

The Iterative Extraction Approach to Clustering

Boris Mirkin

School of Computer Science and Information Systems,
Birkbeck, University of London, UK,
mirkin@dcs.bbk.ac.uk

Summary. The Iterative Extraction approach (ITEX) extends the one-by-one extraction techniques in Principal Component Analysis to other additive data models. We describe additive models for clustering entity-to-feature and similarity data and apply ITEX for deriving computationally feasible clustering solutions. Specifically, two ITEX derived clustering methods, iK-Means and ADDI-S, are presented as well as update results on theoretical, experimental and applicational aspects of these methods.

6.1 Introduction

The iterative extraction approach emerged within the Principal Component Analysis (PCA) framework. The PCA builds aggregated features, “hidden factors” to score hidden capabilities of entities, relying on the Singular Value Decomposition of the data matrix and its mathematical properties: The singular vectors, that underlie the principal components, are mutually orthogonal, thus can be found and extracted one by one in the descending order of singular values. Moreover, the square of a singular value represents the share of the data scatter taken into account by the corresponding principal component.

The author extended this approach to clustering in [30, 31, 32, 33, 34] by extending the bilinear Singular Value Decomposition model to that of clustering. Specifically, the “scoring” principal components sought are compulsory restricted to be binary, thus representing cluster memberships rather than scores. This extension proved to be reasonable since it encompasses many a popular method including *K*-Means clustering. An analogous extension, of the spectral decomposition of a square symmetric semi-positive matrix applied to similarity data also brings forward clusters that are provably tight.

The ITeRative EXtraction approach applied in this setting finds clusters one by one, which has a number of advantages as well as drawbacks. Among advantages are the following: (i) computational efficiency, (ii) the possibility of the additive decomposition of the data scatter in such a way that the

contribution of each cluster, as well as of its elements, can be clearly evaluated, (iii) the easiness of getting overlapping clusters by not bothering to take into account the clusters already found. This approach will be abbreviated further on as ITEX; in [32, 33] it was referred to as SEFIT. Some methods emerging within ITEX, such as iK-Means and ADDI-S, are described in the further text. The data scatter decompositions proved effective in extending clustering methods to categorical and mixed scale data as well as for interpretation of the results (see [34]). Here we concentrate on reviewing methods and some applications, and do not consider usage of the decompositions.

The paper is organized as follows. Section 2 describes the ITEX approach for the entity-to- feature data and methods of Anomalous Pattern and iK-Means clustering following from it. Method iK-Means can be considered as a version of the conventional K -Means in which the number of clusters nor cluster seeds are not pre-specified but found sequentially by extracting “principal” clusters one by one. An experiment over generated data is described involving a number of different approaches to choosing the “right” number of clusters published in the literature. The experiment demonstrates that the iK-Means is superior to other methods, especially if supplemented with Hartigan’s rule for choosing the cluster “discarding threshold.” Section 3 describes the ITEX applied to additive structuring and clustering models over similarity data. The material clearly demonstrates that there have been a bunch of heuristic similarity clustering methods proposed that nicely fit into the framework. A concept that appears to be crucial in this models is the intercept that also can be interpreted as a similarity scale shift or the similarity threshold. Within the ITEX, its value becomes as important as the number of clusters in conventional approaches. With the scale shift specified to be either user-defined or optimal, the ITEX leads to intuitive and fast clustering methods. The final method, ADDI-S, in which all parameters are least-squares adjusted, produces provably tight clusters involving a variable similarity threshold, equal to half the average similarity within the cluster. This method proved useful in applications, three of which are described in brief.

6.2 Clustering Entity-to-feature Data

6.2.1 Principal Component Analysis

The method of iterative extraction extends the process of a a major data analysis tool, the Principal Component Analysis (PCA). Typically, PCA is presented as a heuristic data extraction method [8, 22]. Observed data such as marks of students $i \in I$ at academic disciplines labelled by $v = 1, \dots, V$ constitute a data matrix $X = (x_{iv})$. This matrix is pre-processed by centering and rescaling its columns, after which a normed $|I|$ -dimensional “scoring” vector z is sought such that the linear combination $c = X^T z$ takes into account the maximum possible share of the data scatter $Tr(X^T X)$. This problem reduces

to finding the maximum eigenvalue λ_1 and corresponding normed eigenvector c_1 of the covariance matrix $S = X^T X$, $S c_1 = \lambda_1 c_1$, so that the solution is $z_1 = X c_1 / \sqrt{\lambda_1}$ and λ_1 is the share of the data scatter taken into account by it. Vectors z_1 and c_1 form what is referred to the first Principal Component “scoring” and “loading” vectors, respectively. The second component is found with the same process, but applied to the residual matrix $S' = S - \lambda_1 c_1 c_1^T$ rather than matrix $S = X^T X$ itself. The second Principal Component corresponds to the second largest eigenvalue, its eigenvector being orthogonal to the first one. The process can be reiterated to find the third, the fourth, etc. mutually orthogonal components. The Principal Components are interpreted as “better” features, bearing the maximum possible share of the data scatter, so that a few of them can approximate all the data.

In this narrative, PCA is but a heuristic method for the iterative extraction of the principal components. In fact, as is rather well known, PCA can be considered a method for fitting the model presented in equation (6.1) below.

Assume that each entry x_{iv} reflects the i -th entity scores (the student’s hidden abilities) z_{ik} ($i \in I$) along with feature v impact coefficients c_{kv} , over a number of hidden factors $k = 1, \dots, K$, so that

$$x_{iv} = c_{1v} z_{i1} + \dots + c_{Kv} z_{iK} + e_{iv} \quad (6.1)$$

for all $i \in I$ and $v = 1, \dots, V$, or, in the matrix algebra notation,

$$X = Z_K C_K + E. \quad (6.2)$$

We are interested in finding the least squares solution to equation (6.1) – (6.2), that is, matrices Z_K and C_K minimizing the sum of squared elements of the residual matrix E . What is nice in this formulation is that the components are not defined as linear combinations of X columns, nor they are supposed to be normed; and the criterion is but the conventional statistical approximation of the observed data by the “ideal” data produced by the model.

The least-squares solution is defined only up to a K -dimensional linear subspace of the space of N -dimensional vectors, whose base is formed by columns of matrix Z_K . The optimal linear subspace can be specified in terms of the so-called singular value decomposition (SVD) of matrix X , typically after it is standardized. In fact, matrices Z_K and C_K , whose columns are the first K singular vectors of X , form orthonormal bases of the least-squares optimal subspaces. They can be found by iterative application of the same process of obtaining just one principal component of matrix X by least-squares fitting the one-factor model

$$x_{iv} = c_v z_i + e_{iv} \quad (6.3)$$

with respect to unknown vectors c and z . At each k -th step of the process, matrix X is substituted by the residual data matrix calculated by subtraction of the current component matrix $\mu_k z_k^T c_k$ from the previous X . (The traditional assumptions of SVD are assumed here: μ_k is k -th singular value of X ; z_k , c_k are the normed versions of the singular vectors corresponding to μ_k .)

The conventional process of extracting eigenvectors from $S = X^T X$ described above can be considered an implementation of this method since computation of singular vectors can be performed not necessarily with the matrix X but also with its derivative matrices: $V \times V$ matrix $X^T X$ or $|I| \times |I|$ matrix XX^T , because z_k is an eigen vector of XX^T and c_k an eigen vector of $X^T X$ corresponding to their respective k -th eigen-values, both equal to μ_k^2 ($k = 1, 2, \dots, K$).

6.2.2 Additive Clustering Model and ITEX

Assuming that the score vectors z_1, z_2, \dots, z_K are restricted to be 0/1 binary, the model (6.1) can be reinterpreted as a clustering model [31, 32]. According to this model, binary vector z_k is the membership vector for cluster $S_k \subseteq I$ so that $z_{ik} = 1$ if $i \in S_k$ and $z_{ik} = 0$ if $i \notin S_k$. Vector c_k is a representation of cluster k in the feature space so that every data row $x_i = (x_{iv})$ approximately equals the sum of representative vectors c_k over all such k that $i \in S_k$. In the case when cluster sets S_k are mutually disjoint, that is, vectors z_k are mutually orthogonal, any row x_i approximates just one representative vector c_k .

The clustering problem according to model (6.1) is similar to that of PCA: given matrix X and number K , find binary z_k and real c_k minimizing a prespecified monotonely growing function of the residuals e_{iv} .

