Applying Matrix Decompositions to Counterterrorism

D.B. Skillicorn
School of Computing
Queen's University, Kingston, Canada
skill@cs.queensu.ca

May 2004
External Technical Report
ISSN-0836-0227-
2004-484

Department of Computing and Information Science
Queen's University
Kingston, Ontario, Canada K7L 3N6

Document prepared May 19, 2004

Abstract

Governments collect data in which they hope to find patterns of terrorist activity. It is hard to know what such patterns look like and, in any case, terrorists are actively trying to avoid leaving any distinctive traces. However, if they work as a group, it is impossible to avoid some correlation among their attributes and actions. We show that such correlation can be detected, partly because it is likely to be qualitatively different from the correlations among groups with more innocent purpose. We show that matrix decompositions, in particular singular value decomposition and semidiscrete decomposition, have several useful properties for this problem. In many cases it is possible to identify a terrorist group with few false positives.

Applying Matrix Decompositions to Counterterrorism

D.B. Skillicorn
skill@cs.queensu.ca

Abstract: Governments collect data in which they hope to find patterns of terrorist activity. It is hard to know what such patterns look like and, in any case, terrorists are actively trying to avoid leaving any distinctive traces. However, if they work as a group, it is impossible to avoid some correlation among their attributes and actions. We show that such correlation can be detected, partly because it is likely to be qualitatively different from the correlations among groups with more innocent purpose. We show that matrix decompositions, in particular singular value decomposition and semidiscrete decomposition, have several useful properties for this problem. In many cases it is possible to identify a terrorist group with few false positives.

1 Introduction

One important component of counterterrorism efforts is the analysis of data looking for traces of planned terrorist activity. The available data is extremely large, of multiple kinds, and collected via a number of different pathways, both overt and covert.

It is extremely unlikely that planned terrorist activity can be directly discovered in such data, for two reasons. First, there is not enough experience to judge what patterns in the data might correspond to terrorist action and, in any case, each new attack is presumably designed to differ in significant ways from previous attacks. Second, terrorists are aware of potential surveillance and hence make every effort to look innocuous with respect to *any* collectable data.

An effective data analysis strategy for counterterrorism must therefore aim to discover the precursors to a terrorist attack rather than an attack itself. These might include unusual patterns of money transfer, patterns of surveillance of a potential target, or patterns of collusion or collaboration. Precursors are likely to be both more common, so that some assessment of the effectiveness of the detection mechanism can be made; and more consistent, since different forms of attack may use, for example, similar patterns of funding. An effective data analysis strategy must also take full account of the efforts of terrorists to hide in the background noise of the ordinary actions of ordinary people.

Fortunately there is some evidence [2] that terrorist groups, like criminal groups, are structured differently from the other groups that make up a society. For example, such groups are typically more cohesive internally, both because of a common purpose and the need to keep that purpose undetected; and less connected to other groups. Hence, while it may not be possible to detect a single terrorist, it may be possible to detect a terrorist group because of differences in the correlative structure among its members.

At present it is not plausible to build a classification model that would perfectly separate terrorists and their actions from the innocent, partly because of our limited understanding of data-mining technology, and partly because of problems with the quality of data that might be collected. What is plausible is to build a ranking model, which provides a predicted measure of the risk of terrorism posed by each object or person. This ranked list can be used to reduce the data to a manageable size, either for more sophisticated downstream data mining or for human analysis. The use of a staged approach also has the advantage of providing a point for judicial or procedural oversight, after which the data may be perhaps be augmented with more confidential attributes [19], for example names or other identifiers.

We show that it is possible, using matrix decompositions, to detect small groups of objects with unusual correlative structure against a background typical of societal groups. Some lesser known properties of singular value decomposition (SVD) are used to produce various kinds of ranked lists, and also clusterings in which clusters corresponding roughly to terrorist groups can be seen. We then show how semidiscrete decomposition (SDD) can be used to classify objects in an unsupervised way, often detecting terrorist groups with low false positive rates.

The results described here can be used for early-stage analysis of large datasets, potentially reduced the number of objects requiring further consideration by more than 90%.

Section 2 describes the major approaches to detecting signatures or patterns of terrorist action in large datasets. Section 3 discusses the datasets we will work with. Section 4 presents the properties of singular value decomposition, and Section 5 the properties of semidiscrete decomposition. Section 6 describes the experiments we perform and shows their results. Finally we draw some conclusions.

2 Approaches to detection

Terrorists whose information will be captured in a dataset, and who are aware of the fact, will take steps to ensure that the values of their attributes will be, as far as possible, innocuous. They cannot be entirely successful at this because:

- They have a purpose, and this purpose requires certain actions (purchasing fertilizer, surveilling a target) that necessarily force certain attribute values. They can hope that each of these attribute values is, in itself, either common or innocuous, and so that such actions will not make them stand out. For example, many people buy fuel oil and many people buy fertilizer. It is only the conjunction, together perhaps with a city address, that makes each individual purchase suspicious.
- They are working together, and this forces certain other actions such as meeting or communicating which generate related attribute values.

Three broad strategies for analyzing datasets exist, each assuming the existence of a different signature or pattern in the data:

1. Analyze each object in the dataset independently. This approach assumes that objects related to terrorism have attribute values that are somehow anomalous. Many standard data-mining techniques, for example decision trees, support vector machines, and supervised neural networks can be used to build predictive models of this kind (although obtaining accurately labelled data is still problematic).

   The countermeasure to this kind of analysis is to make sure that each object's attribute values are (together and separately) in the 'normal' range. For example, if airline profiling flags passengers who travel in first class, who pay cash, and who buy one-way tickets, the obvious strategy is to travel economy, pay with a credit card, and buy a return ticket. In general, models of this kind can be defeated by probing [3].

   However, single-object analysis is not necessarily useless. First of all, it may sometimes be hard to determine what attributes are being collected, so it may be hard to avoid generating anomalous attribute values 'by accident'. Second, the existence of such analysis forces unusual behavior in the effort to seem 'normal', and this unusual behavior is therefore self-conscious. This itself may generate a signature that is visible to more sophisticated analysis. For example, the paper [18] shows how the existence of a watch list of words in messages may force unusual word usage that is readily detectable.

2. Find connections among the objects in the dataset based on values of attributes that they share. Such analysis usually creates a graph in which the nodes are objects, and there is a link between two nodes when their objects share an attribute value. Particular subgraphs may represent a particular form of terrorist precursor.

   Link analysis and social network analysis [6] have been used to analyze such graphs [4, 9], and are responsible for the ubiquitous 'connecting the dots' metaphor. Krebs showed [13], in retrospect, that there were many links among the September 11th hijackers, and also that the diameter of the graph of connections among them was substantially reduced by a single meeting of some members of the group. The paper [15] describes experiments using Inductive Logic Programming on relational datasets recording nuclear smuggling and contract killing. This work could presumably be generalized to counterterrorism.

   Link analysis approaches have two weaknesses. The first has to do with the pattern that is searched for in the data. This pattern must be identified beforehand, which is problematic if some form of novel attack is being mounted. Also, there is no straightforward way to 'approximately' match a pattern (for example, if an edge is missing in the dataset because the corresponding attribute value was not collected).

   The second weakness is that countermeasures to decouple the links are relatively easy. For example, many links capture the fact that two or more people were in the same place at the same time; techniques such as the use of drops (in the physical world) and wikis (in the online world) can decouple the temporal aspects of such links. Intermediaries help to decouple the spatial aspects (especially since changing identities in the electronic and online worlds is easy). For example, it is known that groups can be determined by examining patterns of email traffic [20], but this can be defeated by using readily-available transient email accounts.

3. Find connections between objects based on correlations among the attribute values that they share. Analysis of this kind is the subject of this paper.

   Correlations have three advantages in comparison to link analysis. First, many attempts to avoid similar attribute values lead to correlated values: instead of all travelling to a meeting on one day, members of a group may travel spread over several days, but the similarity of travel patterns is still visible as correlation. Second, it is hard for a group of individuals to assess how similar their correlation is to that of other groups, and so it is hard for them to engineer the way in which they appear to an analysis technique. Third, and perhaps most important of all, techniques that detect correlation do not have to be primed with what to look for; they will detect correlation wherever it appears.

3 Data Generation Models

Datasets of a number of different kinds have been collected for counterterrorism analysis. Each dataset describes a number of objects by providing values for some set of attributes belonging to the objects. Some examples of such datasets are:

- Datasets where the objects are individuals, and the attributes are facts about them (age, address, educational attainment, citizenship). Such datasets are dense: every individual should have values for every attribute. For example, the CAPSS II airline passenger profiling system will use information from both the Lexis/ Nexis and Axciom databases [17].

- Datasets where the objects are individuals, and the attributes are actions they have taken (for example, travel to particular places during given time periods). Such datasets are usually sparse: most individuals will have done only a small subset of the possible actions. For example, airline travel databases are of this form.

- Datasets where the objects are messages, and the attributes are senders/receivers or the content of each message. Such datasets might be either dense or sparse, depending on which message attributes are collected. For example, the Echelon system intercepts many forms of communication and examines the properties of a legally-defined subset, for example the presence of particular words from a watch list [5].

Since, for obvious reasons, real datasets containing terrorist actions are not available, the quality of detection models will be evaluated using artificial datasets. This immediately raises the question of what kinds of datasets are plausible and, of course, any choice is open to criticism.

Dense datasets can plausibly be modelled using Gaussian distributions, both because this naturally captures the properties of large populations, and also the likely structure of attribute values within a terrorist group. For example, the locations where the members of a group live reflect a balance between the practicality of living in close proximity to each other (and perhaps to a target); and the need not to form an obvious group.

Sparse datasets can plausibly be modelled using Poisson distributions with small mean for the non-zero entries. These generate datasets in which small values are common and large ones uncommon. For example, in any given time period, most people will not have visited a particular city; those who have are most likely to have visited it only once; and there will be a small minority who have visited it several times.

4 Singular Value Decomposition

4.1 Structure of the decomposition

Singular Value Decomposition (SVD) [7] is a well-known technique for reducing the dimensionality of data.

Suppose that a dataset is represented as a matrix A with n rows (corresponding to objects) and m columns (corresponding to their attributes). Then the matrix A can be expressed as

$$A = USV'$$

where U is an $n \times m$ orthogonal matrix, S is an $m \times m$ diagonal matrix whose r non-negative entries (where A has rank r) are in decreasing order, and V is an $m \times m$ orthogonal matrix. The superscript dash indicates matrix transpose. The diagonal entries of S are called the *singular values* of the matrix A.

SVD is as an axis transformation to new orthogonal axes (represented by V), with stretching in each dimension specified by the values on the diagonal of S. The rows of U give the coordinates of each original row in the coordinate system of the new axes.

The useful property of SVD is that this transformation is such that the maximal variation among objects is captured in the first dimension, as much of the remaining variation as possible in the second dimension, and so on. Hence, truncating the matrices so that U_k is $n \times k$, S_k is $k \times k$ and V_k is $m \times k$ gives a representation for the dataset in a lower-dimensional space.

A way to understand SVD is the following: suppose that points corresponding to both rows and columns are plotted in the same k-dimensional space. Then each point corresponding to a

row is at the weighted median of the positions of the points corresponding to the columns and, simultaneously, each point corresponding to a column is at the weighted median of the positions of the points corresponding to the rows. Hence SVD can be viewed as translating correlation or similarity into proximity.

SVD measures variation with respect to the origin, so it is usual to transform the matrix A so that the attributes have zero mean and unit variance.

While SVD is a workhorse of data manipulation, it has number of subtle properties that are not well-known. We will use five of them.

Fact 1: The correlation between two objects is proportional to the dot product between their positions regarded as vectors from the origin. Two objects that are highly correlated have a dot product (the cosine of the angle between the two vectors) that is large and positive. Two objects that are highly negatively correlated have a dot product that is large and negative. Two objects that are uncorrelated have dot product close to zero.

The usefulness of this property comes because there are two ways for a dot product to be close to zero. The obvious way is for the vectors concerned to be orthogonal. However, when m is less than n (as it typically is) there are many fewer directions in which vectors can point orthogonally than there are vectors. Hence if most vectors are uncorrelated, they must still have small dot products but cannot all be orthogonal. The only alternative is that their values must be small. Hence vectors that are largely uncorrelated must have small magnitudes, and the corresponding objects are placed close to the origin in the transformed space. Hence, in a transformed space from an SVD, the points corresponding to objects that are 'uninteresting' (they are correlated either with nothing or with everything) are found close to the origin, while points corresponding to interesting objects are located far from the origin (potentially in different direction indicating different clusters of such objects).

The dot products between rows of a US matrix, even when truncated, capture these correlation relationships well because the neglected terms in the dot products are small.

Fact 2: The singular value decomposition of a matrix is insensitive to the addition (or subtraction) of independent zero-mean random variables with bounded variance [1]. This property has been used to speed up the computation of SVD by sampling or by quantizing the values of the matrix. In counterterrorism, the effect we are looking for is so small and the results so important that neither of these is attractive. However, the fact does explain why SVD is good at detecting clusters within clusters – the outer cluster representing the majority of the data has zero mean (by normalization) and so, by the *fuzzy central limit theorem*, increasingly resembles a normal distribution as the number of ordinary objects (and the number of attributes) increases.

Fact 3: SVD is a numerical technique, and so the magnitudes of the attribute values matter. However, multiplying the attribute values of a row of A by a scalar larger than 1 has the effect of moving the corresponding point further from the origin. Because the positions of all of the other points depend, indirectly, on their correlations with the scaled point, via their mutual interactions with the attributes, points that are correlated with the scaled point are pulled towards it. When there is little structure in the low-dimensional representation of a dataset, this scaling technique can be used to find the objects that are (positively) correlated with a given object. In practice, this often makes it easier to see a cluster that would otherwise be hidden inside another in a visualization.

Fact 4: A typical data matrix with m columns will have rank m. There is a general construction, due to Wedderburn [8], that can be used to remove values of a particular form from the matrix, leaving a new matrix whose rank is one smaller. The particular values removed can represent already-known information, which is thereby discounted, or information that is of particular interest.

Suppose that y is an $n \times 1$ vector (so it looks like a column of A) and x is an $m \times 1$ vector (so its transpose looks like a row of A). Then if $\omega = y'Ax \neq 0$, the matrix $B = \frac{1}{\omega}Axy'A$ has the same shape as A, and the matrix $A - B$ has rank one less than the rank of A. The products Ax and $y'A$ are the cosine similarities of the rows and columns of A to the specified vectors, and therefore the product $(Ax)(y'A)$ can be thought of as a stencil of the locations and values in A that together are similar to those specified.

Fact 5: The decomposition depends on all the data used, both normal and anomalous. The precise geometry of the separation of clusters of SVD is hard to predict without performing the decomposition, and impossible without knowledge of the dataset. Hence, a terrorist group cannot reverse engineer the transformation to determine how they will appear, even knowing that SVD is being used. In particular, SVD is resistant to probing attacks since any attempt to probe cannot control for the innocent objects considered at the same time.

4.2 Complexity

The complexity of SVD is $\mathcal{O}(nm^2)$, where n is the number of objects and m the number of attributes. For dense data, such a complexity verges on impractical. However, for sparse data SVD can be computed with complexity $\mathcal{O}(rk)$ where r is the number of nonzero entries in A and k is the number of dimensions retained. For travel data, this amounts to complexity linear in the number of objects, that is linear in the number of people considered.

4.3 Ways to use SVD

These properties allow SVD to be used in a number of ways to analyze complex datasets:

1. Denoising and dimensionality reduction. Because the axes of the transformed space are arranged in decreasing order of importance (visible in the magnitudes of the singular values on the diagonal of S), the SVD can be truncated at some number of dimensions, k, while retaining the greatest possible information. In some settings, it is natural to interpret the dimensions from $k+1$ to m as containing 'noise' that appears in the data as a side-effect of the collection process. Removing this 'noise' may make it easier to analyze the data subsequently.

   There are also some advantages to reducing the dimensionality of the data, even if some of its information is lost. For example, when $k = 2$ or 3, the rows of U can be plotted and visualized. Even for larger values of k, the geometry of the transformed space can be easier to work with than the full m dimensions.

