281–297.
- [16] T. Zhang, R. Ramakrishnan and M. Livny, *BIRCH: an efficient data clustering method for very large databases*, Proc. ACM SIGMOD Conf. on Management of Data, Montréal, Canada, 1996, pp. 103–114.