For the least-squares criterion, the one-by-one extracting strategy ITEX here builds clusters S_k one by one, each time minimizing one-cluster criterion of the model (6.3):

$$l = \sum_{i \in I} \sum_{v \in V} (x_{iv} - c_v z_i)^2 \quad (6.4)$$

over unknown c_v and binary z_i , index k being omitted. The membership vector z is characterized by subset $S = \{i : z_i = 1\}$. Criterion (6.4) can be rewritten in terms of S :

$$W(S, c) = \sum_{i \in S} d(x_i, c) + \sum_{i \notin S} d(x_i, 0), \quad (6.5)$$

where d is the Euclidean distance squared, $d(x, y) = \sum_j (x_j - y_j)^2$, and x_i is i -th row of the residual data matrix.

This is a conventional clustering square error criterion [20, 34] for a partition consisting of two clusters, S and its complement $\bar{S} = I - S$, with regard to their respective centroids, c and 0. However, in contrast to conventional clustering formulations, the centroid 0 here is not the gravity center of the complementary set $I - S$, but is being kept constant and does not change when S and $I - S$ change.

Given S , the optimal c in (6.5) is obviously the center of gravity of S because the first sum is minimum at that c and the second sum does not depend on c . Given c , a subset S to minimize (6.5) must include every $i \in I$ such that $d(x_i, c) < d(x_i, 0)$. These properties immediately give rise to the following

implementation of the alternating minimization algorithm for criterion (6.5) [34].

1. *Pre-processing.* Specify a reference point $a = (a_1, \dots, a_n)$ (this can be the data grand mean) and standardize the original data table using the reference point coordinates as shift parameters a_v . (This way, the space origin is shifted into a .) In the case when feature scales significantly differ, the standardization should also involve rescaling the feature scales (see details in [34]).
2. *Initial setting.* Put a tentative centroid, c , as an entity which is the most distant from the origin, 0. [This minimizes (6.5) with respect to all singleton clusters.]
3. *Cluster update.* Determine cluster list S around c against the only other “centroid” 0 with the Minimum distance rule so that y_i is assigned to S if $d(y_i, c) < d(y_i, 0)$.
4. *Centroid update.* Calculate the within S mean c' and check whether it differs from the previous centroid c . If c' and c do differ, update the centroid by assigning $c \leftarrow c'$ and return to Step 3. Otherwise, go to 5.
5. *Output.* Output list S and centroid c , with accompanying interpretation aids, as the most anomalous pattern.

The process is illustrated on Figure 6.1.

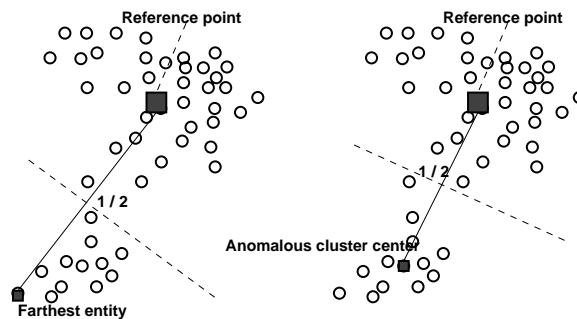


Fig. 6.1. Extracting an “Anomalous Pattern” cluster with the reference point in the gravity center: the initial iteration is on the left and the final one on the right

The AP method is a reference-point based version of the popular clustering method K -Means in which:

- (i) the number of clusters K is 2;
- (ii) centroid of one of the clusters is forcibly kept at the 0 reference point through all the iterations;
- (iii) the initial centroid of the anomalous cluster is taken as an entity point which is the most distant from 0.

6.2.3 Overlapping and Fuzzy Clustering Case

The ITEX can be easily applied to the case of overlapping clusters. After one cluster has been found, the next one can be sought by applying the same AP method to the matrix of residuals $X - zc^T$, in a manner similar to that in PCA.

When clusters in the feature space are well separated from each other or the cluster structure can be thought of as a set of differently contributing clusters, the clusters can be found with the iterative application of Anomalous Pattern algorithm that would mitigate the need for pre-setting the number of clusters and their initial centroids.

Consider, for example, the setting in Table 6.1 which represent a set of entities of which one fourth are ideally located in $c1$, five eighths in $c2$, and the remaining one eighth is assigned to the summary point $c1 + c2$. These represent two clusters comprising $3n$ and $6n$ entities, respectively, in such a way that their intersection consists of n entities. This setting, designed according to [7], perfectly fits into model (6.1) with all residuals being zero. The first three rows of the Table 6.1 can be considered a shortcut for a data table comprising $N = 8n$ entities and 2 features that may have only patterns presented in its lines one ($2n$ entities), two ($5n$ entities) and three (n entities); the fourth line represents the feature values at the grand mean of the data.

Table 6.1. Two overlapping ideal clusters presented by their centroids: $c1$, the first cluster; $c2$, the second cluster; $c1 + c2$, their overlap; c , the grand mean. The column on the right represents the relative numbers of entities 2:5:1, in the first cluster short of the overlap, the second cluster short of the overlap, and the overlap, respectively

Notation	Feature I	Feature II	Quantity
$c1$	12	2	$2n$
$c2$	-1	-2	$5n$
$c1 + c2$	11	0	n
Mean c	2.75	-0.75	

To apply ITEX clustering to this data, let us pre-process each column by subtracting the corresponding component of the grand mean c and dividing by the range (13 for feature I, 4 for feature II) afterwards. This transforms $2n$ entities at $c1$ to $c1' = (0.64, 0.69)$, $5n$ entities at $c2$ to $c2' = (-0.36, -0.31)$ and n entities at $c1 + c2$ to $c3' = (0.56, 0.19)$, which is not the sum of the former two anymore. Still $c3'$ lies not too far from $c1'$, which is picked up by AP as the anomalous starting point. These two make the first AP cluster, which exactly coincides with Cluster 1 in Table 6.1. The next cluster can be found using the residual data matrix that can be computed by finding the centroid of the first AP cluster, $c' = (2c1' + c3')/3 = (0.61, 0.52)$ and subtracting it from all the elements in the AP cluster, thus leading to $c1'' = c1' - c' = (0.03, 0.17)$ and $c3'' = c3' - c' = (-0.05, -0.33)$. Now the anomalous seed is $c2'$ that has been remained intact. Of the two residual centroids, $c1''$ and $c3''$, the former

is closer to 0 while the latter to c_2' . This brings Cluster 2 as the second AP, and leaves all the entries in the next residual matrix close to zero, accounting for less than 5% of the initial standardized data scatter.

Unfortunately, when cluster contributions are less different, ITEX may be not as successful and fail to properly identify the clusters, even in a similarly “ideal” situation when all residuals are zero [7]. For example, with the same data entries as in Table 6.1 but proportions of the entities being equal to 1:1:1 rather than 2:5:1 as in the Table, the ITEX with AP clustering will find not two but three clusters, corresponding to each of the three parts under consideration: the overlapping part of two clusters and the clusters’ parts short of the overlap.

In general, the ITEX AP may tend to produce non-overlapping parts of clusters rather than those in the data structure. This may suggest that this method should be used for disjoint rather than overlapping clustering, which is described in next section. Nonetheless, even in the overlapping case, the ITEX method can provide a useful initialization for other algorithmic strategies such as based on the follow-up iterations of centroid and cluster updates [7]. On the other hand, one may think that the situation may be alleviated if fuzzy rather than crisp belongingness vectors z_k are involved. Indeed, this would make the summary centroid more similar to the averaged one, since the individual cluster centroids would be weighted by memberships that should sum up to unity over the set of centroids. Such a possibility was considered by Mirkin and Satarov [35, 32]. The model (6.1) considered with fuzzy vectors z_k leads to an interesting concept of ideal types. Indeed, with the condition that vectors z_k are not negative and sum up to unity ($k = 1, 2, \dots, K$), all entity points should be convex combinations of centroids c_1, c_2, \dots, c_K , which are thus supposed to be the extreme points of a convex polytope covering the entity set rather than within cluster averages. Applied as is, this may lead to rather wild “ideal type” centroids possibly located quite far away from the entities. A reasonable modification of model (6.1) leading to better fitting fuzzy clusters was proposed and experimentally explored in [42]; this however, utilizes finding all clusters in parallel rather than sequentially. As shown in [42], the modified model retains, in a weaker form, the “ideal type” property, which can be useful in some applications.