   However, in a counterterrorism setting, dimensionality reduction *per se* carries some risks because the 'noise' may carry the interesting and useful information.

2. Spectral clustering. There are several ways in which the results of SVD can be used to cluster the objects (or indeed the attributes). For example, objects can be placed in clusters based on their similarity to the (left) singular vectors. Those objects whose vectors have dot product greater than 1/2 with the first singular vector are placed in the first cluster, those with dot

product greater than 1/2 with the second singular vector in the second cluster, and so on. Alternatively, the values of the first column of U can be sorted into ascending order, and sharp increases in value considered as boundaries between clusters. There are many other possibilities (see, for example, [21] and [10] for a survey of some possibilities).

The existence of multiple, principled ways of clustering the same data into different clusters makes the use of these techniques problematic unless the form of the solution is already well-understood (as it is, for example, in image analysis). However, in a counterterrorism setting, the form of the solution is, and will continue to be, unknown and so it is not clear how to apply spectral clustering reliably.

3. Using distance from the origin as a surrogate for interestingness. Fact 1 above explains that points corresponding to objects must be mutually arranged so that the dot products of slightly correlated vectors are close to zero. Hence objects that are uninteresting (correlated with everything or with nothing) are close to the origin. The objects in the dataset can therefore be ranked in order of distance from the origin, and the objects at the bottom of the list will be most important. Some information is lost by doing this, because the directions in which interesting points lie contain some information, but this can still be a useful procedure.

   Alternatively, points corresponding only to those objects that are far from the origin (greater than the median distance, say) can be plotted. This preserves directional information, and allows the cluster structure of the remaining points to be visualized.

4. Using correlation with a target object. When there is a natural interpretation for one object in the dataset as the target object (for example, the terrorist cell might be expected to have correlated attribute values because of surveillance), then the dataset can be further reduced by omitting those objects that are negatively correlated with the target.

5. Using both distance from the origin and correlation. The preceding techniques can both be applied, so that points that are either too close to the origin or negatively correlated with the target are omitted. This reduces the number of objects to be considered still further.

6. Looking for differences in the local neighborhood of each object. The nature of the correlative structure around each object can be explored by repeated application of SVD. In the first round, an SVD is performed on the full dataset, and those objects negatively correlated with the object under consideration are removed. The SVD is then repeated on the remaining rows of the original matrix (those corresponding to objects positively correlated with the object being considered). Some of these objects will now be negatively correlated in the context of the reduced number of objects, and these can be removed, and the process repeated. The size of the remaining sets of objects provide information about how well the object being considered is connected to the other objects. Intuitively, an object with a small, tight set of related objects will produce a sequence of subsets of the objects of rapidly shrinking size; while one with more normal, widespread connections will produce a sequence that shrinks more slowly.

7. Weighting objects and/or attributes. Objects and attributes of particular interest can be given increased weights before the SVD is calculated. As mentioned in Fact 3, the effect is to move the weighted objects further from the origin (making them seem more interesting) but also moving other correlated objects further from the origin as well. The size of the movement reflects the importance of the correlation between these other objects and the

weighted objects. This can be used to validate apparent connections among objects – they should become stronger and more visible when some of them are weighted.

8. Using rank 1 reductions. The rank 1 reductions described in Fact 4 can be used in several ways. One is to find those object-attribute pairs that are similar to those of the target, a kind of similarity-based lookup. For example, suppose that the target is row 27 of the matrix and attributes 13 and 57 are particularly significant. The vector y can be chosen to be zeroes except at row 13, and the vector x chosen to the zeroes except at columns 13 and 57. The resulting B matrix has entries whose magnitude corresponds to their similarity to the given stencil. In particular, rows that are particularly well-correlated with the target with respect to these attributes will have entries of large magnitude. On the other hand, the matrix that results from subtracting this matrix from A discounts the effect of this stencil.

5 SemiDiscrete Decomposition

5.1 Structure of the decomposition

Semidiscrete decomposition (SDD) [11, 12, 16] is superficially similar to SVD but is, underneath, a bump-hunting technique [14]. It finds regions of rectilinearly aligned locations in a matrix that contain elements of similar magnitude (the bumps).

Once again, given a matrix A representing data, its SDD is

$$A = XDY$$

where X is $n \times k$, D is a $k \times k$ diagonal matrix, and Y is $k \times m$. The differences from SVD are (a) k can take any value, including $k > m$, (b) the entries on the diagonal of D are non-negative but need not be decreasing, and (c) the entries of X and Y are all -1, 0, or $+1$.

The easiest way to see what SDD is doing is to consider A_i the (outer product) matrix obtained by multiplying the ith column of X and the ith row of Y. Each such matrix has the same shape as A and contains rectilinear patterns of $+1$s (representing positive bumps) and -1s (representing negative bumps) against a background of 0s. Hence each A_i represents the stencil of a region of similar (positive and negative value) and the value of d_i represents its height. Note that A is the sum of the A_i weighted by the d_i.

It is natural to sort X and Y so that the corresponding entries of D are in decreasing order, so that the most significant bumps are selected first. The X matrix can then be naturally interpreted as a hierarchical ternary classification of the rows of A. The first column of X classifies the rows of A into three groups: those whose X entry is $+1$, those whose X entry is -1, and those whose X entry is 0. Those whose entries are $+1$ and -1 are similar but opposite, while those whose entries are 0 are not in the bump being selected at this level.

Here is a small example:

$$\begin{bmatrix} 1 & 1 & 4 & 4 \\ 8 & 8 & 1 & 1 \\ 8 & 8 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 8 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

There are no -1 values in this example. The product of the first column of X and the first row of

Y is

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

which is a stencil covering the region of the array where the elements have the value 8 (which is the value of d_1). The second outer product selects the regions where the elements have the value 2. The third and fourth outer products select regions where the elements have value 1. These two could not be selected as a single stencil because they cannot be rectilinearly aligned.

5.2 Complexity

The algorithm to compute SDD has a heuristic component, but its complexity is usually considered to be $\mathcal{O}(k^2(n+m) + n\log n + m\log m)$. For $k = m$ this is comparable to the complexity of SVD but, of course, for smaller values of k it is much less.

5.3 Ways to use SDD

Semidiscrete decomposition is useful to find regions (not necessarily contiguous) in a matrix that contain values of similar magnitude. It can be applied in three ways:

1. Directly to the data matrix A. This amounts to finding sub-blocks of the matrix of similar magnitude. This could be quite effective as a method of link analysis, but it is disappointing when the matrix entries are correlated but of different magnitude.

2. To the correlation matrix AA'. Applying SDD to the correlation matrix performs better than applying it to the data matrix directly (although when $n > m$ the correlation matrix is much larger and correspondingly more difficult to work with).

3. To a truncated version of the correlation matrix. SVD and SDD can be combined into a single technique called the JSS (Joint SVD SDD) methodology. The following steps are performed:

   - The SVD of A is computed, the component matrices U, S and V are truncated at some k (say $k = 15$), and the truncated matrices are multiplied to produce a matrix A_k. This matrix has the same shape as A.
   - The correlation matrix $C = A_k A_k'$ is computed. This matrix is $n \times n$ and its entries represent the 'higher-order' correlation among objects. Some correlation due to 'noise' has been removed and some indirect correlation is now explicitly visible in this matrix (e.g. entries that would have been 0 in the correlation matrix of A may now contain non-zero values).
   - Each entry of C is replaced by its signed square. Unlike SVD, SDD is not scale independent. The selection of a bump depends on both the average magnitude of the values it includes and also the number of array positions that it covers. Increasing the relative magnitude of the entries weights the selection towards bumps of high magnitude but low area, which is appropriate for this problem.
   - The SDD of the scaled C matrix is computed. This SDD finds regions of similar value in the matrix; since it is a correlation matrix, such regions correspond to correlated rows in the original matrix, A.

The results of SVD require inspection to determine the possible presence of a terrorist cluster; SVD transforms the data into a form where such anomalous clusters are more visible. On the other hand, SDD produces a hierarchical classification in which objects are allocated to clusters with a proximity structure (closeness in the classification tree). Hence the technique can sometimes identify a terrorist cluster itself, particularly if the target is known.

6 Experiments

In the experiments that follow, the part of the matrix A representing normal objects will usually consist of 1000 rows and 30 columns. The 30 columns represent a set of attributes about each object – we assume that these are intrinsic attributes and that a threat is forced to correlate with a target in the values of at least some of these attributes. Each dataset has a small number of additional rows added to represent a terrorist group. The results are for the first random dataset of each kind generated – no selection of datasets to provide better than average results was made. Many of our experiments were more clear-cut than the examples reported here. We use the datasets described in Figure 1.

In plots of two- or three-dimensional space derived from SVD, points corresponding to normal objects are shown as (blue) dots, the target is shown as a (red) star, and the points corresponding to terrorists as (blue) squares. In plots involving SDD, the color and shape coding comes from the SDD classification. The rows of U are plotted in three dimensions but the points are labelled by their classification from the top three levels of the SDD hierarchical ternary classification. Color is used as the indicator for the first level (red = +1, green = 0, blue = −1) and shape as the indicator for the subsequent two levels like this:

+1	+1	dot	0	+1	+	−1	+1	diamond
+1	0	circle	0	0	star	−1	0	triangle down
+1	−1	cross	0	−1	square	−1	−1	triangle up

Dataset 1. The results for Dataset 1 are shown in Figures 2, 3, 4, 5, 6, 7, 8, and 9. This dataset consists of one large 30-dimensional cluster with a much smaller diffuse cluster centered at one of its points. This situation illustrates that basic ability of these matrix decompositions to see clusters that fall within other clusters.

Figure 2 shows the positions corresponding to each object plotted in 3 dimensions. The SVD does not have any information about either the 'target' (the object around whom the terrorist cluster is centered) nor about the terrorist cluster. We can center the terrorist cluster around a target point without loss of generality, since the target point can then be deleted from the dataset if desired without changing the results in any significant way. Hence the target object can be regarded as either a genuine target with which the terrorist group's attributes are connected, or a mathematical fiction used to generate a correlated terrorist group.

The labelling in the figure is done externally; the SVD does not 'detect' the terrorist cluster but simply arranges the points in such a way that the terrorist cluster is easily detectable by inspection. In the list of objects ordered by distance from the origin, the last 15 entries are: 849, 127, 630, 179, 931, 1010, 1009, 1006, 1, 1002, 1004, 1005, 1001, 1008, 1007, 1003, and row 1 is the target object. Hence the terrorist cluster is easily found among the objects farthest from the origin.

The SVD also has the useful property that the point corresponding to the 'target' is placed far from the origin. Hence, the activities of a terrorist group may reveal their target simply because they have attributes that correlate with that object.

Dataset	Description
1	In 30 dimensions, 1000 points normally distributed with variance 1, terrorist group of size 10 normally distributed with variance 1 centered at one of the points
2	In 30 dimensions, 1000 points normally distributed with variance 1, terrorist group of size 10 normally distributed with variance 0.5 centered at one of the points
3	In 30 dimensions, 100 points normally distributed with variance 1, 100 clusters of 10 points normally distributed with variance 1 whose centers are the original points, terrorist group of size 10 normally distributed with variance 1 around a random one of the second level points
4	In 30 dimensions, 100 points normally distributed with variance 1, 100 clusters of 10 points normally distributed with variance 1 whose centers are the original points, terrorist group of size 10 normally distributed with variance 1 around a random one of the second level points, weight on the point used as the center of the terrorist group (the 'target') increased to 1.2
5	In 30 dimensions, 100 points normally distributed with variance 1, 100 clusters of 10 points normally distributed with variance 1 whose centers are the original points, 20 groups of size 10 normally distributed with variance 1 around random second level points, one of these groups chosen as the terrorist group
6	In 30 dimensions, 5000 points normally distributed with variance 1, terrorist group of size 10 normally distributed with variance 1 centered at one of the points
7	In 30 dimensions, 1010 points normally distributed with variance 1, but 70% of the entries set to zero. The terrorist group is then correlated with a randomly chosen target row like this: if the target attribute has a non-zero value for a particular attribute, the corresponding attribute of a terrorist is changed to the a normally distributed value whose mean is the value of the target attribute and whose variance is 1; otherwise the terrorist attribute is not changed
8	In 30 dimensions, 1000 points Poisson distributed with mean 1, terrorist group of size 10 Poisson distributed with mean 1 centered at one of the points. Mean subtracted to approximate zero mean data
9	In 30 dimensions, 1000 points uniformly 0s or 1s, terrorist group of size 10 in which each new point and attribute is uniformly 0 or 1 if the corresponding attribute of the target was 1; otherwise only a slight probability of being set to 1

Figure 1: Datasets used in experiments

Figure 3 is the same plot, but with only the points of objects correlated with the target remaining. This, of course, requires knowledge of the target. In practice, this might be obtained either by inspection of plots such as Figure 2 or because only a number of worthwhile targets are known to be present in the dataset. Correlation here means being on the same side of a hyperplane through the origin as the target in 15 dimensions; this property will not, in general, hold exactly when the plot uses only 3 dimensions.

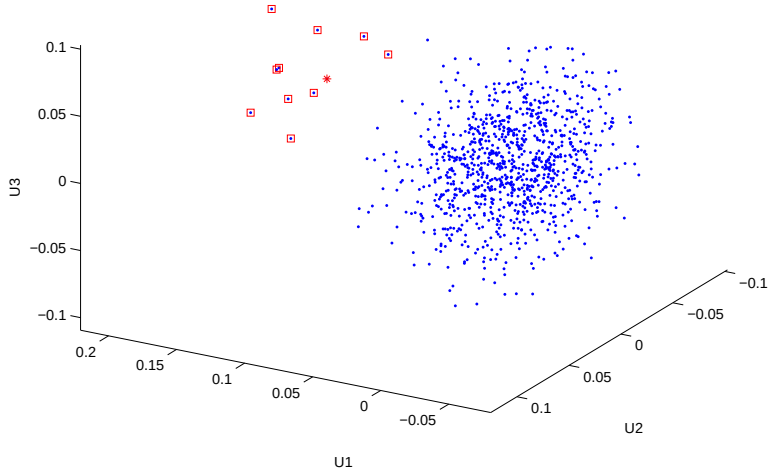

Figure 2: Dataset 1, SVD clustering showing positioning of the terrorist cluster

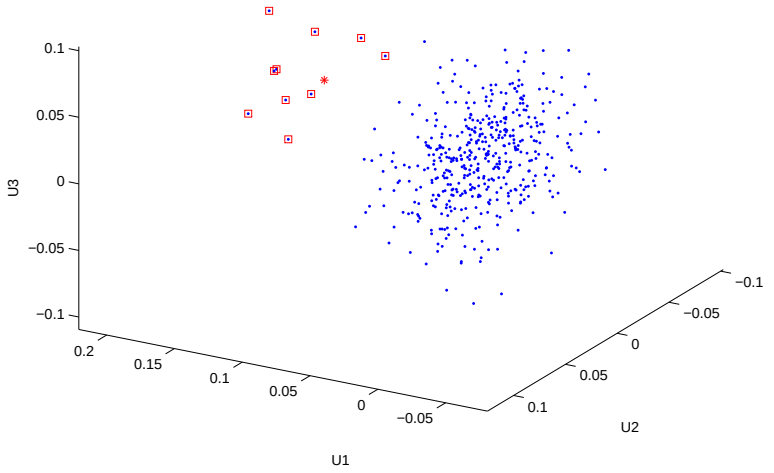

Figure 3: Dataset 1, 474 objects correlated with the target

Figure 4 is the same plot, but with only those points farther than the median distance from the origin (in 15 dimensions) remaining. This discards those points that are not interesting in the sense that they show only weak correlation with all of the other points. Removing these points does not require knowledge of the target and, reassuringly, both the target and all of the terrorists remain in the plot. Figure 5 shows the points that remain when only those points farther than 1.3 times the median distance from the origin are retained. As expected, this removes even more of

the points, leaving a residue of approximately 10% of the original objects for further consideration.