6.2.4 *K*-Means and i*K*-Means Clustering

K-Means is one of the most popular clustering methods. It iteratively updates a set of cluster centroids by assigning to them their closest entities and finding the gravity centers of thus obtained clusters. An issue of *K*-Means is a mandatory setting of the *K* and initial seeds for the centroids. Properties of the Anomalous Pattern algorithm mitigate the issue of determining appropriate initial seeds, which allows using it for finding an initial setting for *K*-Means.

Some other potentially useful features of the method relate to its flexibility with regard to dealing with outliers and the “swamp” of inexpressive, ordinary, entities situated around the grand mean.

iK-Means

0. *Setting.* Put $k = 1$ and I_k the original entity set. Specify a threshold of resolution to discard all AP clusters whose cardinalities are less than the threshold.

1. *Anomalous pattern.* Apply AP to I_k to find S_k and c_k .
2. *Control.* If Stop-condition (see below) does not hold, put

$$I_{k+1} \leftarrow I_k - S_k \text{ and } k \leftarrow k + 1 ,$$

and go to Step 1.

3. *Removal of small clusters.* Remove all of the found clusters that are smaller than a pre-specified cluster *discarding threshold* for the cluster size. Denote the number of remaining clusters by K and their centroids by c_1, \dots, c_K .
4. *K-Means.* Do K -Means with c_1, \dots, c_K as initial seeds.

The Stop-condition in this method can be any or all of the following:

1. **All clustered.** $S_K = I_K$ so that there are no unclustered entities left.
2. **Large cumulative contribution.** The total contribution of the first K AP clusters to the data scatter has reached a pre-specified threshold such as 60 %.
3. **Small cluster contribution.** Contribution of k -th AP cluster is too small, say, compared to the order of average contribution of a single entity, $1/N$, where N denotes the total number of entities.
4. **Number of clusters reached.** Number of clusters, k , has reached a pre-specified value K .

The first condition is natural if the data consists of “natural” clusters, that indeed differ in their contributions to the data scatter. The second and third conditions can be considered as imposing certain degrees of resolution, reflected in the contribution thresholds, with which the user looks at the data.

At step 4, K -Means can be applied to either the entire dataset or to the set from which the smaller clusters have been removed. This may depend on the application domain: in some problems, such as structuring of a set of settlements for better planning or monitoring, no entity should be left out of the consideration, whereas in other problems, such as developing synoptic descriptions for text corpora, some deviant texts should be left out of the coverage.

An extensive set of experiments to test how well iK-Means recovers the “true” number of clusters have been described in [28, 29]. These experiments involved the data generated according to a mixture of Gaussians distributions,

several other methods for finding the “right” number of clusters, and a number of evaluation criteria, that are briefly described below.

1. Data generation.

The Gaussian mixture distribution data are generated as random samples using the functions in Neural Network NetLab, which are applied as implemented in a MATLAB Toolbox freely available on the web [13]. Our sampling scheme is based on a modified version of that proposed in Wasito and Mirkin (2006) utilizing two geometrically meaningful parameters: the clusters spread (the average distance between cluster centroids) and spatial cluster size which is proportional to the norm of the covariance matrix. We use either of two types of covariance structure: the ordinary spherical shape or the probabilistic principal component analysis (PPCA) shape [49]. The spatial cluster sizes are taken constant at the spherical shape, and variant at the PPCA shape. We maintain two types of the spatial cluster size by scaling covariance matrices using factors that are either proportional to the clusters index k (the linear distribution of sizes) or its square k^2 (the quadratic distribution of sizes) ($k = 1, 2, \dots, K$). Cluster centroids are generated randomly from a normal $N(0, 1)$ distribution with the follow-up scaling them by a factor expressing spread of the clusters. In the experiments, we generate data sets of size 1000×15 with the hidden dimension 6 involving two K settings (7 and 9 clusters), two spreads (small and large) and three types of distributions of cluster “diameters” (equal, k -proportional and k^2 proportional).

2. Methods involved.

We distinguish between four different approaches to the problem of determining the “right” number of clusters of which one is our ITEX approach and the other three involve statistics calculated at random trials of K -Means at different K in a prespecified range, typically from $K = 2$ to $K = 12 - 20$. Given K value, a random initialization of K centroids is generated (within the features’ ranges) and K -Means applies until convergence. After a number of runs, set to 100 in our experiments, for a specified statistic s , the statistic’s value s_K is calculated. Then the best K , according to the statistic, is selected. The three approaches we found in the literature involve statistics based on variance, structure and consensus as follows.

a) Variance based statistics.

These are formulated in terms of the clustering criterion

$$W = \sum_{k=1}^K \sum_{i \in S_k} d(i, c_k),$$

where S_k is the k -th cluster, c_k its centroid and $d(i, c_k)$ the Euclidean distance squared between i -th row of the data matrix and c_k . Obviously, W is the summary weighted within-cluster variance of the features.

Probably the earliest was Hartigan's criterion $H = (W_K/W_{K+1} - 1)(N - K - 1)$, where W_K is the minimum value of W at the results of a number of runs of the algorithm at given K and random initial settings. The criterion is based on the following intuition. Let there be a natural clustering of K_0 clusters in data. Then, at $K < K_0$, K -Means would tend to produce a clustering that aggregates some of the 'natural' clusters so that increasing K by 1 would just separate a cluster or cluster aggregate from the K -cluster clustering so that the relative change $(W_K - W_{K+1})/W_{K+1}$ would be relatively small. If however $K > K_0$, then the clusterings will tend to be the K 'natural' clusters randomly subdivided into smaller chunks. Thus, the very first K at which the relative difference becomes large enough, making $H \geq 10$ [16], should be the right number K_0 . Somewhat similar reasoning lies behind the Calinski and Harabasz's Fisher-like criterion [6]. We also utilize the so-called Jump statistic based on the maximum jump in value of $M_K^{V/2}$ where M_K is K -Means criterion computed according to Mahalanobis distances and divided by V ; this is proven to be the case if the data are generated according to a mixture of Gaussian distributions (see [48]).

b) *Structure based statistic.*

We utilize the popular average silhouette width introduced in [24] to reflect, for every entity, the difference between its within cluster distances and distances to the closest of other clusters.

c) *Concensus based statistics.*

In contrast to the other approaches, this one utilizes results of all, say M , runs of K -Means from random initialisations at a given K , not just the best of them. Monti et al. [40] proposed using the distribution of entries in the so-called consensus matrix that can be defined after a set of M runs of K -Means. This is an entity-to-entity similarity matrix, whose (i, j) -th entry is the number of those of the M clusterings in which i and j belong to the same cluster. In the ideal case, all clusterings coincide, which would make the distribution of the consensus matrix entries to be binomial. We use the jumps of two related indexes, one measuring the area under the cumulative distribution function proposed in [40] and the other, the average distance between clusterings that can be expressed in terms of the distribution's variance [34].

The ITEX clustering, in our experiments, was represented by two algorithms, one based on the least-squares criterion described above, the other based on the least modules criterion, thus differing in that the median is taken instead of the average, and the citi-block distance instead of the Euclidean squared distance. The cluster discarding threshold is taken to be 1.

3. Evaluation criteria.

The number of clusters generated is fixed at $K = 7$ or 9 , which is easy to compare with the number of clusters found at any method utilized. We also evaluate two aspects of a clustering, the intensional and extensional ones, by measuring the differences between cluster centroids and cluster contents.

To compare centroids of found clusters with those generated, in the situation when the numbers of clusters may differ, we utilize the following strategy. Let the generated clustering have K clusters and that found one K' clusters so that $K < K'$. We first one-to-one assign each of the K clusters with the closest one from the K' clusters, using the between-centroid distance as the criterion; after that we assign each of the remaining $K' - K$ clusters to that of the K clusters that are closest in terms of the between-centroid distances. Then we calculate the average distance between linked centroids either weighted by the cluster cardinalities or not. The weighted distance, in our experiments, appears to be orthogonal to other evaluation measures and thus, dropped off as an unworthy one.

To compare cluster contents between two partitions, we utilize four different measures of (dis)similarity between partitions: the distance [34], the adjusted Rand index [18], the Tchouproff coefficient and the averaged relative cluster-to-cluster overlap [34]. The four are highly correlated and they all support the general findings in [28, 29].

According to the experiments, the Hartigan statistics based method shows the best performance in terms of the number of clusters, though not in terms of centroids and cluster contents. In terms of the similarities between generated and found centroids and partitions, in most cases, the ITEX based methods performed better than the rest, and the least-squares ITEX somewhat better than that least-modules ITEX. Further analysis suggests that, most likely, ITEX results are inferior to those by other methods (typically, these are the silhouette width or jump statistics based methods that can be superior sometimes) in the cases in which ITEX based methods produce too many clusters.

This has led us to the following adjustment of the ITEX clustering methods. First, produce the H based evaluation of the number of clusters K_H . Second, if iK-Means leads to much more clusters than K_H , increase the cluster discarding threshold until the number of iK-Means clusters becomes reasonably close to K_H . In further experiments, with this adjustment, iK-Means clustering results on average were superior to all other methods in consideration [29].

6.3 ITEX Structuring and Clustering for Similarity Data

6.3.1 Similarity Clustering: a Review

The following review of the subject is reminiscent to that in [38].

Let $A = (a_{ij})$ be a symmetric matrix of similarities (or, synonymously, proximities or interactions) between entities $i, j \in I$. The greater the value of

a_{ij} , the greater is the similarity between i and j . A cluster is a set of highly similar entities whose similarity to entities outside of the cluster is low.

Similarity clustering emerged quite early in graph theory, probably before the discipline of clustering itself. A graph may be thought of as a structural expression of similarity data, its nodes corresponding to entities with edges joining similar nodes. Cluster related graph-theoretic concepts include: (a) *connected component* (a maximal subset of nodes in which every pair of nodes is connected by a path), (b) *bicomponent* (a maximal subset of nodes in which each pair of nodes belongs to a cycle), and (c) *clique* (a subset of nodes in which each pair of nodes is connected by an edge).

Other early clustering concepts include the B-coefficient method for clustering variables using their correlation matrix [17] and the Wroclaw taxonomy [9]. These are precursors to the ADDI and ADDI-S methods [31], described later, and the single linkage method [15, 14], respectively.

Two more recent graph-theoretic concepts are also relevant: *maximum density subgraph* [11] and *min-multi-cut* in a weighted graph [12].

The density $g(S)$ of a subgraph $S \subset I$ is the ratio of the number of edges in S to the cardinality of S . For an edge weighted graph with weights specified by the matrix $A = (a_{ij})$, the density $g(S)$ is equal to the *Raleigh quotient* $s^T A s / s^T s$, where $s = (s_i)$ is the characteristic vector of S , viz. $s_i = 1$ if $i \in S$ and $s_i = 0$ otherwise. A subgraph of maximum density represents a cluster. After removing such a cluster from the graph, a maximum density subgraph of the remaining graph can be found. This may be repeated until no “significant” clusters remain. Such an incomplete clustering procedure is natural for many types of data, including protein interaction networks. However, to our knowledge, this method has never been applied to such problems, probably because it involves rather extensive computations. A heuristic analogue can be found in [2]. We consider that the maximum density subgraph problem is of interest because it is a reasonable relaxation of the maximum clique problem and fits well into data recovery clustering (see section 6.3.3). The maximum value of the Raleigh quotient of a symmetric matrix over any real vector s is equal to the maximum eigenvalue and is attained at an eigenvector corresponding to this eigenvalue. This gives rise to *spectral clustering*, a method of clustering based on first finding a maximum eigenvector s^* and then defining the spectral cluster by $s_i = 1$ if $s_i^* > t$ and $s_i = 0$ otherwise, for some threshold t . This method may have computational advantages when A is sparse. Unfortunately, it does not necessarily produce an optimal cluster [33], but empirically it produces good clusters in most cases.

The concept of min-multi-cut is an extension of the max-flow min-cut concept in capacitated networks, and essentially seeks a partition of nodes into classes having minimum summary similarities between classes or, equivalently, maximum summary similarities within classes. When similarities are non-negative, this criterion may often lead to a highly unbalanced partition with one huge class and a number of singleton classes. This line of research has led to using the *normalized cut*, proposed in [45], as a meaningful clustering

criterion. The normalized cut criterion assumes that the set I should be split into two parts, S and \bar{S} , so that the normalized cut

$$nc(S) = a(S, \bar{S})/a(S, I) + a(S, \bar{S})/a(\bar{S}, I)$$

is minimized. Here $a(S, T)$ denotes the summary similarity between subsets S and T . The criterion $nc(S)$ can be expressed as a Raleigh quotient for a generalized eigenvalue problem [45], so the spectral clustering approach may be applied to minimizing the normalized cut too.

It should be noted that the user typically finds it meaningful, in the framework of domain knowledge, to define a similarity threshold α , such that entities i and j should be aggregated if $a_{ij} > \alpha$ but not if $a_{ij} < \alpha$. When this is the case, the data should be pre-processed to take the threshold into account. There are two different ways of implementing this idea: (i) by zeroing all similarities a_{ij} that are less than α , or (ii) by shifting the zero similarity to α by subtracting α from each similarity a_{ij} . The former is popular, for example, in image analysis because it makes the similarity data sharper and sparser. However, we favour the latter as better fitting in with the additive structure recovery models presented later. In fact, the similarity shift originated from these models (see, for example, [30, 31]).

6.3.2 The Additive Structuring Model and ITEX

To represent a set of structures assumed to underly the similarity matrix A , we use the terminology of binary relations. A binary relation on the set I can be defined by a (0,1) matrix $R = (r_{ij})$ such that $r_{ij} = 1$ if i and j are related and $r_{ij} = 0$ otherwise. Partitions, rankings and subsets can be represented by equivalence, order and square relations, respectively. A quantitative expression of the intensity of a relation can be modelled by a real value λ . So a relation of intensity λ is represented by the product λR .

Given a set of binary relations \mathcal{R} defined by a general property (for example, equivalence or order relations), an additive structuring model for a given $N \times N$ matrix $A = (a_{ij})$ is defined by the equations

$$a_{ij} = \sum_{k=0}^K \lambda_k r_{ij}^k + e_{ij}, \quad \text{for all } i, j \in I, \quad (6.6)$$

where $R^k = (r_{ij}^k) \in \mathcal{R}$ and λ_k is the intensity of R^k ; the number of relations $K+1$ in (6.6) is typically assumed to be much smaller than $|I|$, the cardinality of I . The goal is to minimize the residuals e_{ij} with respect to the unknown relations R^k and intensities λ_k . In some problems, the intensities λ_k may be given, based on substantive or model considerations.

To minimize the residuals in (6.6), the least-squares criterion can be applied again. This criterion brings in an important property. The data matrix A is not necessarily symmetric. However, in the situations in which all relations

in \mathcal{R} are symmetric the matrix A can be equivalently substituted by symmetric matrix $\tilde{A} = (A + A^T)/2$ where A^T denotes A transposed. Indeed, for any given set $R^k \in \mathcal{R}$ ($k = 1, \dots, K$) let us take any pair $i, j \in I$ and denote the sum in (6.6) by λ . Then the contribution of the pair i, j to the sum of squared residuals, $\sum e_{ij}^2$, will be equal to $(a_{ij} - \lambda)^2 + (a_{ji} - \lambda)^2 = f - 2\lambda(a_{ij} + a_{ji}) + \lambda^2$ where $f = a_{ij}^2 + a_{ji}^2$. If we change a_{ij} in (6.6) for $\tilde{a}_{ij} = (a_{ij} + a_{ji})/2$, then the contribution of i, j to the summary quadratic criterion will be $2(\tilde{a}_{ij} - \lambda)^2 = \tilde{f} - 4\lambda\tilde{a}_{ij} + \lambda^2$ where $\tilde{f} = 2\tilde{a}_{ij}^2 \leq f$. Since $\tilde{a}_{ij} = (a_{ij} + a_{ji})/2$, both expressions have the same variable parts, which proves that any least-squares solution to (6.6) remains a least-squares solution after a_{ij} is changed for \tilde{a}_{ij} at all $i, j \in I$.

In certain cases, we may require one of the relations R^k to be the universal relation, for which $r_{ij}^k = 1$ for all $i, j \in I$. The corresponding intensity λ_k then plays a role in the model (6.6) similar to that of the intercept in linear regression. Conventionally, we relabel the universal relation as R^0 and denote its matrix by $\mathbf{1}$. The intercept value λ_0 may be interpreted as a similarity shift, with the shifted similarity matrix $A' = (a'_{ij})$ defined by $a'_{ij} = a_{ij} - \lambda_0$. Equation (6.6) for the shifted model has a'_{ij} on the left and the sum on the right starting from $k = 1$.