Figures 6 and 7 show the effect of combining both of these techniques. In the second case, the number of objects requiring further consideration is only 6% of the original population, and both the target and all of the terrorists are still present (and increasing well-separated from the points corresponding to other objects).

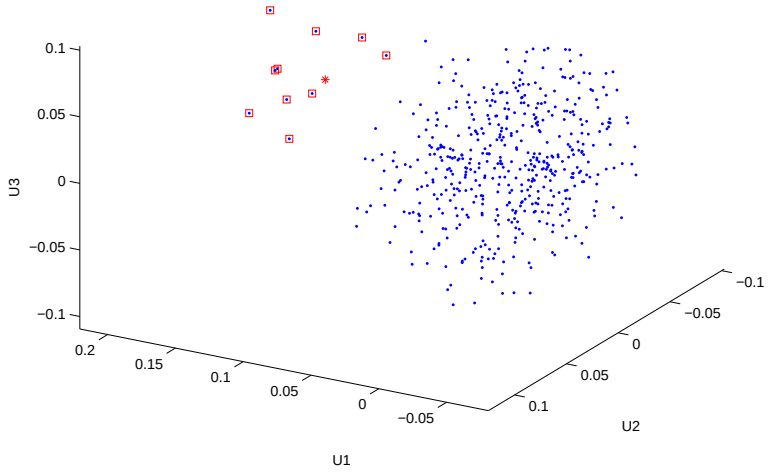

Figure 4: Dataset 1, 504 objects greater than median distance from the origin

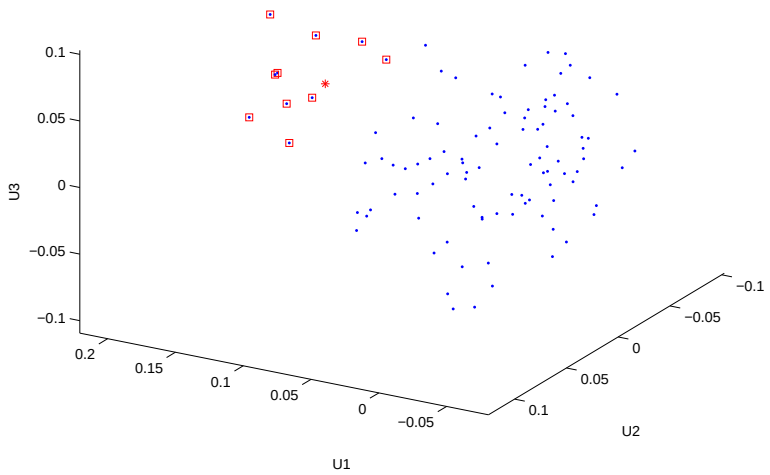

Figure 5: Dataset 1, 101 objects greater than 1.3 times the median distance from the origin

SVD is unable to predict or classify points as likely to be terrorists, other than by distance from the origin. In contrast, techniques based on SDD generate a hierarchical classification allowing the algorithm itself to predict at least the degree of anomaly of each point. When the target is known, SDD-based techniques can also report those other points that fall into the same branch of the hierarchical classification at any depth. This is clearly more powerful.

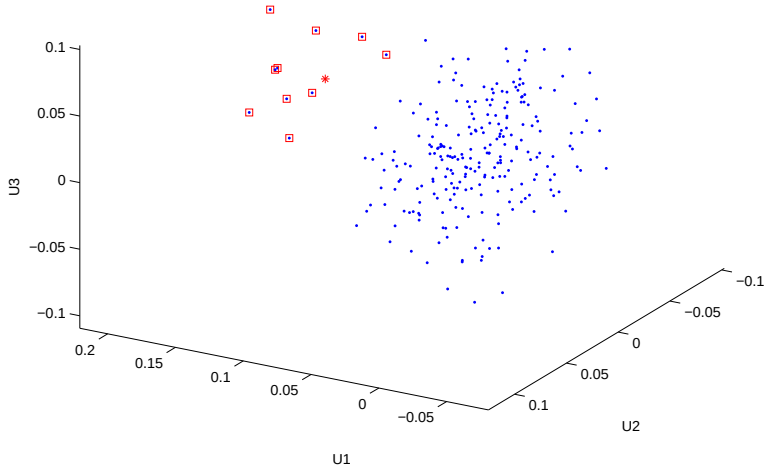

Figure 6: Dataset 1, 253 objects greater than median distance from the origin and correlated with the target

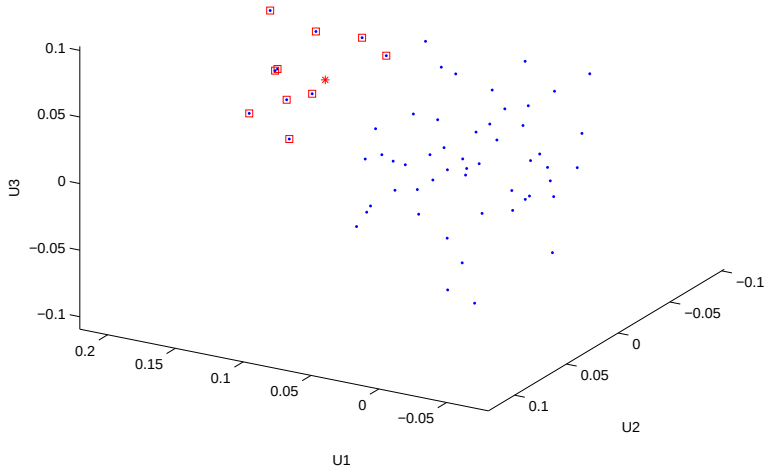

Figure 7: Dataset 1, 61 objects greater than 1.3 times the median distance from the origin and correlated with the target

Figure 8 shows the points plotted at positions determined by SVD (so the positions are the same as in the preceding figures) but with color and position determined by the top three levels of the hierarchical classification. In this case, the target and terrorist group are represented by $\times$ (and almost no other points are). However, the crosses are of two different colors because the terrorist group falls into the classes down the $0, +1, -1$ and $+1, +1, -1$ branches. Without knowledge of the target it would be difficult to know which group represented the terrorist group, since most of the other points are distributed across of number of other branches in groups of moderate size. In general, these results are typical of applying SDD directly to the data matrix – the values

themselves correctly identify groupings among the objects but are not sufficient to distinguish the terrorist cluster from the other groups without more information.

Figure 9 shows a similar plot, but with the SDD color and position labelling derived from the JSS methodology. This result is much more useful – the terrorist cluster is completely identified at the top level (it is a different color). In this case, the technique itself is able to identify the terrorist cluster and report it exactly. Notice that almost all of the terrorist group fall into the same branch as the target (the $+1, +1, +1$ branch labelled by red dots) even after three levels.

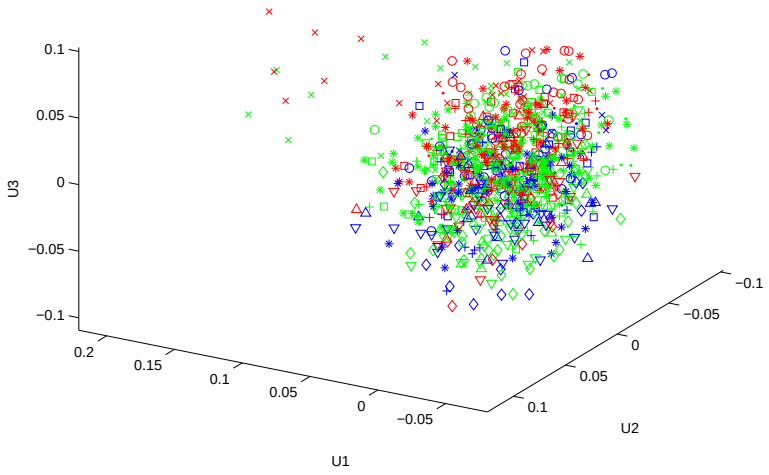

Figure 8: Dataset 1, position from SVD, color and shape from SDD.

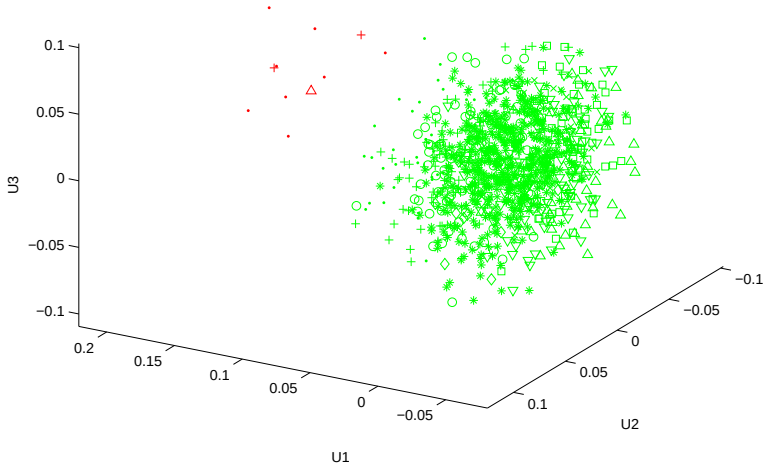

Figure 9: Dataset 1, position from SVD, color and shape from JSS. The terrorist cluster is identified by the JSS hierarchical classification

Dataset 2. In the previous experiment, the terrorist cluster had the same variance as the base cluster. Hence, it is likely that points from the terrorist cluster are overrepresented among points far from the origin in the original 30-dimensional space. We now show that this is not the reason for the quality of the SVD plot by repeating the experiment with the variance of the terrorist cluster at 0.5. We now expect points from the terrorist cluster to remain inside the background cluster on average.

The results for Dataset 2 are shown in Figures 10, 11, 12, 13, 14, 15, and 16. Figure 10 shows that the terrorist cluster is still visible although there are other points that are quite far from the origin. If the terrorist cluster were unlabelled, the terrorist cluster would certainly arouse suspicion, but another 10–20 points would also require further investigation. As before, about half the objects are correlated with the target, about half are farther than the median distance from the origin, and about a quarter are both. When the required distance from the origin is increased, the set requiring further attention is again reduced to about 6% of the dataset and includes all of the terrorist group.

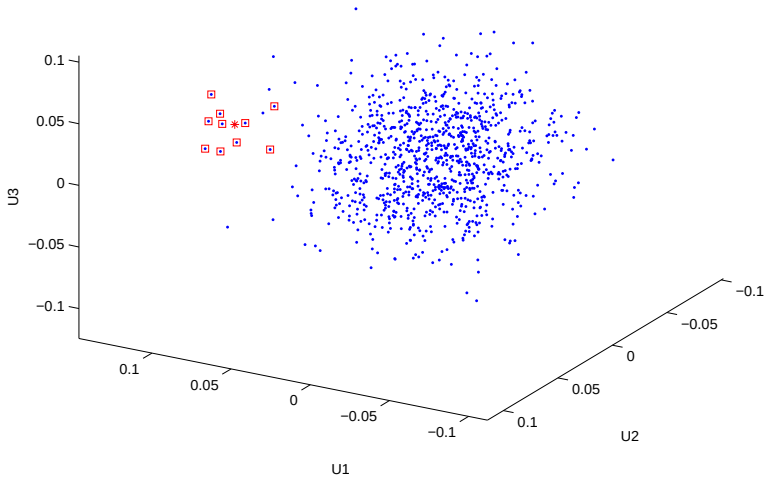

Figure 10: Dataset 2, SVD clustering showing positioning of the terrorist cluster

The plot in Figure 16 shows the classification by the JSS methodology in which the terrorist group is correctly identified, with one false positive. Note the presence of a cluster of two objects labelled by a blue circle and a blue +. These represent the objects who are most 'opposite' to the terrorist cluster. They therefore deserve special consideration because of the possibility that they represent alternative personas for members of the terrorist cluster – if two individuals are never at the same place at the same time, they may be two completely unconnected people – but the *may* be the same person using two identities.

This is arguably an easy dataset, but not entirely trivial because the fuzzy central limit theorem suggests that, given enough data, and given that normalization takes place after the data is collected, we can expect that many parts of a real dataset should look as if they were generated by a normal distribution.

Dataset 3. We now consider a dataset with following structure: 100 points are generated, normally distributed around 0 with variance 1. 100 clusters of 10 points are generated, normally distributed with variance 1 with centers at each of the original points. A terrorist cluster of size 10,

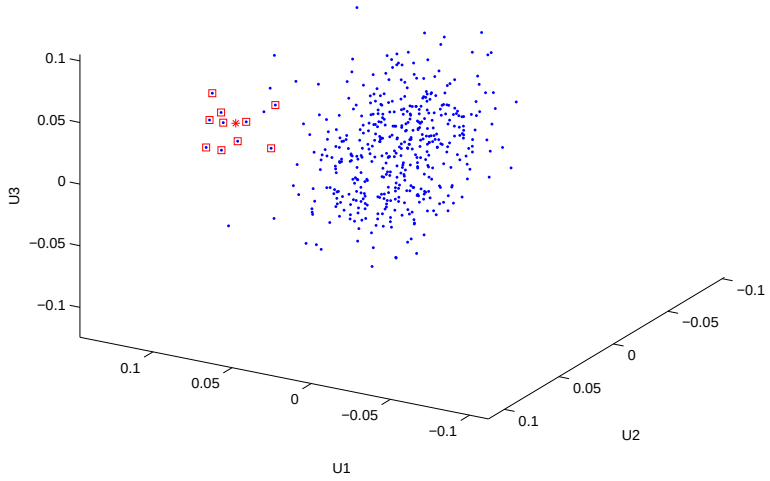

Figure 11: Dataset 2, 462 objects correlated with the target

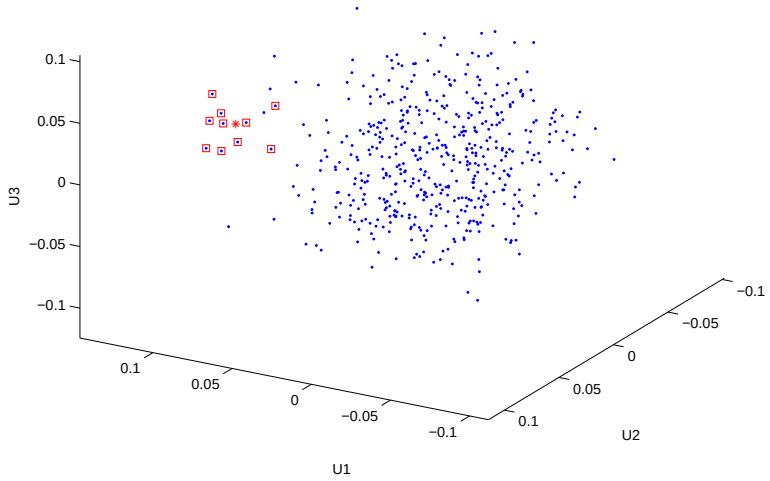

Figure 12: Dataset 2, 504 objects greater than median distance from the origin

normally distributed with variance 1 is generated around a random one of the second level points. So rather than a single background cluster around zero, we have a large set of background clusters with many different centers. This better represents the grouped structure and many connections in real sets of people. The results for Dataset 3 are shown in Figures 17, 18, 19, 20, 21, 22, and 23.

As Figures 17, 18, and 19 show, it is now much more difficult to identify the terrorist cluster without the external labelling. Restricting attention to points much further than the median distance from the origin reduces the number of objects to consider but, for the first time, eliminates some of the terrorist group from consideration.