With the least-squares criterion, we can employ again the greedy heuristic of extracting the relations R^k one by one in order to reduce the amount of computation and have a useful decomposition of the data scatter over found relations. This may be particularly useful if the relations R^k contribute very unequally to the data, for example, when the λ_k vary significantly. At step k , for $k = 0, 1, 2, \dots, K$, we find R^k using an algorithm for minimizing

$$L^2(R) = \sum_{i,j \in I} (a_{ij}^k - \lambda r_{ij}^k)^2 \quad (6.7)$$

over $R \in \mathcal{R}$ and λ (unless pre-specified). The residual similarity matrix $A^k = (a_{ij}^k)$ is updated after step k by subtracting $\lambda_k R^k$ from it. At the start, $A^0 = A$ and, at the end, $A^{K+1} = (e_{ij})$, the matrix of residuals.

Given R , the optimal value of λ is equal to the average similarity a_{ij}^k over all related pairs (i, j) , i.e. those for which $r_{ij} = 1$. The complexity of this minimization problem depends on the type of relations in \mathcal{R} . Therefore, in some cases, we only find a local minimum of (6.7).

When the λ_k are not pre-specified, then, at each step, the residual similarity matrix is orthogonal to the relation extracted. This implies the following Pythagorean decomposition [32, 33]:

$$\sum_{i,j \in I} (a_{ij})^2 = \sum_{k=0}^K \lambda_k^2 \sum_{i,j \in I} r_{ij}^k + \sum_{i,j \in I} e_{ij}^2. \quad (6.8)$$

This equation additively decomposes the data scatter into the contributions of the extracted relations R^k (“explained” by the model) and the minimised residual square error (the “unexplained” part). The decomposition (6.8)

makes it possible to prove that the residual part converges to zero under relatively mild and easily checked assumptions on the solutions found at each iteration [32, 33].

Obviously, our convention implies that, when the ITEX is to be applied to the shifted model, the universal relation R^0 must be extracted first. In this case, the optimal value of λ_0 will be equal to \bar{a} , the average of the similarities in A .

Also, the property that matrix A can be equivalently substituted by its symmetrized version $\tilde{A} = (A + A^T)/2$ if \mathcal{R} consists of symmetric relations, holds when the ITEX is utilized.

6.3.3 Additive Clustering Model

A square relation $r = (r_{ij})$ is defined by a subset $S \subseteq I$ in such a way that $r_{ij} = 1$ if both i and j belong to S and $r_{ij} = 0$, otherwise. In other words, $r_{ij} = s_i s_j$ where $s = (s_i)$ is the membership vector so that for any $i \in I$, s_i is 1 or 0 depending on whether $i \in S$ or not.

By restricting \mathcal{R} to consist of all square relations, the shifted version of the model (6.6) becomes what is referred to as the additive clustering model [44]. The universal relation $R^0 = \mathbf{1}$, used in the shifted model, is the square relation corresponding to the universal cluster I .

When we assume that the similarities in A are generated by a set of “additive clusters” $S^k \subseteq I$, $k = 0, 1, \dots, K$, in such a way that each a_{ij} approximates the sum of the intensities of those clusters that contain both i and j , the shifted version of (6.6) becomes:

$$a_{ij} = \sum_{k=1}^K \lambda_k s_i^k s_j^k + \lambda_0 + e_{ij}, \quad (6.9)$$

where $s^k = (s_i^k)$ are the membership vectors of the unknown clusters S^k , $k = 1, 2, \dots, K$, and e_{ij} are the residuals to be minimised. In this model, introduced in [44], the intensities λ_k , $k = 1, 2, \dots, K$, and the shift λ_0 also have to be optimally determined. In the more general formulation of the “categorical factor analysis” [30, 31], these values may be user specified.

We note that the role of the intercept λ_0 in (6.9) is three-fold: it can be considered as

1. an intercept of the bilinear model, similar to that in the linear regression or
2. the intensity of the universal cluster I or
3. a ‘soft’ similarity threshold in the sense that it is the shifted similarity matrix a'_{ij} , rather than the original A , is used to determine the clusters S^k , $k = 1, 2, \dots, K$. This role is of a special interest when λ_0 is user specified.

When the one-by-one ITEX strategy is applied to fitting (6.9) with none of the λ s pre-specified, the data scatter decomposition (6.8) holds for the

optimal values of λ_k . In this case, λ_k is equal to \bar{a}_k , the average of the residual similarities a_{ij}^k for $i, j \in S^k$. Substituting $s_i^k s_j^k$ for r_{ij}^k and \bar{a}_k for λ_k , (6.8) can be written in the form:

$$(A, A) = \sum_{k=0}^K [s^{kT} A^k s^k / s^{kT} s^k]^2 + (E, E). \quad (6.10)$$

The inner products (A, A) and (E, E) denote the sums of the squares of the elements of the matrices, considering A and E as vectors; these are conventionally expressed as the traces (sums of diagonal elements) of the products $A^T A$ and $E^T E$, respectively.

6.3.4 Approximate Partitioning

In this section, we restrict the additive clustering model to nonoverlapping clusters.

If clusters S^k , $k = 1, \dots, K$, are mutually disjoint (so the membership vectors s^k are mutually orthogonal), the optimal intensity λ_k depends only on the elements a'_{ij} , $i, j \in S^k$, of the shifted matrix $A' = A - \lambda_0 \mathbf{1}$ and not on the residual matrix A^k . The following decomposition of A' corresponding to (6.10) then holds and is independent of the the order of the clusters.

$$(A', A') = \sum_{k=1}^K [s^{kT} A' s^k / s^{kT} s^k]^2 + (E, E). \quad (6.11)$$

Although similar in form to the decomposition for A in (6.10), this decomposition for A' differs in that: (i) the terms in the summation involve the original matrix A' , not the residual matrix, and (ii) the summation starts from 1, not 0.

Since $A' = A - \lambda_0 \mathbf{1}$, it follows that

$$(A, A) = 2\lambda_0(\bar{a} - \lambda_0/2)(\mathbf{1}, \mathbf{1}) + \sum_{k=1}^K [s^{kT} A' s^k / s^{kT} s^k]^2 + (E, E). \quad (6.12)$$

When λ_0 is not pre-specified and must be found according to the least-squares criterion, its optimal value, found by differentiating (6.12) with respect to λ_0 , is:

$$\lambda_0 = \frac{\sum_{i,j \in I} a_{ij}(1 - s_{ij})}{\sum_{i,j \in I} (1 - s_{ij})}, \quad (6.13)$$

where $s_{ij} = \sum_{k=1}^K s_i^k s_j^k$ (so $s_{ij} = 1$ if both i and j belong to S^k for some $k = 1, 2, \dots, K$ and $s_{ij} = 0$ otherwise).

Thus, the optimal λ_0 is the average of similarities a_{ij} for i and j belonging to different clusters.

Equation (6.11) is analogous to the representation of the trace of $A^T A$ as the sum of the squares of the eigenvalues of A' because the terms are squares of the Raleigh quotients

$$g(s^k) = s^{kT} A' s^k / s^{kT} s^k . \quad (6.14)$$

which are attained at zero/one rather than arbitrary vectors s^k .

According to (6.11), an optimal partition with weights λ_k adjusted according to the least-squares criterion must maximize the sum of the cluster contributions $g^2(s^k)$, that is,

$$\sum_{k=1}^K g^2(s^k) = \sum_{k=1}^K \left(\sum_{i,j \in S^k} a'_{ij} / N_k \right)^2 , \quad (6.15)$$

where $N_k = |S^k|$, the cardinality of S^k .

An “unsquared” version of this criterion comes from applying the data recovery approach to an entity-to-feature data matrix in section 1, which leads to

$$\sum_{k=1}^K g(S^k) = \sum_{k=1}^K \sum_{i,j \in S^k} a_{ij} / N_k , \quad (6.16)$$

as the contribution of the clusters to the entity-to-feature data scatter. The similarity a_{ij} is defined, in this approach, as the inner product of the feature vectors corresponding to entities i and j . In matrix terms, if Y is an entity-to-feature data matrix then A is defined as $A = YY^T$. The difference between criteria (6.15) and (6.16) is somewhat similar to that between the spectral decomposition of $A = YY^T$ and singular-value decomposition of Y .

In contrast to (6.16), criterion (6.15) has never been analysed, neither theoretically nor experimentally.

To illustrate the difference between preset and optimal values of the shift λ_0 when model (6.9) is used for approximate partitioning, let us consider the similarity data between eight entities in Table 6.2.