The JSS classification, shown in Figure 23, correctly groups all but one member of the terrorist group in the same cluster as the target, and includes 12 other objects (false positives). Hence a

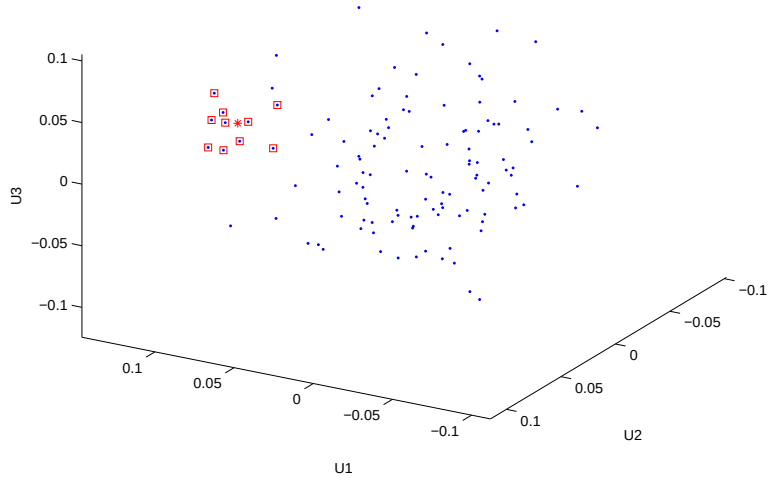

Figure 13: Dataset 2, 120 objects greater than 1.3 times the median distance from the origin

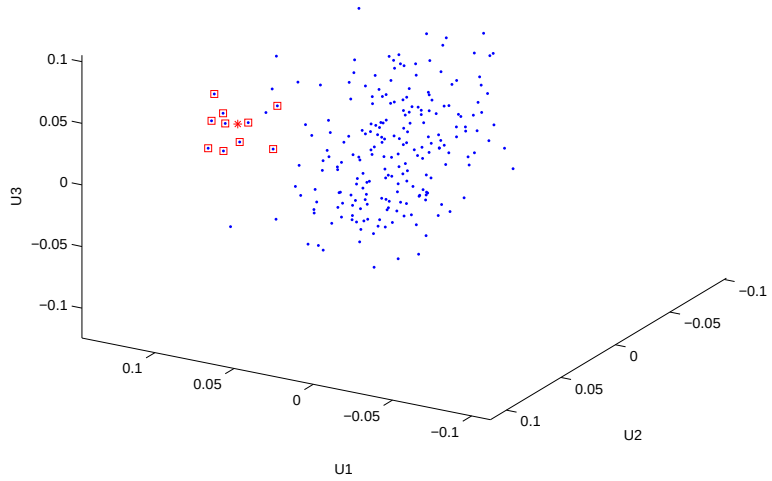

Figure 14: Dataset 2, 226 objects greater than median distance from the origin and correlated with the target

total of 2.5% of the population would be selected for further scrutiny and half of them are terrorists.

Dataset 4. When the target is known, but there is a large amount of background correlation, the weight of the target row can be increased. This has the effect of moving the point corresponding to the target farther from the origin, but also tends to pull other points that are correlated with the target farther from the origin as well (Fact 3). We increase the weight on the target row by a modest factor of 1.2. Even this small change produces a visible movement of the location of points and, in particular, improves the JSS classification.

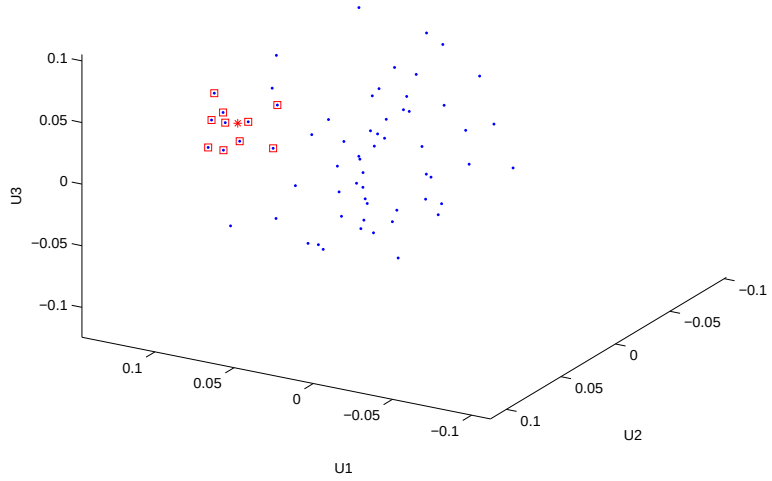

Figure 15: Dataset 2, 64 objects greater than 1.3 times the median distance from the origin and correlated with the target

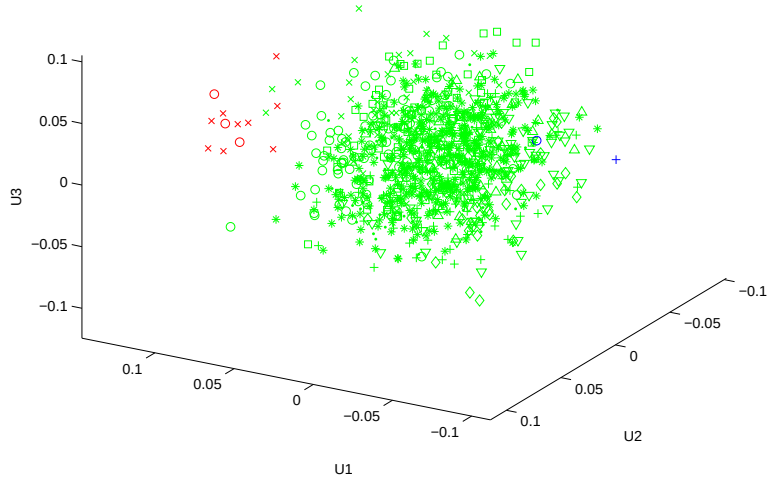

Figure 16: Dataset 2, position from SVD, color and shape from JSS. The terrorist cluster is identified by the JSS hierarchical classification

The results for Dataset 4 are shown in Figures 24, 25, 26, 27, 28, 29, and 30. The results are almost identical to the previous case, except for the JSS classification (Figure 30). Here the group containing the target contains exactly 9 other objects, all from the terrorist group. Hence, with modest weight added, JSS is able to determine the terrorist group with no false positives and one false negative.

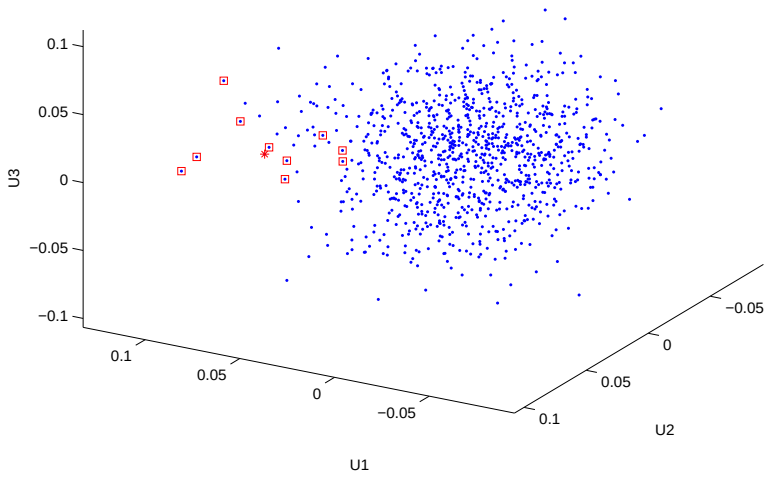

Figure 17: Dataset 3, SVD clustering showing positioning of the terrorist cluster

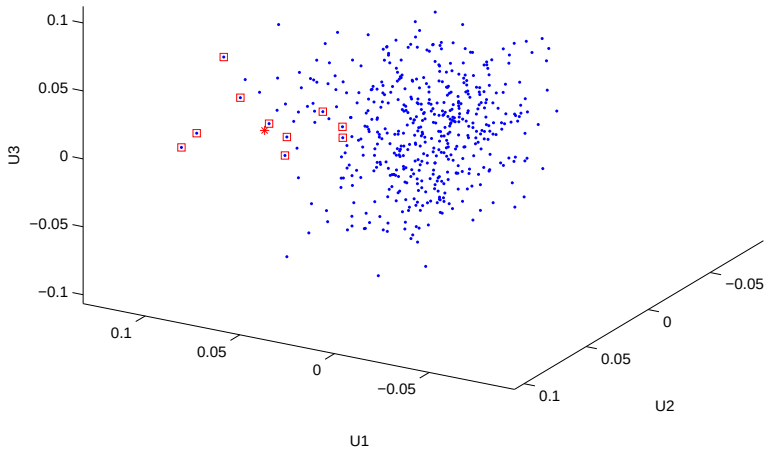

Figure 18: Dataset 3, 503 objects correlated with the target

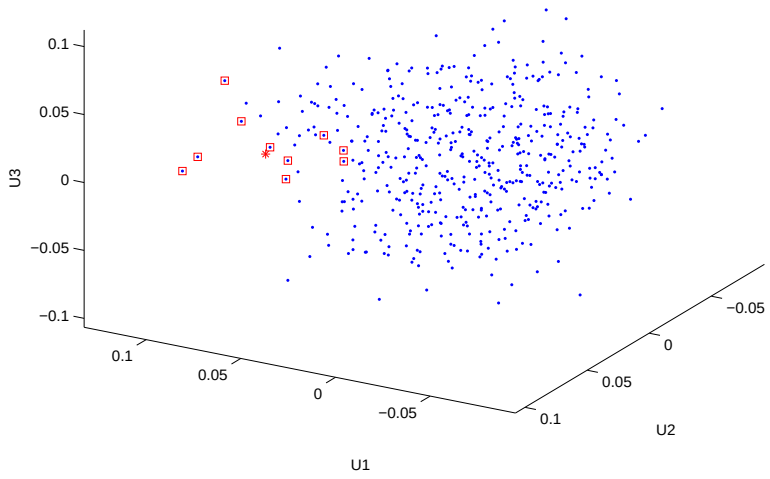

Figure 19: Dataset 3, 504 objects greater than median distance from the origin

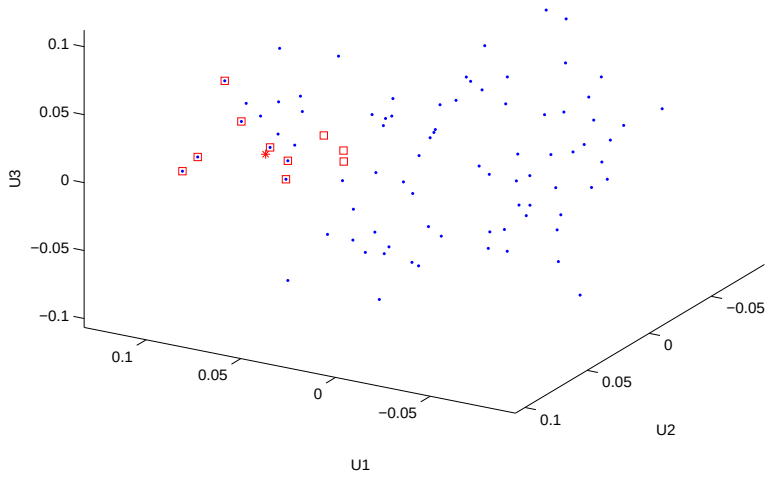

Figure 20: Dataset 3, 84 objects greater than 1.3 times the median distance from the origin. Boxes without a dot inside them represent members of the terrorist group who have been removed from consideration. For the first time, some members of the terrorist cluster are not detected.

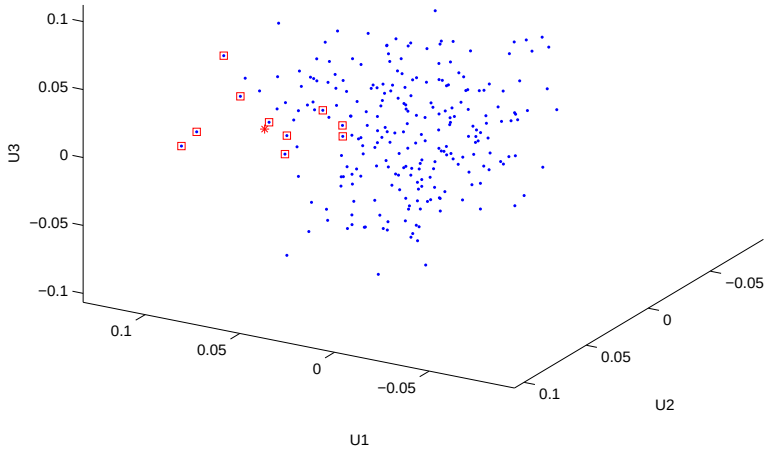

Figure 21: Dataset 3, 248 objects greater than median distance from the origin and correlated with the target

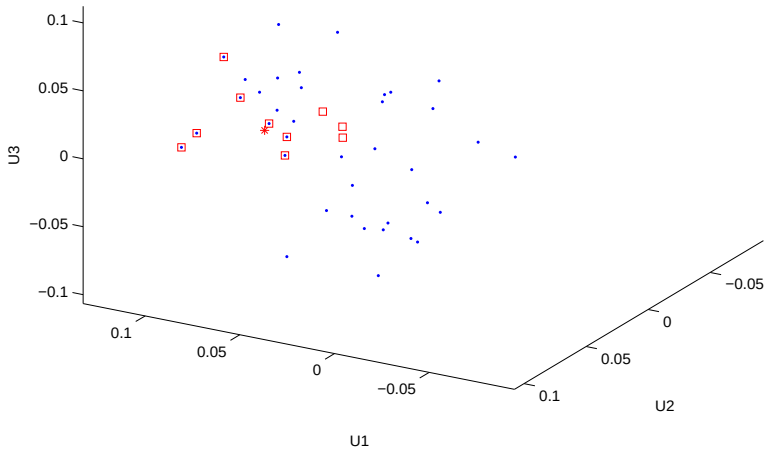

Figure 22: Dataset 3, 38 objects greater than 1.3 times the median distance from the origin and correlated with the target.

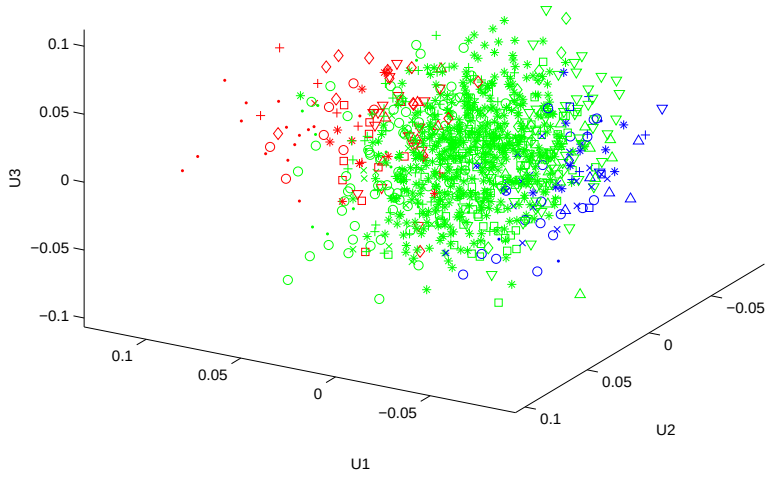

Figure 23: Dataset 3, position from SVD, color and shape from JSS. The terrorist cluster (the red dots) is identified by the JSS hierarchical classification, with some false positives

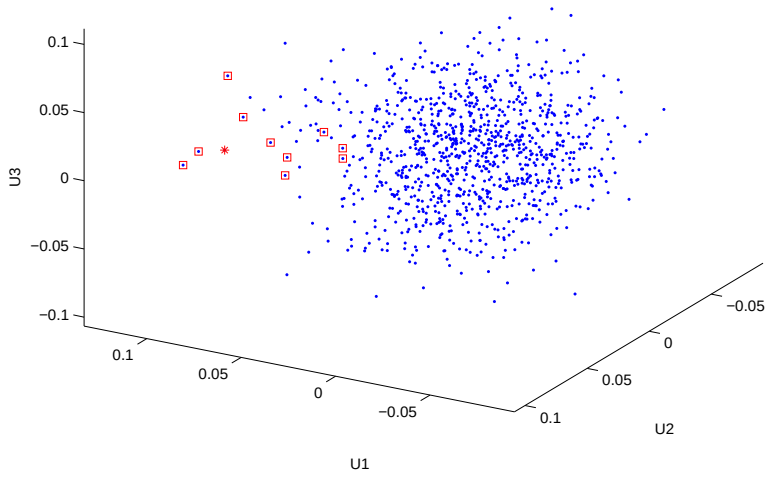

Figure 24: Dataset 4, SVD clustering showing positioning of the terrorist cluster

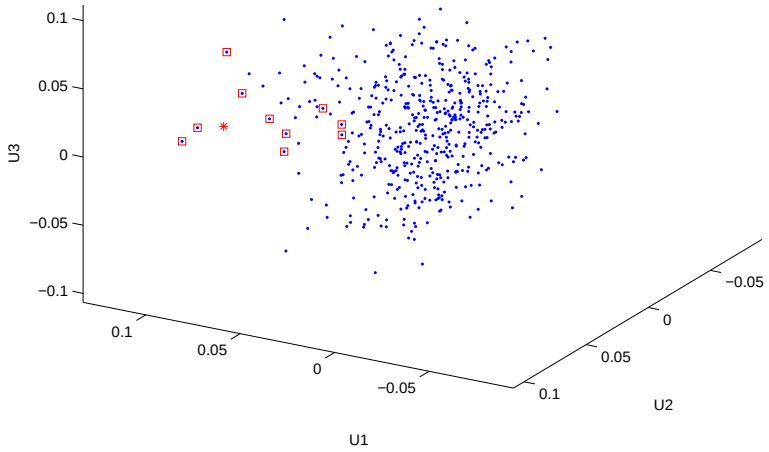

Figure 25: Dataset 4, 504 objects correlated with the target

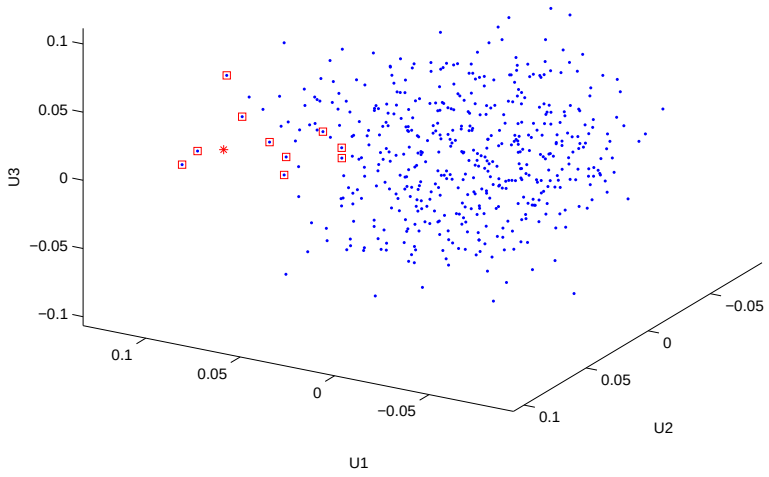

Figure 26: Dataset 4, 504 objects greater than median distance from the origin

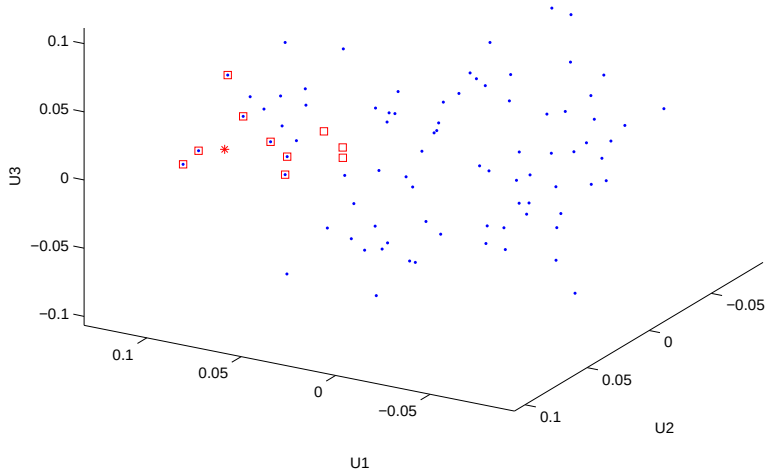

Figure 27: Dataset 4, 84 objects greater than 1.3 times the median distance from the origin

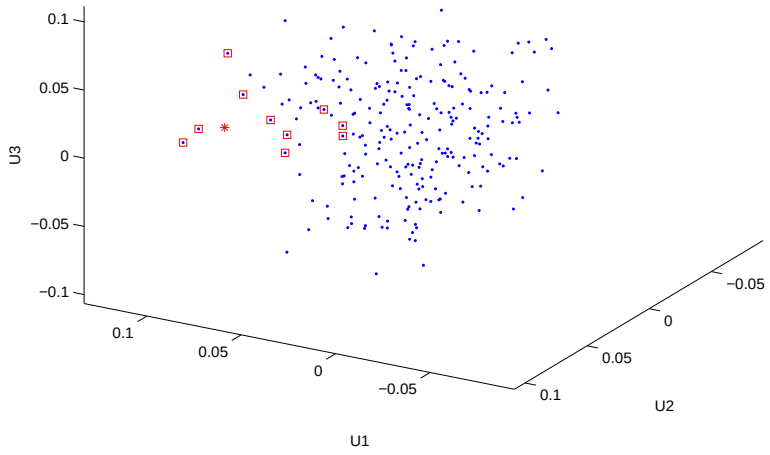

Figure 28: Dataset 4, 248 objects greater than median distance from the origin and correlated with the target

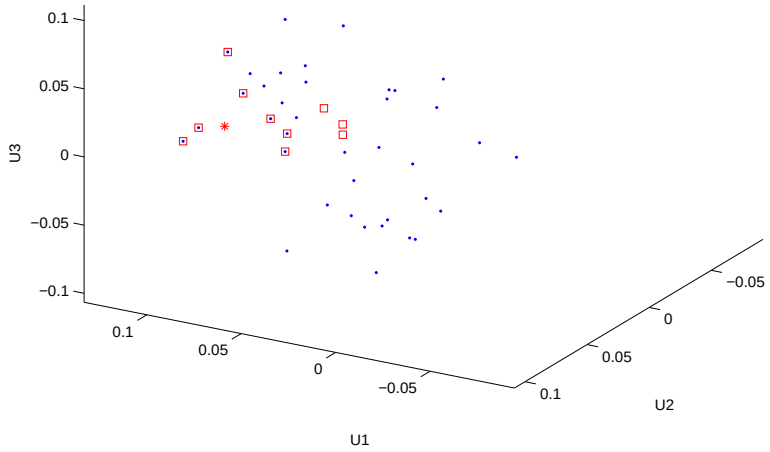

Figure 29: Dataset 4, 38 objects greater than 1.3 times the median distance from the origin and correlated with the target. Once again some members of the terrorist group are not detected

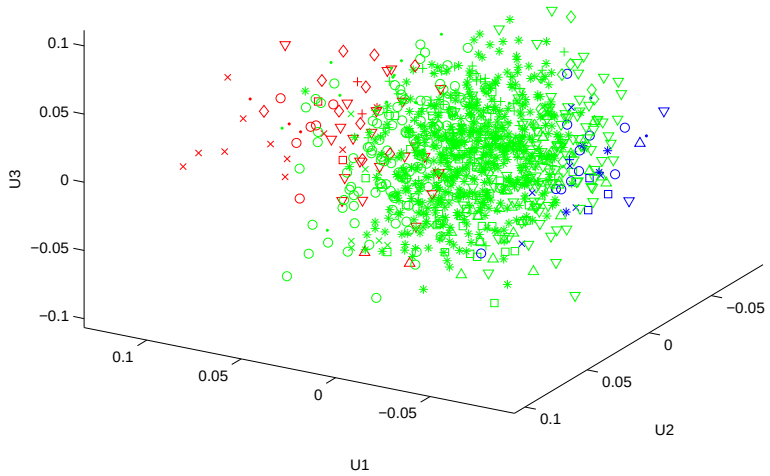

Figure 30: Dataset 4, position from SVD, color and shape from JSS. The terrorist cluster is identified by the JSS hierarchical classification

Increasing the weight on the target does not necessarily improve performance. Beyond a certain point, the target appears in a group by itself. Although the terrorist cluster remains in a close branch of the hierarchical classification, it becomes harder to select it with confidence. For example, if the target weight is increased to 4, the target and the terrorist group are similarly classified, but only at level four of the classification tree.

Dataset 5. In the previous datasets (Datasets 3 and 4), the terrorist cluster was still distinguished because it was the only cluster at the 'third' level. We now generate a dataset with 100 points, normally distributed around 0 with variance 1. 100 clusters of 10 points are generated, normally distributed with variance 1 around each of the original points, and 20 clusters of size 10 normally distributed with variance 1 are generated around randomly chosen points in the second level. One of these 'third' level clusters is chosen as the terrorist cluster and its center as the target.

The results for Dataset 5 are shown in Figures 31, 32, 33, 34, 35, 36, and 37. Techniques based on SVD now find it difficult to detect the terrorist cluster, although using correlation and distance from the origin is still able to reduce the pool of potential terrorists to 12% of the total if the target is known.

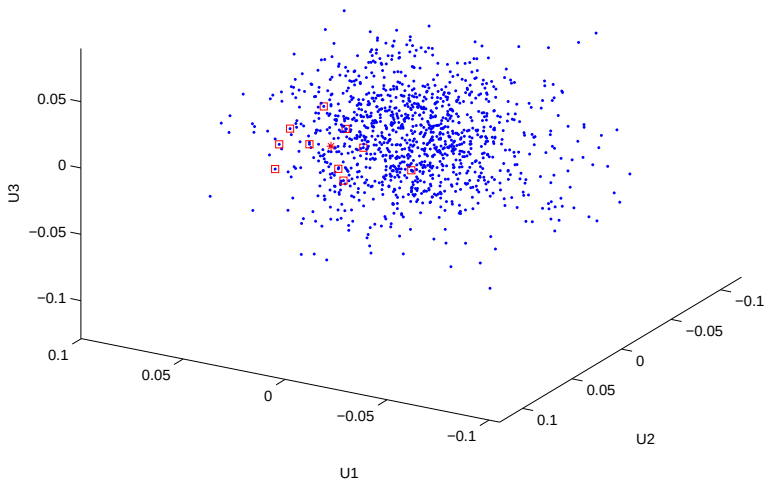

Figure 31: Dataset 5, SVD clustering showing positioning of the terrorist cluster

The JSS methodology remains strong, selecting the entire terrorist cluster and fewer than 10 other points (Figure 37).

In the dataset, the local environment of each of the second level cluster centers is the same and we can choose any of them as possible terrorist clusters. On the other hand, the local environment of all of the other points is quite different. Figure 38 shows the sizes of the sets of points correlated with a particular point, when that point is a second-level cluster center (a possible target) and when it is one of the other points.

Those points that are targets have neighborhoods that start out smaller and shrink more rapidly than the neighborhoods of points that are not targets. The difference between the two types of points is marked, even by the third round.

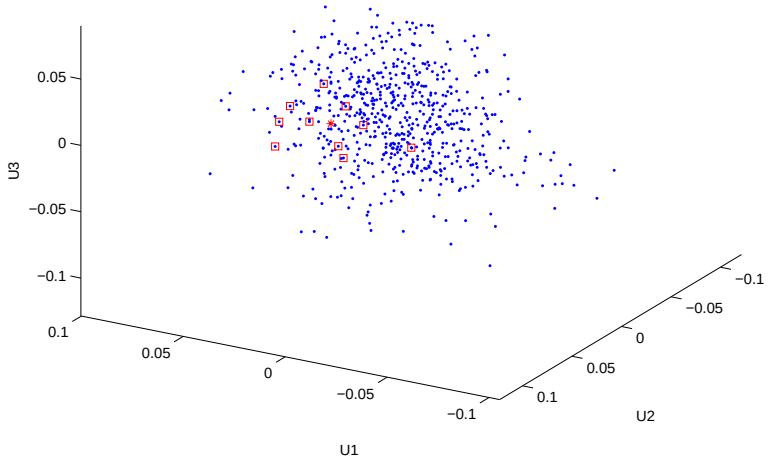

Figure 32: Dataset 5, 655 objects correlated with the target

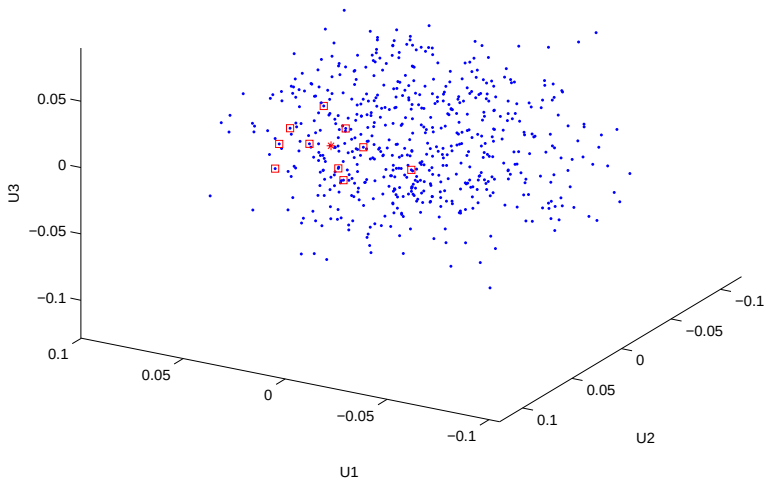

Figure 33: Dataset 5, 599 objects greater than median distance from the origin

Dataset 6. In our experiments so far, the number of terrorists has been about 1% of the total number of objects. This fraction is too large to be realistic, even if a substantial prescreening process is applied before this kind of data mining is used.

Figure 39 shows the three-dimensional plot of a dataset with 5000 rows, normally distributed around the origin with variance 1, with a 10-terrorist cluster normally distributed with variance 1 generated around one of the ordinary objects.

The results for Dataset 5 are shown in Figures 39, 40, 41, 42, 43, 44, and 45. The smaller relative size of the terrorist cluster means that it has less effect on the remainder of the points, and particularly on the target. However, a significant number of terrorist points are well outside

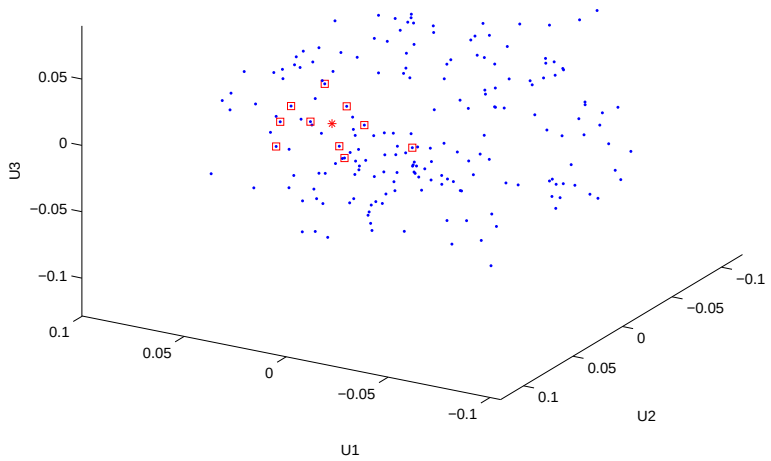

Figure 34: Dataset 5, 196 objects greater than 1.3 times the median distance from the origin

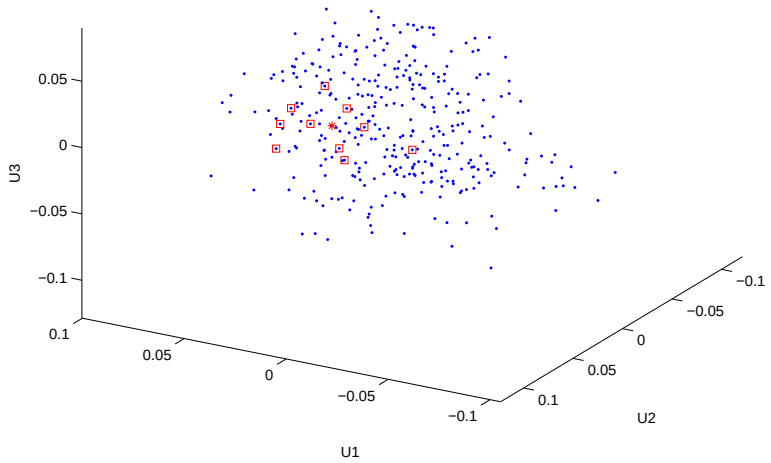

Figure 35: Dataset 5, 354 objects greater than median distance from the origin and correlated with the target

the main cluster. As expected, discarding both points uncorrelated to the target and points close to the origin leaves the same fraction of points, but this fraction is a larger number and it is correspondingly harder to be sure of the location of the terrorist cluster. Aggressive winnowing reduces the population to 264, 5% of the total population, but at the expense of missing three terrorists.