For $\lambda_0 = 2$, the only positive values of $a'_{ij} = a_{ij} - \lambda_0$ are within clusters 1-2-3, 4-5, and 6-7-8 plus similarities between entity 4 and both 6 and 7. These positive extra-cluster similarities lead to differences in the clustering if λ_0 is changed. At the average similarity shift $\lambda_0 = \bar{a} = 1.49$, these three clusters with respective intensities 3.46, 3.13 and 3.70 form the optimal partition. This partition contributes 37.1% to the original data scatter. For the globally optimal partition, the $\lambda_0 = 0.49$ and entity 4 joins the cluster 6-7-8. The optimal partition then consists of clusters 1-2-3 (with intensity 4.47), 4-6-7-8 (with intensity 3.17), and singleton 5 (since self-similarity is not defined, the intensity has no meaning). This contributes 65.6% of the data scatter. The rather large difference between the two contributions to the data scatter is mainly due to the difference in the first term on the right-hand side of (6.12) involving λ_0 .

Table 6.2. Illustrative similarities between eight entities; self-similarity is not defined

Entity	1	2	3	4	5	6	7	8
1	-	4.33	5.60	-0.20	-0.16	-0.21	-0.49	0.17
2	4.33	-	4.93	0.79	0.06	1.22	-0.10	-0.45
3	5.60	4.93	-	0.21	0.79	-1.20	-0.15	0.80
4	-0.20	0.79	0.21	-	4.62	3.29	2.80	0.32
5	-0.16	0.06	0.79	4.62	-	-1.00	0.25	-0.08
6	-0.21	1.22	-1.20	3.29	-1.00	-	5.96	4.38
7	-0.49	-0.10	-0.15	2.80	0.25	5.96	-	5.23
8	0.17	-0.45	0.80	0.32	-0.08	4.38	5.23	-

6.3.5 One Cluster Clustering

Applying ITEX to the additive clustering involves extracting a single cluster from, possibly residual, similarity data presented in the form of a symmetric matrix A , assuming that any required shift λ_0 has already been made. As noted above, if A is not symmetric, it can be equivalently changed for symmetric $\tilde{A} = (A + A^T)/2$. For the sake of simplicity, in this section, we assume that the diagonal entries a_{ii} are all zero.

Pre-specified intensity

We first consider the case in which the intensity λ of the cluster to be found is pre-specified. Noting that $s_i^2 = s_i$ for any 0/1 variable s_i , criterion (6.7) can be expressed as

$$L^2(S) = \sum_{i,j \in I} (a_{ij} - \lambda s_i s_j)^2 = \sum_{i,j \in I} a_{ij}^2 - 2\lambda \sum_{i,j \in I} (a_{ij} - \lambda/2) s_i s_j. \quad (6.17)$$

Since $\sum_{i,j} a_{ij}^2$ is constant, for $\lambda > 0$, minimizing (6.17) is equivalent to maximizing the summary within-cluster similarity after subtracting the threshold value $\pi = \lambda/2$:

$$f(S, \pi) = \sum_{i,j \in I} (a_{ij} - \pi) s_i s_j = \sum_{i,j \in S} (a_{ij} - \pi). \quad (6.18)$$

This criterion implies that, for an entity i to be added to or removed from the S under consideration, the difference between the value of (6.18) for the resulting set and its value for S , $f(S \pm i, \pi) - f(S, \pi)$, is equal to $\pm 2f(i, S, \pi)$ where

$$f(i, S, \pi) = \sum_{j \in S} (a_{ij} - \pi) = \sum_{j \in S} a_{ij} - \pi |S|.$$

This gives rise to a local search algorithm for maximizing (6.18): start with $S = \{i^*, j^*\}$ such that $a_{i^* j^*}$ is maximum element in A , provided that $a_{i^* j^*} > \pi$.

An element $i \notin S$ may be added to S if $f(i, S, \pi) > 0$; similarly, an element $i \in S$ may be removed from S if $f(i, S, \pi) < 0$. The greedy procedure ADDI [31] iteratively finds an $i \notin S$ maximising $+f(i, S, \pi)$ and an $i \in S$ maximizing $-f(i, S, \pi)$, and takes the i giving the larger value. The iterations stop when this larger value is negative. The resulting S is returned along with its contribution to the data scatter, $4\pi \sum_{i \in S} f(i, S, \pi)$. The following version of ADDI reducing the dependence on the initial S proved successful in experiments. The computations here start from the singleton $S = \{i\}$, for each $i \in I$, so that N ADDI based results are generated; of these, that cluster S is selected that contributes most to the data scatter, i.e., that minimizes the square error $L^2(S)$ (6.17). In fact, the set of resulting clusters should be of interest on its own since many of them coincide or almost coincide and the structure of not coinciding clusters represents an overlapping structure of the similarity data.

The heuristic algorithm CAST [3], popular in bioinformatics, is in fact a version of the ADDI algorithm, because it uses the same iterative process of adding or removing an entity by utilizing criterion $\sum_{j \in S} a_{ij} > \pi|S|$, for the case of adding, with the $\sum_{j \in S} a_{ij}$ referred to as the affinity of i to S – which is equivalent to criterion $f(i, S, \pi) > 0$.

Another property of the criterion is that $f(i, S, \pi) > 0$ if and only if the average similarity between a given $i \in I$ and the elements of S is greater than π , which means that the final cluster S produced by ADDI/CAST is rather tight: the average similarities between $i \in I$ and S is at least π if $i \in S$ and no greater than π if $i \notin S$ [31].

Intuitively, changing the threshold π should lead to corresponding changes in the optimal S . Indeed, it has been proven that the greater π is, the smaller S will be [31].

Optimal intensity

When λ in (6.17) is not fixed but chosen to further minimize the criterion, it is not difficult to prove that

$$L^2(S) = (A, A) - [s^T A s / s^T s]^2, \quad (6.19)$$

in line with the decomposition (6.11), with $K = 1$ and $L^2(S) = (E, E)$. The proof is based on the fact that the optimal λ is the average similarity $a(S)$ within S , i.e.,

$$\lambda = a(S) = s^T A s / [s^T s]^2, \quad (6.20)$$

since $s^T s = |S|$.

The decomposition (6.19) implies that the optimal cluster S must maximize the criterion

$$g^2(S) = [s^T A s / s^T s]^2 = a^2(S) |S|^2. \quad (6.21)$$

According to (6.21), the maximum of $g^2(S)$ may correspond to either positive or negative value of $a(S)$. The latter case may emerge when the similarity

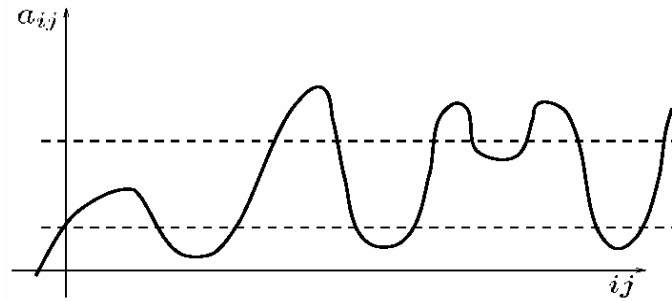


Fig. 6.2. A pattern of clustering depending on the subtracted similarity shift λ_0 is large and corresponds to S being the so-called *anti-cluster* [33]. In this paper, we do not consider this case, but focus on maximizing (6.21) for positive $a(S)$ only. This is equivalent to maximizing the Raleigh quotient,

$$g(S) = s^T A s / s^T s = a(S) |S|. \quad (6.22)$$

To maximize $g(S)$, one may utilize the ADDI-S algorithm [31], which is the same as the algorithm ADDI/CAST, described above, except that the threshold π is recalculated after each step as $\pi = a(S)/2$, corresponding to the optimal λ in (6.20).

A property of the resulting cluster S , similar to that for the constant threshold case, holds: the average similarity between i and S is at least half the within-cluster average similarity $a(S)/2$ if $i \in S$, and at most $a(S)/2$ if $i \notin S$.

To obtain a set of (not necessarily disjoint) clusters within the framework of the additive clustering model, one may use ITEX by repeatedly extracting a cluster S using ADDI-S and then replacing A by the residual matrix $A - a(S) s s^T$.

We can apply this method to the partitioning problem, by repeatedly using ADDI-S to find a cluster S and then removing from consideration all the entities in S . The process stops when the similarity matrix on the remaining entities has no positive entries. The result is a set of non-overlapping clusters S_k , $k = 1, \dots, K$, each assigned with its intensity $a(S_k)$ and contribution to the data scatter $g^2(S_k)$, and also the remaining unclustered entities in I .