Once again, the JSS classification correctly identifies the terrorist cluster, although with a larger number of false positives.

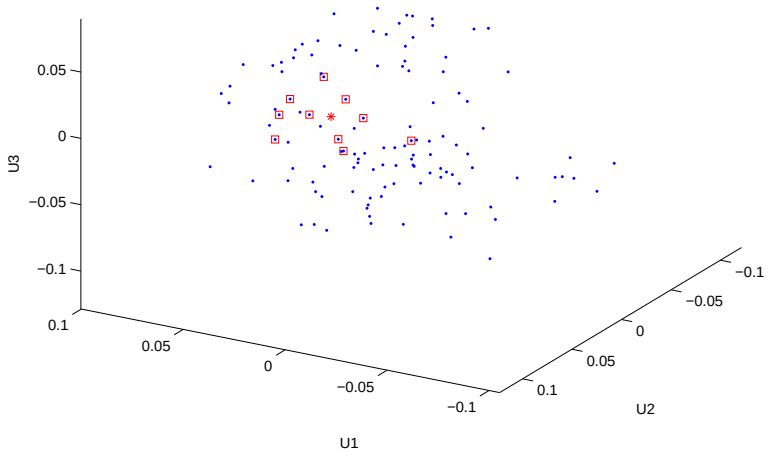

Figure 36: Dataset 5, 121 objects greater than 1.3 times the median distance from the origin and correlated with the target

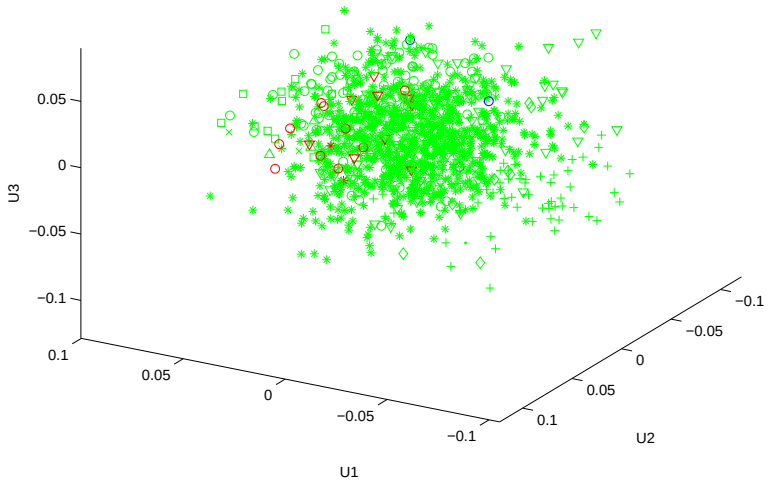

Figure 37: Dataset 5, position from SVD, color and shape from JSS. The terrorist cluster is identified by the JSS hierarchical classification

Dataset 7. Fact 2 suggests that sparseness in datasets will not cause difficulties for SVD. This illustrates one of the strong properties of SVD – it is capable of detecting correlation even between objects that have no (non-zero values of) attributes in common, via higher-order correlations.

The results for Dataset 7 (with 30% non-zeros) are shown in Figures 46, 47, 48, 49, 50, 51, and 52. Techniques based on SVD perform well in this setting, although they are all slightly weaker than in the dense case.

After round	Size of sets correlated with a point that is a target			that is not a target			
1	145	419	199	831	370	586	416
2	20	27	47	513	90	194	150
3			20	461	48	86	78
4				400	42	56	65

Figure 38: Sizes of correlated sets after elimination of uncorrelated objects. Initial size of all sets is 1200.

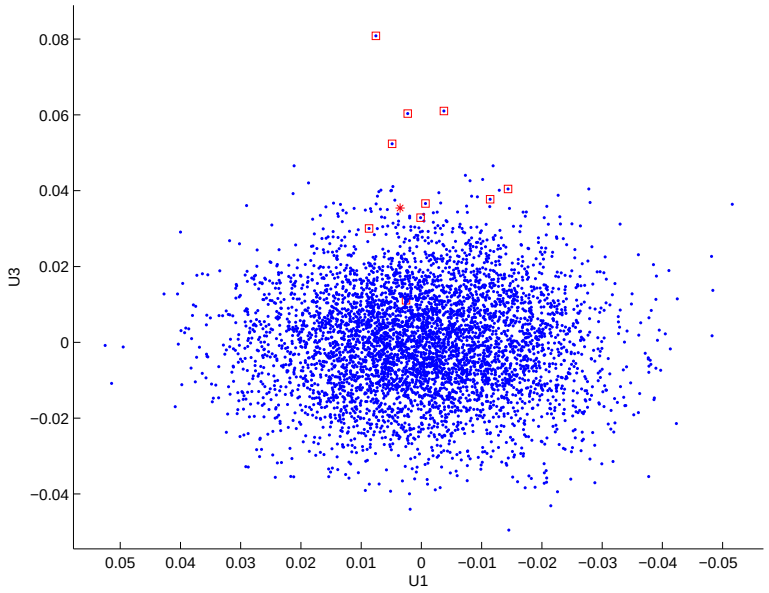

Figure 39: Dataset 6, SVD clustering showing positioning of the terrorist cluster

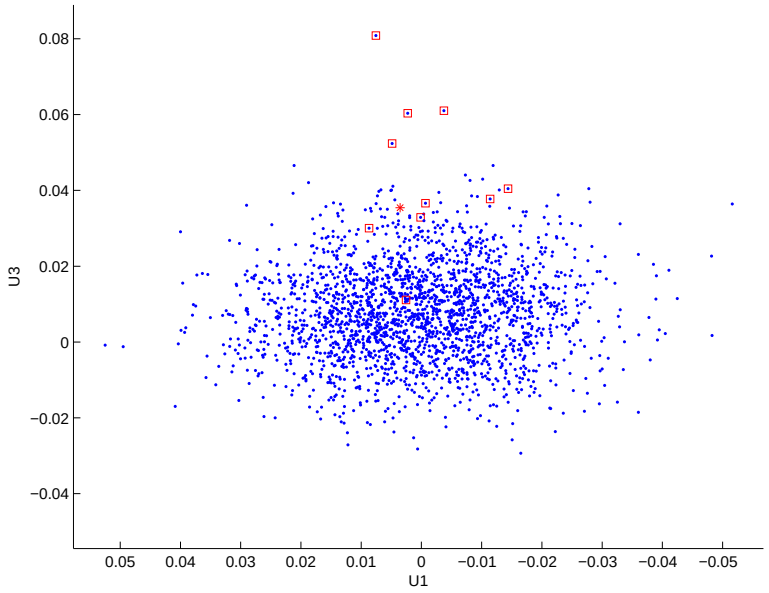

Figure 40: Dataset 6, 2251 objects correlated with the target

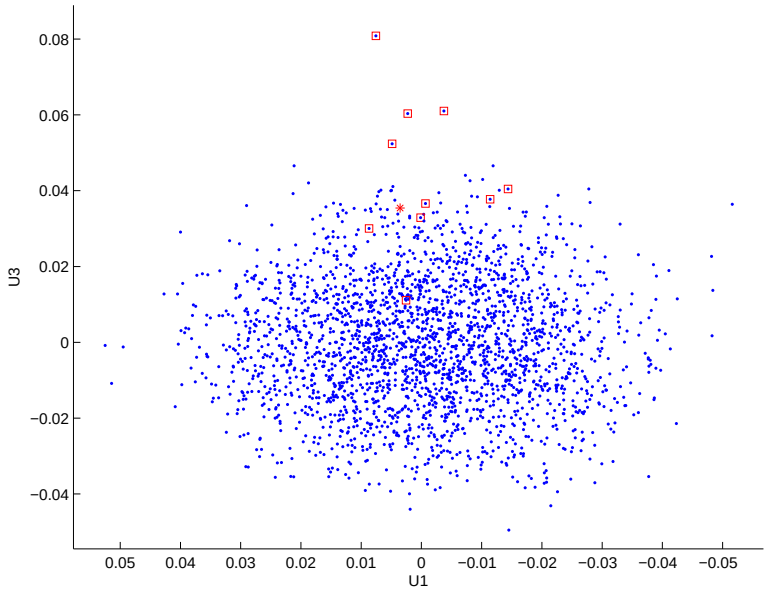

Figure 41: Dataset 6, 2504 objects greater than median distance from the origin

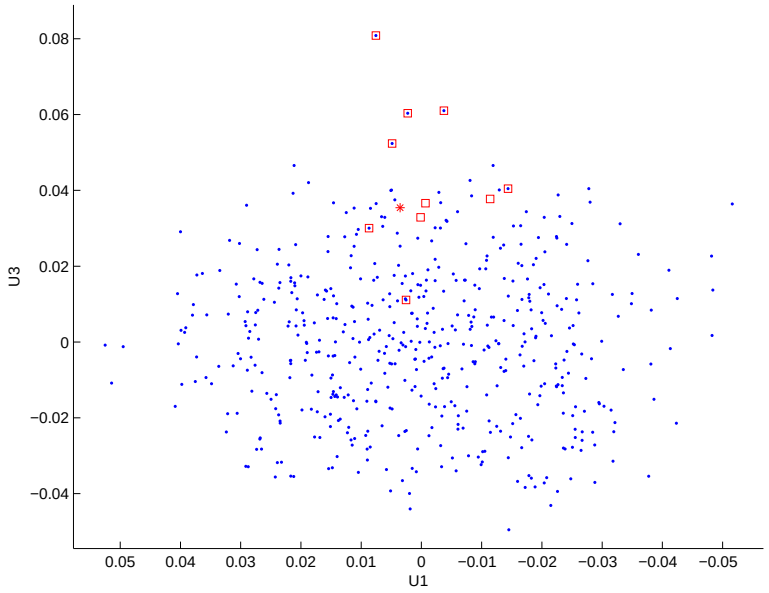

Figure 42: Dataset 6, 528 objects greater than 1.3 times the median distance from the origin

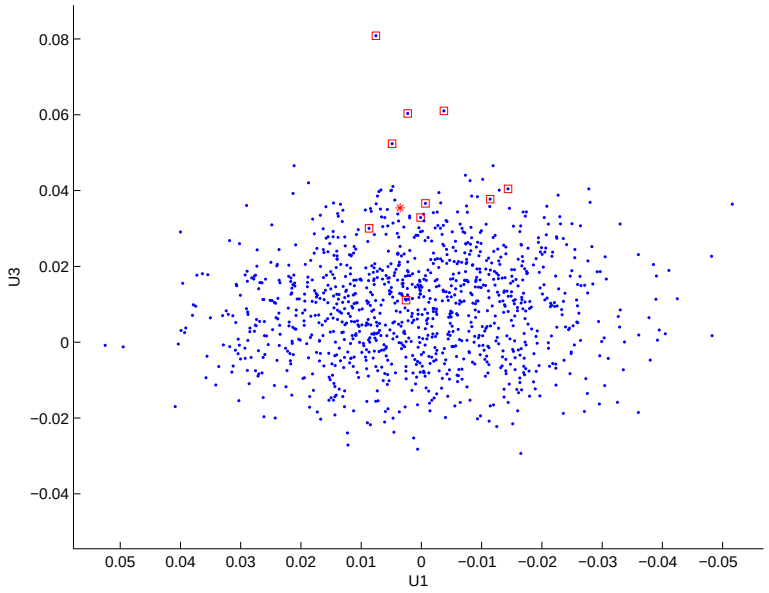

Figure 43: Dataset 6, 1129 objects greater than median distance from the origin and correlated with the target

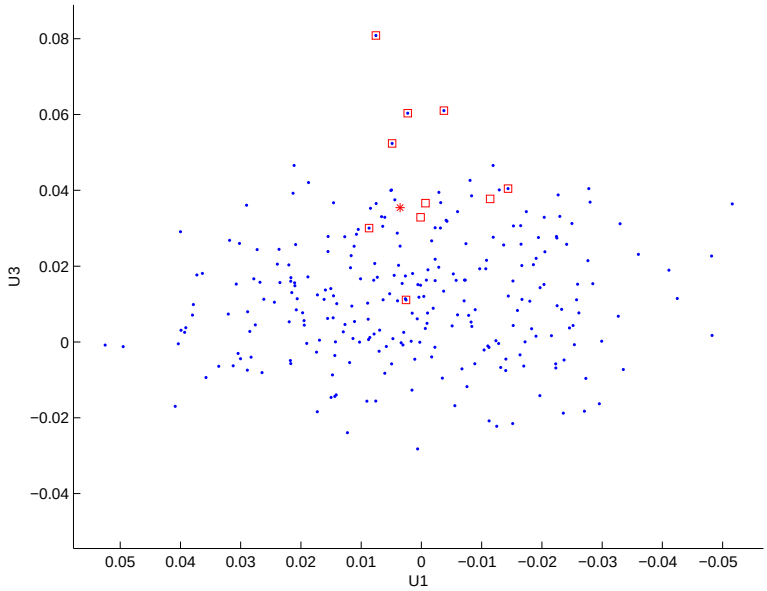

Figure 44: Dataset 6, 264 objects greater than 1.3 times the median distance from the origin and correlated with the target

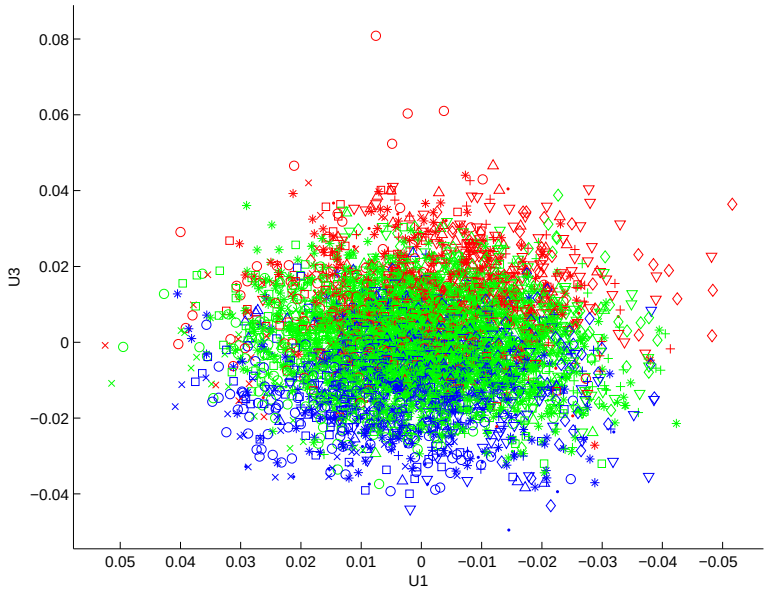

Figure 45: Dataset 6, position from SVD, color and shape from JSS. The terrorist cluster is identified by the JSS hierarchical classification

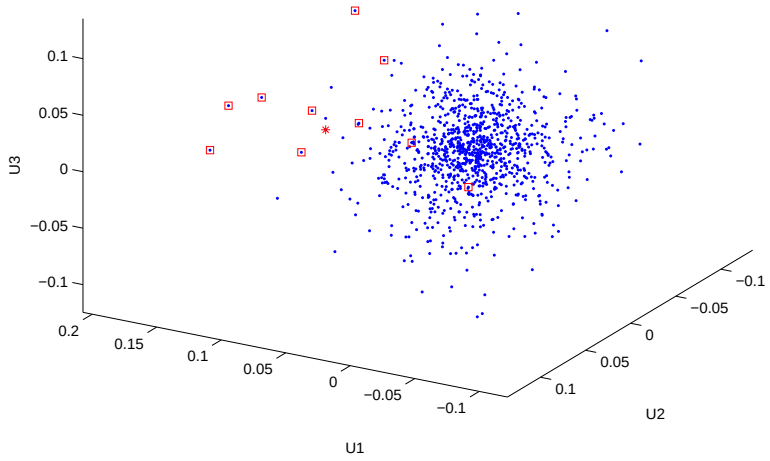

Figure 46: Dataset 7, SVD clustering showing positioning of the terrorist cluster