ADDI-S utilizes no ad hoc parameters, so the number of clusters is determined by the process of clustering itself. However, changing the similarity shift λ_0 may affect the clustering results, which can be of advantage in contrasting within- and between- cluster similarities. Figure 6.2 demonstrates the effect of changing a positive similarity a_{ij} to $a'_{ij} = a_{ij} - \lambda_0$ for $\lambda_0 > 0$; small similarities $a_{ij} < \lambda_0$ are transformed into negative similarities a'_{ij} .

6.3.6 Some Applications

For the similarity data, the ITEX may lead to relevant overlapping clusters using similarity data. In our experience, the ITEX produced meaningful over-

lapping clusters in the situations in which the model of additive similarities was applicable. Here we briefly review three applications.

Elementary meanings in sorting experiments

A sorting experiment, in psycho-linguistics, goes as follows [43]. A number of nouns expressing concepts related to an aspect of the real world, such as the “kin” or “kitchenware” are put down on a separate card each, and respondents are requested to sort the cards into groups according to subjective similarity of the concepts. The extent of similarity between two concepts is expressed then by the number of respondents who put the concepts into the same group (“consensus” similarity). These similarities reflect the semantic similarities within the community represented by the respondents. Considering that an elementary meaning can be expressed as a cluster of certain intensity, it is reasonable to suggest that the similarity between two concepts should be equal to the sum of the intensities of those clusters that contain both of them. For example, the similarity between kinship concepts “son” and “father” should sum up intensities of elementary meanings such as “Nuclear family” and “Male relatives”. Therefore, the additive clustering model should be applicable here.

The iterative one-by-one extraction with ADDI-S algorithm has been applied for finding out additive clusters underlying a sorting consensus similarity matrix several times. In the analysis of similarities between 72 kitchenware terms, it was found that none of the clusters reflected logical or structural similarities between the kitchenware items; all the clusters related to the usage. Specifically, three types of communality were represented by the clusters: (i) a common process, such as frying or boiling; (ii) a common consumption use, such as drinking or eating, and (iii) a common situation such as a banquet [10].

Kim and Rosenberg data of sorting 15 kinship terms, observed six times [43], have been analyzed with the iterative ADDI-S adapted to the three-way data type in [33], p. 223. The clusters, in general, supported previously published findings such as clusters of “male relatives” or “female relatives”, but also added more subtle groupings such as “aunt, uncle”. The group “brother, daughter, father, mother, sister, son” that had been interpreted as “nuclear family” was further divided into “daughter, father, mother, son” and “brother, sister”, which some might view as more elementary groupings expressing the concepts of “nuclear family” and “siblings”, respectively.

Subject clusters in profiling a research organization

Profiling is a relatively new activity in computation, related to finding such features in data that are relevant to a pre-specified list of properties. The data may be rather unstructured, such as a text or set of texts. Profiling can be done rather conveniently with respect to a taxonomy or ontology in which all properties of interest are clearly delineated and well structured. The

paper [39] specifically refers to the ACM classification of Computer Sciences (ACMC) [1] that can be used for profiling a Computer Science department. The ACM classification is organized as a three layer tree. The first layer items are: A. General Literature, B. Hardware, C. Computer Systems Organization, D. Software, E. Data, F. Theory of Computation, G. Mathematics of Computing, H. Information Systems, I. Computing Methodologies, J. Computer Applications, and K. Computing Milieux. They are further subdivided into the second layer items such as I.5 Pattern Recognition. The third layer comprises further divisions such as I.5.3. Clustering.

The method proposed in [39] maps the set of subjects that are investigated in a research department onto the ACM classification and involves:

1. Selecting the level of classification to be used as the baseline.
2. Measuring similarities between selected ACMC topics according to the research activities of the department in question.
3. Decomposing the similarity structure by finding topic clusters that are not necessarily disjoint. Here the additive cluster model seems appropriate, thus ADDI-S applicable.
4. Mapping topic clusters to the ACM classification and highlighting the *head subjects*, *offshoots* and *gaps* revealed. A head subject of a topic cluster is the ACMC subject of a higher layer, whose “children” in the classification tree belong to the cluster, with gaps being those children that do not belong to the topic cluster.

As an example, we considered all 59 specific topic items of the second layer of the ACM classification, of which 26 have been covered by the research going on in the department under consideration [39]. The similarity between two topics was measured as just the number of academics pursuing both topics in their research. Application of ADDI-S to the similarity matrix leads to six clusters with contributions not less than $1/N = 1/26 = 4\%$. Five of the clusters mainly fall within the corresponding five head subjects, with very few gaps and offshoots to other ACMC nodes. One of the clusters, however, covers two of the head subjects which come on top of two other subject clusters, each pertaining to just one of the head subjects, D. Software or H. Information Systems. This can be interpreted as an indication that the two-headed cluster signifies a new direction in Computer Sciences, combining D and H into a single new direction, which seems to be a feature of the current developments in Computer Sciences unifying software and information systems indeed.

Aggregate protein families

This is an example in which a partition, not a set of potentially overlapping clusters, is sought, with the concept of similarity used to analyze complex objects, such as protein families. Proteins belonging to different organisms are combined into a family if they perform the same function and are considered as orthologous, that is, inherited from a common ancestor and being similar

because of that. Two features of general interest should be noted in the relation to this application: (i) using similarity between neighbourhoods rather than the objects themselves, and (ii) interplay with knowledge domain for getting a “right” similarity shift value [38].

The usage of neighbourhoods is convenient for complex objects. Similarity between complex objects is relatively straightforward to measure when they are similar indeed: the differences can be captured by superpositioning one object over another. When the similarity decreases, however, finding a correct superposition becomes rather tricky and subject to local search and arbitrary parameter values. This is why it is convenient to represent a complex object by its neighbourhood, which is the set of entities that are similar to the object (the idea first proposed in [47]). When the neighbourhoods are defined, two objects, $i, j \in I$, can be compared by comparing their neighbourhoods $L(i)$ and $L(j)$ with a convenient similarity measure between sets. The most popular between-set similarity measure is the Jaccard coefficient, sometimes referred to as Tanimoto’s coefficient, equal to $n/(n1 + n2 - n)$ where $n1$, $n2$ and n are cardinalities of the neighbourhoods and their intersection, respectively. The ratio relates the cardinalities of the overlap and the set theoretic union of the neighbourhoods. This coefficient, however, suffers from an intrinsic flaw of systematically underestimating the similarity [36].

The most natural indexes would be the relative sizes of the overlap $\frac{n}{n1}$ and $\frac{n}{n2}$, but they are not symmetric and are avoided by the researchers because of this. However, in the context of additive clustering, these can be used anyway because they can be equivalently converted to their average, $mbc = \frac{1}{2}(\frac{n}{n1} + \frac{n}{n2})$, as proven in section 3.2. The use of mbc index alleviates the issues of Jaccard-Tanimoto’s coefficient [36].

To define an appropriate similarity shift value, families with known function have been selected and, of those, family pairs have been put into two categories: (I) those whose function is clearly the same (86 pairs), and (II) those whose function clearly differ (279 pairs) [37, 38]. The pair-wise similarities should be high in the category (I) and low in the category (II), so that any intermediate value could be taken as the scale shift. It appears, the two distributions of similarities in this application are not disjoint; some proteins with the same function have very weak similarities. Therefore, two most likely shift values have been chosen: (i) that at which the distributions overlap (0.42) thus minimizing the rate of error in deciding which pairs should have positive and which negative similarity, and (ii) that at which none of the proteins have different functions (0.67). The final decision is made by comparing scenarios of evolution of the aggregate families with the knowledge of gene arrangement. It appears, both the shift values lead to similar clusterings, but the shift of 0.42 provides a cluster whose reconstructed history is more consistent with other knowledge than does the shift of 0.67.

Conclusion

Iterative extraction in clustering is a powerful approach from both theoretical and practical points of view. We tried to demonstrate that with two data formats, quantitative entity-to-feature data and similarity data, and two cluster structures, overlapping clusters and non-overlapping clusters.

The ITEX approach has been applied to other data formats, such as co-occurrence or mixed scale data, too; and a number of other cluster structures were utilized in different applications; the hierarchical cluster structures have been shown to be treatable with ITEX as well [33]. Applying the approach to other, yet not tried, data formats such as temporal and/or spatial data could be of interest.

The Pythagorean decomposition of the data scatter into explained and unexplained parts, pro-intuitive cluster properties, and fast computation are among advantages of the ITEX. Its shortcomings are related to: (i) the compulsory additive structure of the underlying model and (ii) unequal contributions of the underlying clusters. The former is probably behind the lack of intersection among ITEX clusters at the entity-to-feature data. Indeed, the idea of summing up centroids to represent intersections of corresponding clusters may be somewhat odd in some contexts.

The mentioned shortcomings suggest a number of directions for the theoretical and experimental investigation into the ITEX: (i) extending it to non-additive clustering models, (ii) characterization of data structures that are reliably treatable with the ITEX, (iii) development of different criteria for the ITEX, not necessarily based on minimizing residuals in the one-cluster models.

References

1. The ACM Computing Classification System (1998), <http://www.acm.org/class/1998/ccs98.html>.
2. Bader, G.D. and Hogue, C.W.V.: An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4:2 (2003)
3. Ben-Dor, A., Shamir, R., and Yakhini, Z.: Clustering gene expression patterns. *Journal of Computational Biology*, **6**, 281-297 (1999)
4. Benzecri, J.P.: *Correspondence Analysis Handbook*. Marcel Dekker, New York (1992)
5. Brohée, S. and van Helden, J.: Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, 7:488 (2006) (<http://www.biomedcentral.com/1471-2105/7/488>).
6. Calinski, T. and Harabasz, J.: A Dendrite Method for Cluster Analysis. *Communications in Statistics*, **3** (1), 1-27 (1974)
7. Depril, D., Van Mechelen, I., and Mirkin, B.: Algorithms for additive clustering of rectangular data tables submitted. (2007)

8. Everitt, B.S. and Dunn, G.: Applied Multivariate Data Analysis. Arnold, London (2001)
9. Florek, K., Lukaszewicz, J., Perkal, H., Steinhaus, H., and Zubrzycki, S.: Sur la liaison et la division des points d'un ensemble fini. *Colloquium Mathematicum*, **2**, 282–285 (1951)
10. Frumkina, R.M., Mikheev, A.V.: Meaning and Categorization. Nova Science Publishers, Commack, N.Y. (1996)
11. Gallo, G., Grigoriadis M.D., and Tarjan R.E.: A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, **18**, 30–55 (1989)
12. Garg, N., Vazirani, V.V., and Yannakakis, M.: Approximate Max-Flow Min-(Multi) Cut theorems and their applications. *SIAM Journal on Computing*, **25** (2), 235–251 (1996)
13. Generation of Gaussian mixture distributed data 2006, NETLAB neural network software, <http://www.ncrg.aston.ac.uk/netlab>.
14. Gower, J.C. and Ross, G.J.S.: Minimum spanning trees and single linkage cluster analysis. *Applied Statistics*, **18**, 54–64 (1969)
15. Hartigan, J.A.: Representation of similarity matrices by trees: J. Amer. Stat. Assoc., **62**, 1140–1158 (1967)
16. Hartigan, J.A.: *Clustering Algorithms*, J. Wiley & Sons, New York (1975)
17. Holzinger, K.J. and Harman, H.H.: *Factor Analysis*, University of Chicago Press, Chicago (1941)
18. Hubert, L.J. and Arabie, P.: Comparing Partitions. *Journal of Classification*, **2**, 193–218 (1985)
19. Inkpen, D., and Desilits, A.: Semantic similarity for detecting recognition errors in automatic speech transcripts. In: Conference on Empirical Methods in Natural Language Processing. Vancouver, Canada (2005)
20. Jain, A.K. and Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs, NJ (1988)
21. Jarvis, R.A. and Patrick E.A.: Clustering using a similarity measure based on shared nearest neighbors. *IEEE Trans. Comput.*, **22**, 1025–1034 (1973)
22. Jolliffe, I.T.: *Principal Component Analysis*. Springer, New York, (1986)
23. Kawaji, H., Takenaka, Y., and Matsuda, H.: Graph-based clustering for finding distant relationships in a large set of protein sequences, *Bioinformatics*, **20** (2), 243–252 (2004)
24. Kaufman, L. and Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley & Sons, New York (1990)
25. Krzanowski, W. and Lai, Y.: A criterion for determining the number of groups in a dataset using sum of squares clustering. *Biometrics*, **44**, 23–34 (1985)
26. McLachlan, G. and Basford, K.: *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York (1988)
27. McQueen, J.: Some methods for classification and analysis of multivariate observations. In: Fifth Berkeley Symposium on Mathematical Statistics and Probability, II, 281–297 (1967)
28. Chiang, M. M.-T. and Mirkin, B.: Determining the number of clusters in the straight K -Means: Experimental comparison of eight options. In: Proceedings of United Kingdom Computational Intelligence Workshop, Leeds, 143–150 (2006)
29. Chiang, M. M.-T. and Mirkin, B.: Initializing K -Means clustering: An experimental study, submitted (2007)
30. Mirkin, B.: Analysis of Categorical Features, Finansy i Statistika Publishers, Moscow (1976) (In Russian)

31. Mirkin, B.: Additive clustering and qualitative factor analysis methods for similarity matrices. *Journal of Classification*, **4**, 7–31 (1987); Erratum. **6**, 271–272 (1989)
32. Mirkin, B.: A sequential fitting procedure for linear data analysis models. *Journal of Classification*, **7**, 167–195 (1990)
33. Mirkin, B.: *Mathematical Classification and Clustering*. Kluwer Academic Press, Dordrecht (1996)
34. Mirkin, B.: *Clustering for Data Mining: A Data Recovery Approach*. Chapman and Hall, Boca Raton (2005)
35. Mirkin, B. and Satarov, G.: Method of fuzzy additive types for analysis of multidimensional data. *Autom. Remote Control*, I, II, **51** (5, 6), 683–688 (1990)
36. Mirkin, B. and Koonin, E.: A top-down method for building genome classification trees with linear binary hierarchies. In: Janowitz, M., Lapointe, J.-F., McMorris, F., Mirkin, B., and Roberts, F. (eds.) *Bioconsensus*, DIMACS Series, vol. 61, AMS, Providence: 97–112 (2003)
37. Mirkin, B., Camargo, R., Fenner, T., Loizou, G., and Kellam, P.: Aggregating homologous protein families in evolutionary reconstructions of herpesviruses. In: Ashlock, D. (ed.) *Proceedings of the 2006 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*. Piscataway NJ, 255–262 (2006)
38. Mirkin, B., Camargo, R., Fenner, T., Loizou, G., and Kellam, P.: Using domain knowledge and similarity scale shift in clustering, submitted (2007)
39. Mirkin, B., Nascimento, S., Moniz Pereira, L.: Using ACM classification for profiling a research organisation, submitted (2007)
40. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus Clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, **52**, 91–118 (2003)
41. Murtagh, F.: *Correspondence Analysis and Data Coding with JAVA and R*. Chapman & Hall/CRC, Boca Raton, FL (2005)
42. Nascimento, S., Mirkin, B., and Moura-Pires, F.: Modeling proportional membership in fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, **11** (2), 173–186 (2003)
43. Rosenberg, S.: The method of sorting in multivariate research with applications selected from cognitive psychology and person perception. In: Hirschberg, N. and Humphreys, L.G. (eds.) *Multivariate Applications in the Social Sciences*, University of Illinois at Urbana-Champaign: L. Erlbaum Assoc., 117–142. (1982)
44. Shepard, R.N. and Arabie, P.: Additive clustering: representation of similarities as combinations of overlapping properties. *Psychological Review*, **86**, 87–123 (1979)
45. Shi, J. and Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22** (8), 888–905 (2000)
46. Smid, M., Dorssers, L.C.J., and Jenster, G.: Venn Mapping: clustering of heterologous microarray data based on the number of co-occurring differentially expressed genes. *Bioinformatics*, **19** (16), 2065–2071 (2003)
47. Small, H.: Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, **24**, 265–269 (1973)
48. Sugar, C.A. and James, G.M.: Finding the number of clusters in a data set: An information-theoretic approach. *Journal of American Statistical Association*, **98** (463), 750–778 (2003)

49. Tipping, M.E. and Bishop, C.M.: Probabilistic principal component analysis. *J. Roy. Statist. Soc. Ser. B* **61**, 611–622 (1999)
50. Wasito, I. and Mirkin, B.: Nearest neighbours in least-squares data imputation algorithms with different missing patterns, *Computational Statistics & Data Analysis* **50**, 926–949 (2006)