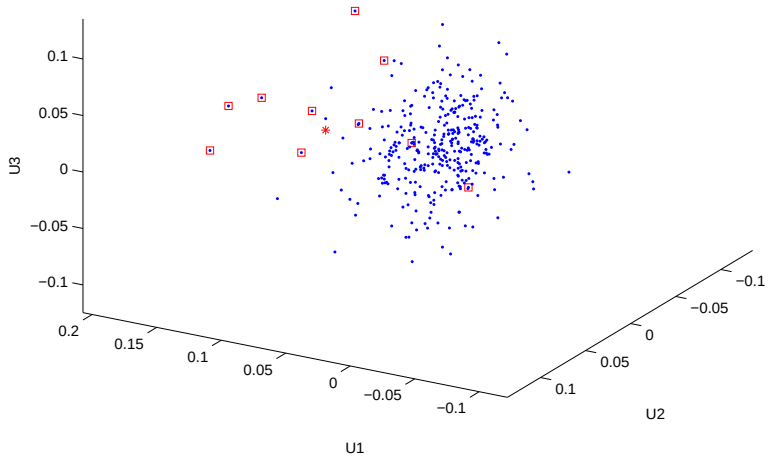

Figure 47: Dataset 7, 358 objects correlated with the target

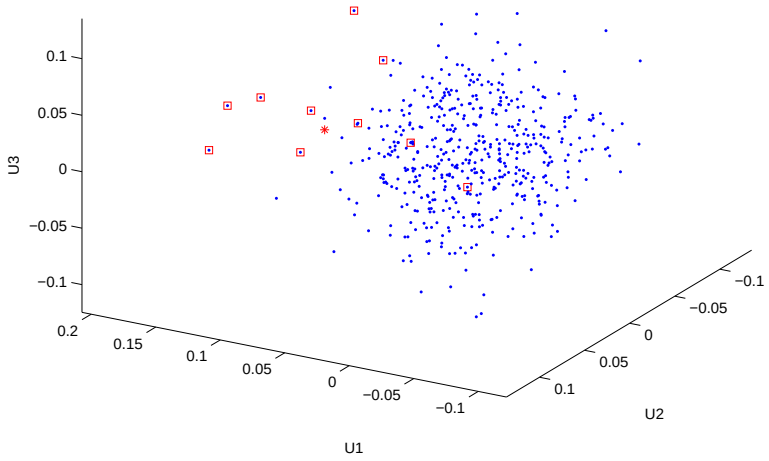

Figure 48: Dataset 7, 504 objects greater than median distance from the origin

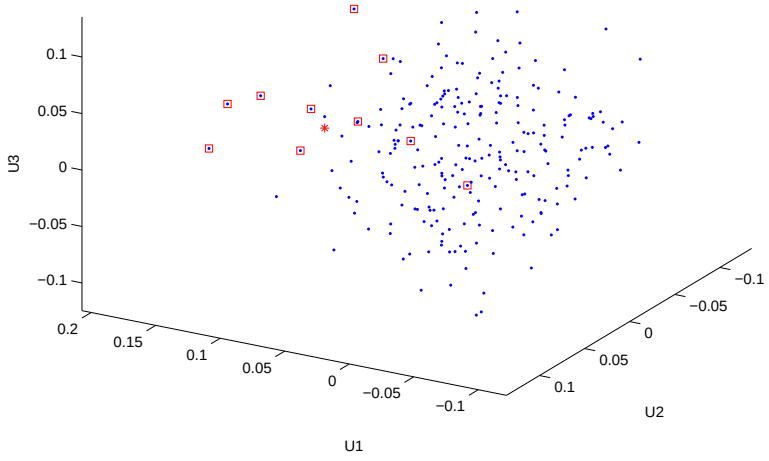

Figure 49: Dataset 7, 248 objects greater than 1.3 times the median distance from the origin

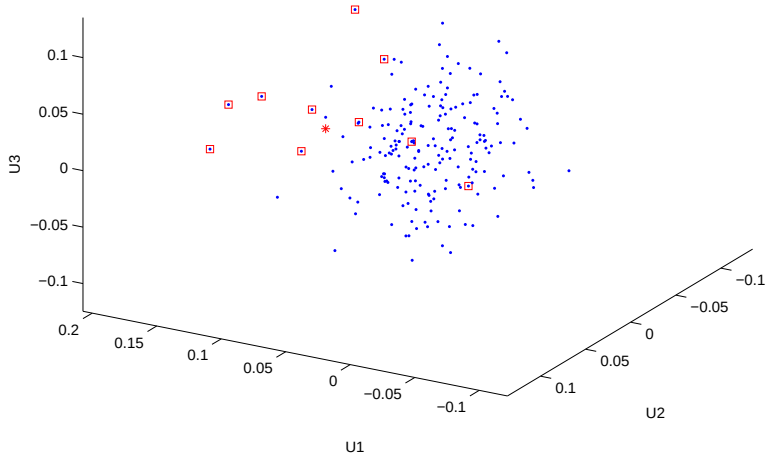

Figure 50: Dataset 7, 205 objects greater than median distance from the origin and correlated with the target

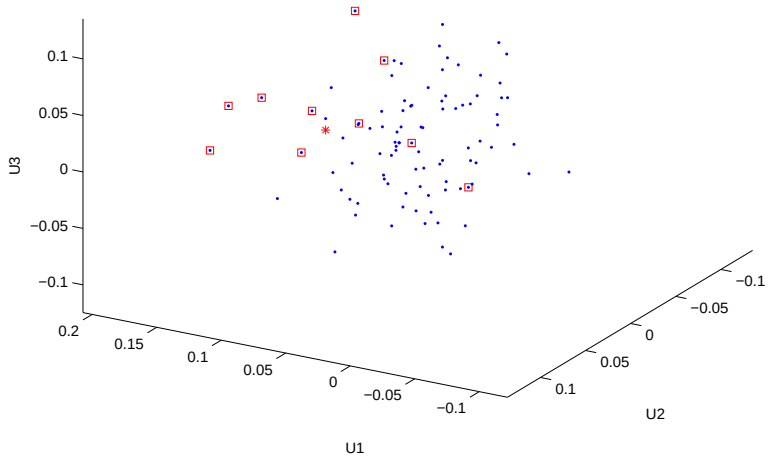

Figure 51: Dataset 7, 96 objects greater than 1.3 times the median distance from the origin and correlated with the target

The classification from JSS does not perform as well as in the dense case. This is not entirely surprising since SDD does not use correlation directly, and the use of indirect correlation in the JSS methodology relies on a weaker result from SVD.

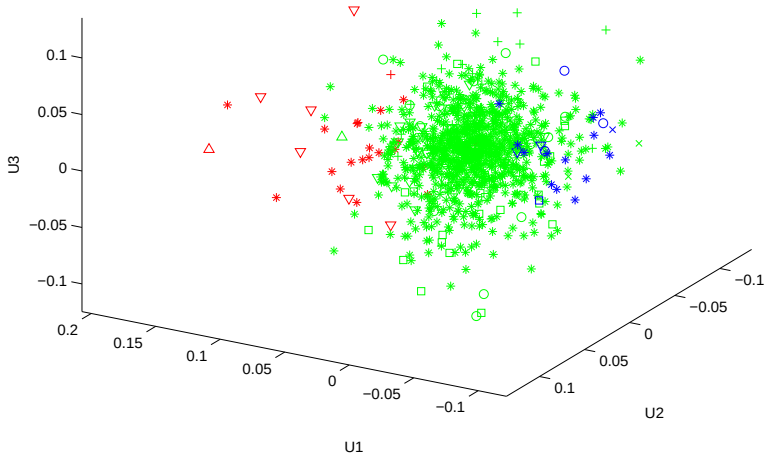

Figure 52: Dataset 7, position from SVD, color and shape from JSS. The terrorist cluster is identified by the JSS hierarchical classification

Dataset 8. We now show that similar effects hold for distributions other than the normal distribution. The Poisson distribution with mean 1 generates many values close to 1, with the frequency decreasing rapidly with magnitude. We build a dataset of a 1000 rows from this distribution, subtracting λ to make the values approximately zero mean as required by SVD.

The results for Dataset 8 are shown in Figures 53, 54, 55, 56, 57, 58, and 59. The results are very similar to those obtained when the data was distributed normally – the terrorist cluster is clearly distinguished in all cases, and the number of points correlated with the target and far from the origin remain about the same. The JSS methodology classifies 7 of 10 terrorists in a group with the target.

Dataset 9. Some settings have data that is binary in nature; each person did, or did not do some action, or does or does not have some particular attribute. We show what happens when the data are restricted to the binary case.

The results for Dataset 9 are shown in Figures 60, 61, 62, 63, 64, 65, and 66. Both SVD-based techniques and the JSS methodology have difficulty in this case – not only are there a limited range and values and hence a limited range of variability, but also the data is effectively sparse because of the number of 0s.

All of the winnowing techniques return about as many points as in the real-valued case but, for the first time, one of the terrorist groups does not show up as a point further from the origin than the median distance; and all but one of the terrorist group are eliminated when 1.3 times the median is used as the threshold. On the other hand, this group contains only 15 objects so, in a sense, it is still performing well.

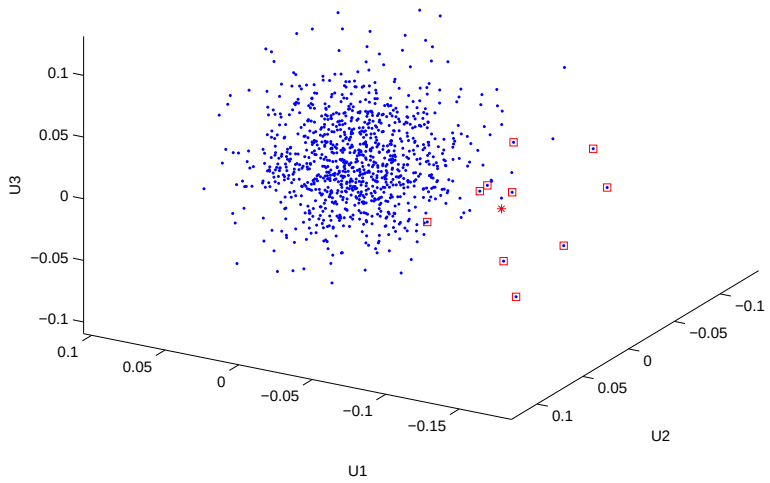

Figure 53: Dataset 8, SVD clustering showing positioning of the terrorist cluster

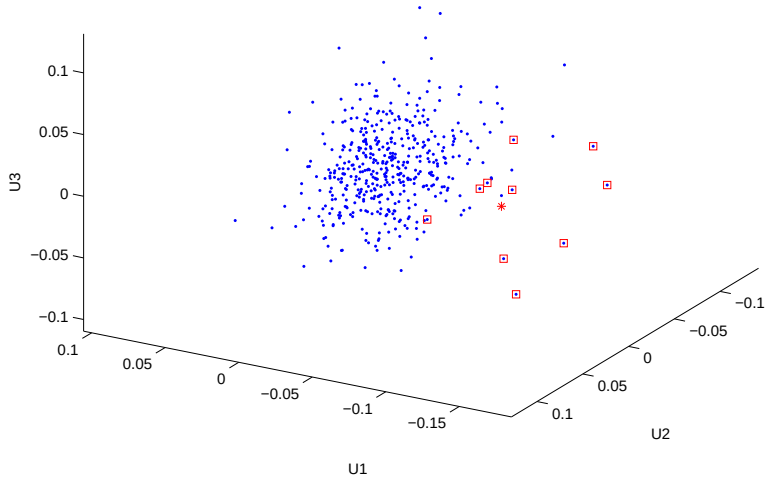

Figure 54: Dataset 8, 465 objects correlated with the target

The JSS methodology is quite weak on this binary dataset. Although the terrorist group is still similar to the terrorist, a large number of other points are also considered just as similar.

6.1 Summary

SVD performs well at separating terrorist clusters from ordinary objects over a wide range of dataset types. However, the results of SVD require human analysis to detect such clusters. Although this can be partly automated, for example by ranking points by their distance from the origin, this

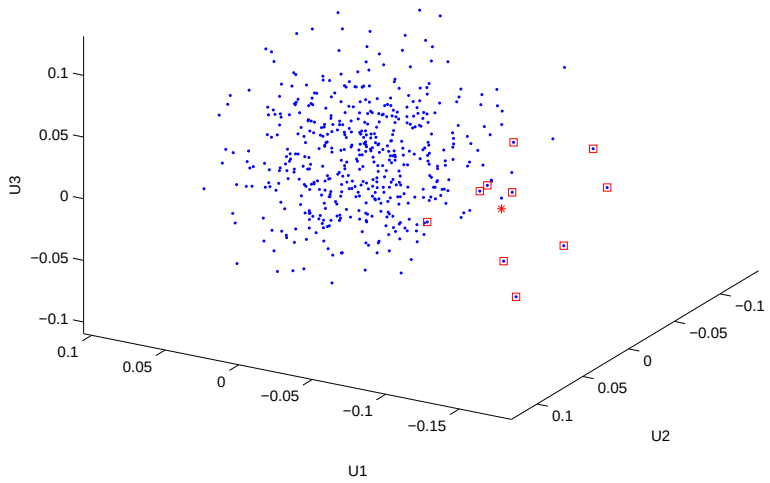

Figure 55: Dataset 8, 504 objects greater than median distance from the origin

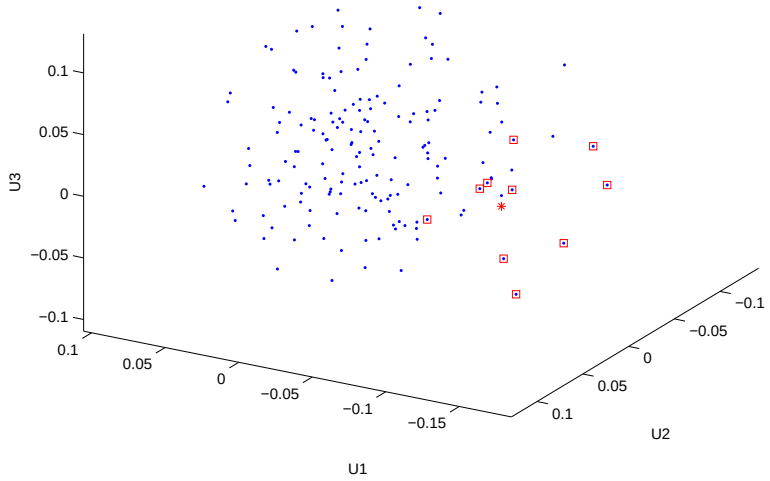

Figure 56: Dataset 8, 171 objects greater than 1.3 times the median distance from the origin

process loses the important directional information. Nevertheless, such a ranking could be used to generate a threat score for downstream analysis.

The combination of SVD and SDD in the JSS methodology is the strongest of the analysis techniques. It exploits SVD's ability to detect correlation, but enhances it by using SDD's ability to detect regions of similar value in a correlation matrix. In general, the JSS methodology is better at partitioning objects into groups, and at identifying the group or groups that is most closely related to the target (when this is known).

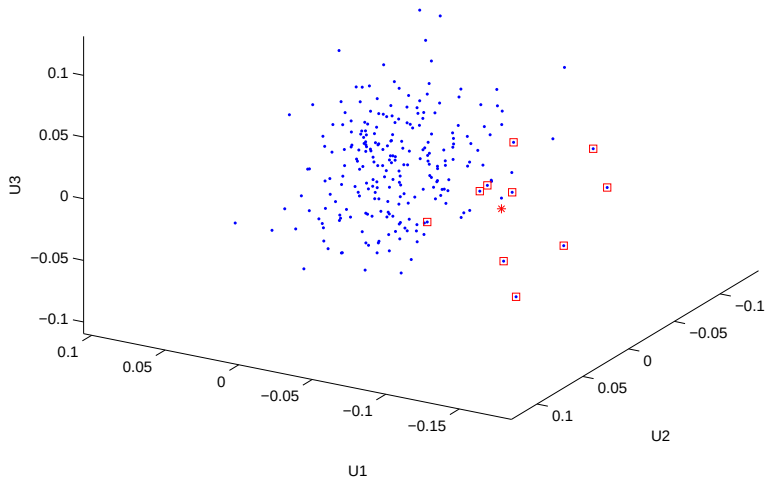

Figure 57: Dataset 8, 247 objects greater than median distance from the origin and correlated with the target

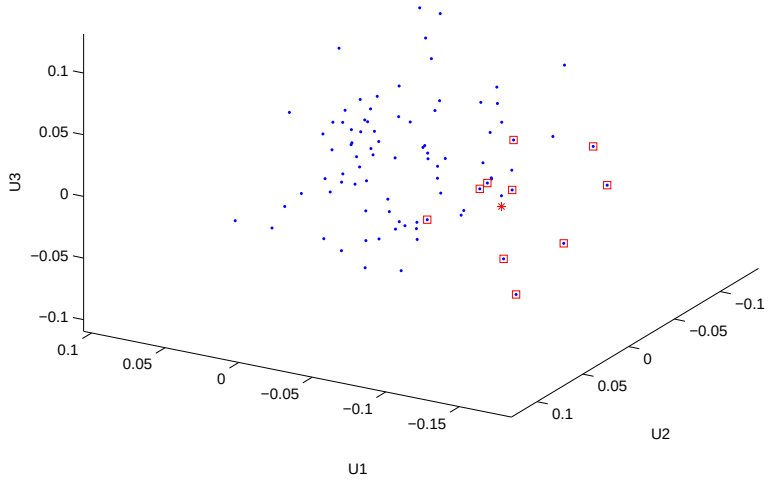

Figure 58: Dataset 8, 89 objects greater than 1.3 times the median distance from the origin and correlated with the target

7 Conclusion

We have shown that two matrix decompositions, SVD and SDD, are able to detect small correlated clusters, representing terrorists, against a variety of backgrounds representing degrees of innocent correlation. In particular, their use in combination using the JSS methodology is able to identify terrorist groups with very few false positives and no false negatives for terrorist groups as a whole.

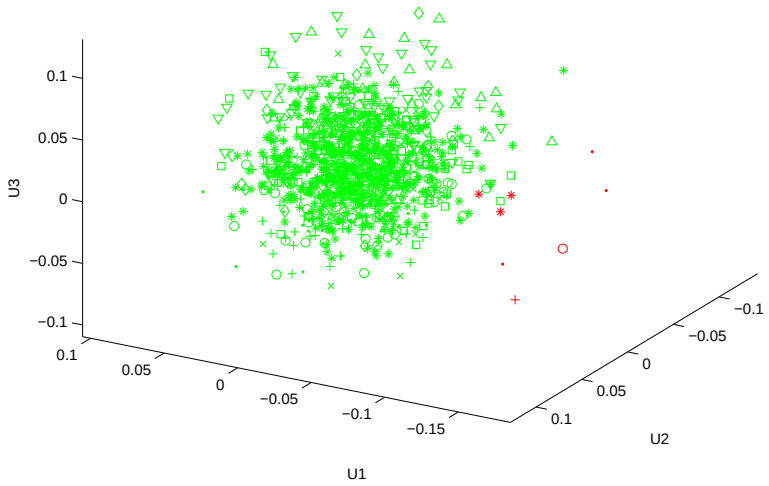

Figure 59: Dataset 8, position from SVD, color and shape from JSS. The terrorist cluster is identified by the JSS hierarchical classification

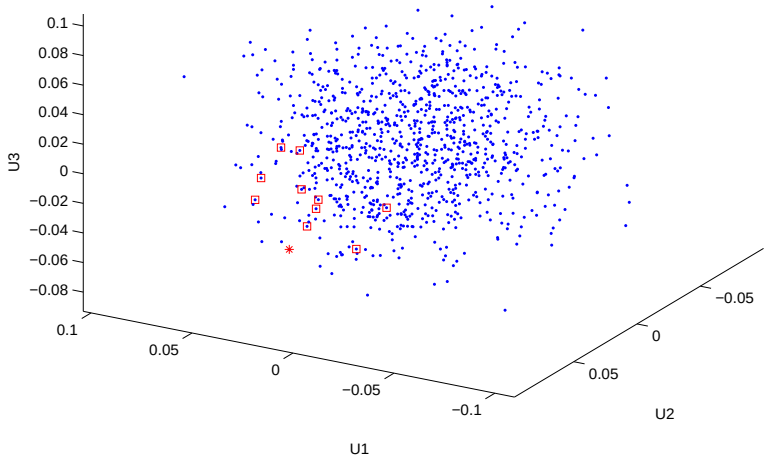

Figure 60: Dataset 9, SVD clustering showing positioning of the terrorist cluster

These techniques represent the front line of data mining for counterterrorism. They are not strong enough to identify terrorists unambiguously, but they reduce the size of the problem for downstream techniques, often reducing the size of the datasets that need to be considered by more than 90%.

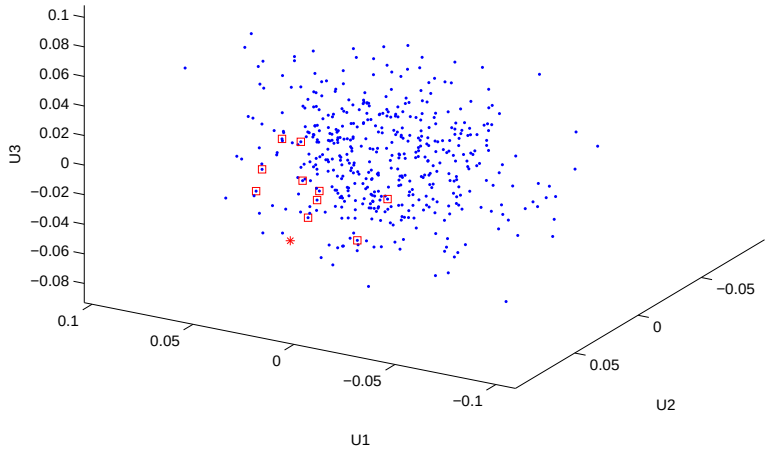

Figure 61: Dataset 9, 478 objects correlated with the target

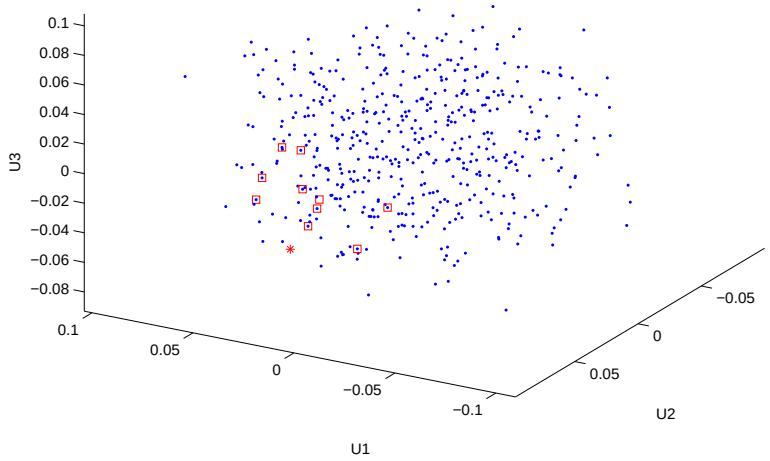

Figure 62: Dataset 9, 504 objects greater than median distance from the origin

References

[1] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2001.

[2] W.E. Baker and R.B. Faulkner. The social organization of conspiracy: Illegal networks in the heavy electrical equipment industry. *American Sociological Review*, 58:837–860, December 1993.

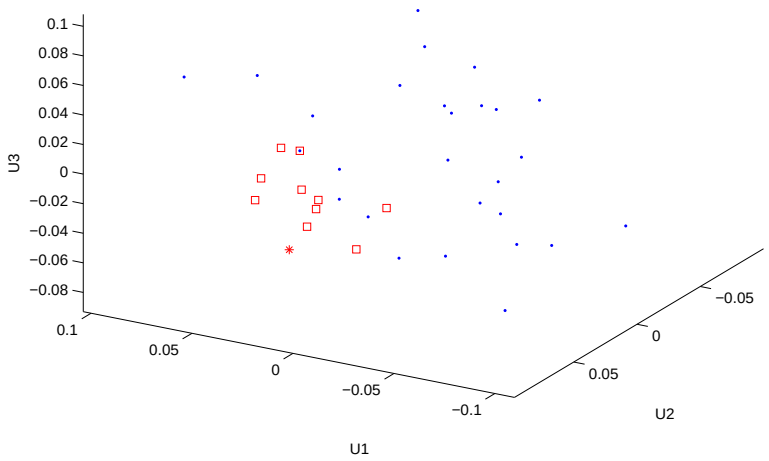

Figure 63: Dataset 9, 27 objects greater than 1.3 times the median distance from the origin. Now only one terrorist is detected

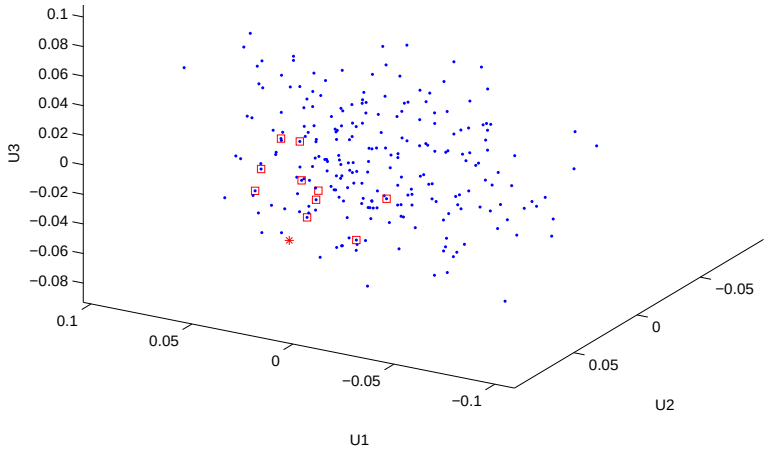

Figure 64: Dataset 9, 240 objects greater than median distance from the origin and correlated with the target

[3] S. Chakrabarti and A. Strauss. Carnival booth: An algorithm for defeating the computer-assisted passenger screening system. Course Paper, MIT 6.806: Law and Ethics on the Electronic Frontier, `http://www.swiss.ai.mit.edu/6805/student-papers/spring02-papers/caps.htm`, 2002.

[4] T. Coffman, S. Greenblatt, and S. Marcus. Graph-based technologies for intelligence analysis.

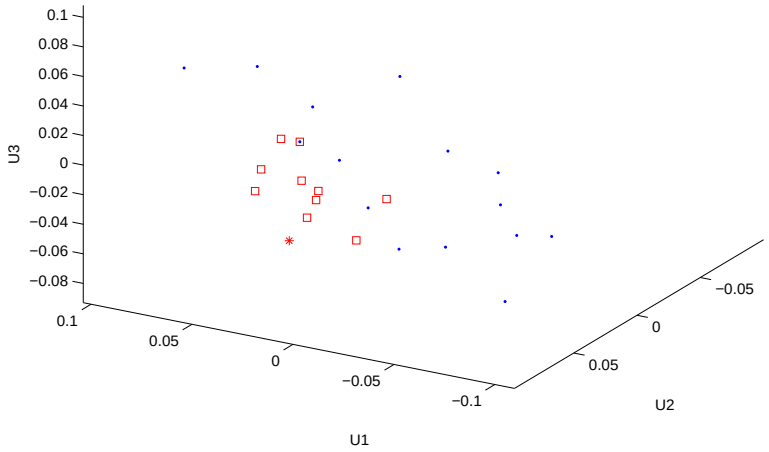

Figure 65: Dataset 9, 15 objects greater than 1.3 times the median distance from the origin and correlated with the target

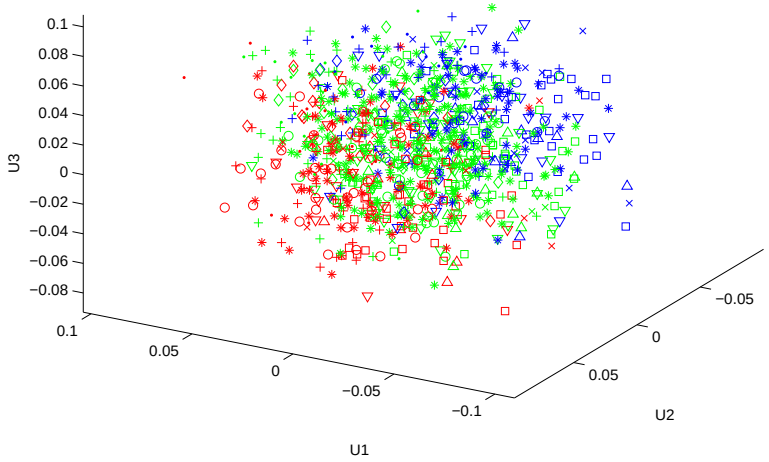

Figure 66: Dataset 9, position from SVD, color and shape from JSS

CACM, 47(3):45–47, March 2004.

[5] European Parliament Temporary Committee on the ECHELON Interception System. Final report on the existence of a global system for the interception of private and commercial communications (echelon interception system), 2001.

[6] L. Garton, C. Haythornthwaite, and B. Wellman. Studying online social networks. *Journal of Computer-Mediated Communication*, 3(1), 1997.

[7] G.H. Golub and C.F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[8] L. Hubert, J. Meulman, and W. Heiser. Two purposes for matrix factorization: A historical appraisal. *SIAM Review*, 42(1):68–82, 2000.

[9] D. Jensen and J. Neville. Data mining in social networks. Invited presentation to the National Academy of Sciences Workshop on Dynamic Social Network Modeling and Analysis, November 2003.

[10] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. In *Proceedings of the 41st Foundations of Computer Science (FOCS '00)*, page 367, 2000.

[11] G. Kolda and D.P. O'Leary. A semi-discrete matrix decomposition for latent semantic indexing in information retrieval. *ACM Transactions on Information Systems*, 16:322–346, 1998.

[12] T.G. Kolda and D.P. O'Leary. Computation and uses of the semidiscrete matrix decomposition. *ACM Transactions on Information Processing*, 1999.

[13] V.E. Krebs. Mapping networks of terrorist cells. *Connections*, 24(3):43–52, 2002.

[14] S. McConnell and D.B. Skillicorn. Semidiscrete decomposition: A bump hunting technique. In *Australasian Data Mining Workshop*, pages 75–82, December 2002.

[15] R.J. Mooney, P. Melville, L.R. Tang, J. Shavlik, I de Castro Dutra, D. Page, and V.S. Costa. Relational data mining with Inductive Logic Programming for link discovery. In *Proceedings of the National Science Foundation Workshop on Next Generation Data Mining*, November 2002.

[16] D.P. O'Leary and S. Peleg. Digital image compression by outer product expansion. *IEEE Transactions on Communications*, 31:441–444, 1983.

[17] J.D. Rhodes. CAPPS II: Red light, green light, or mother, may I?. *Journal of Homeland Security*, March 2004.

[18] D.B. Skillicorn. Detecting related message traffic. In *Workshop on Link Analysis, Security and Counterterrorism, SIAM Data Mining Conference*, pages 39–48, 2004.

[19] K. A. Taipale. Data mining and domestic security: Connecting the dots to make sense of data. *Columbia Science and Technology Law Review*, 2, December 2003.

[20] J.R. Tyler, D.M. Wilkinson, and B.A. Huberman. Email as spectroscopy: Automated discovery of community structure within organizations. HP Labs, 1501 Page Mill Road, Palo Alto CA, 94304, 2003.

[21] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *Proceedings IEEE International Conference on Computer Vision*, pages 975–982, 1999.