Hiding Clusters in Adversarial Settings

J.G. Dutrisac and D.B. Skillicorn
School of Computing, Queen's University
Kingston, Canada

***Abstract*—In adversarial settings, records associated with those who want to conceal their existence or activities tend to be unusual because of their illicit status; but not too unusual because of efforts to make them as normal as possible. Clusters of such records will not be single outliers or even outlying clusters, but rather small clusters on the fringes of normal clusters. Such structures are undetectable by many mainstream clustering algorithms, for example those based on distance and convexity. We show that even more sophisticated clustering algorithms are easily subverted by the addition of only a few carefully chosen records. Robust clustering in adversarial settings will require the development of more sophisticated algorithms tailored to this domain.**

I. Introduction

When clustering algorithms are applied to data in adversarial settings, where there are some who wish to conceal their existence, identities, or activities, it is not obvious what the concealed records might look like. On the one hand, the intentions and actions of such individuals are unusual, so that the records about them should be unusual. But on the other hand, they are (unless stupid or naive) aware that knowledge-discovery techniques will be applied, and so will work hard to make their records look as ordinary as possible. Hence, we do not expect the records associated with 'bad guys' to be outliers in a clustering; but we do not expect them to fit entirely within the clusters that represent normality either.

Having attribute values that are within the normal range, but extremal for several attributes tends to create records that are 'in the corners' of ordinary clusters. As the number of attributes increases, it becomes increasingly unlikely that normal records will have extremal values for many attributes (the reason for the Bonferroni correction for multivariate data). Therefore we expect that the clusters of interest in adversarial settings will tend to be small; close to larger clusters representing normality; and of arbitrary shape, perhaps hugging the boundaries of larger clusters. We immediately conclude that distance-based clustering algorithms (for example, k-means), and clustering algorithms that assume convex clusters (for example, hierarchical clustering using many common metrics) will not detect such clusters.

The contribution of this paper is to show that several mainstream clustering algorithms are also easily subverted. Hence, detecting clusters in adversarial settings is a difficult problem because known algorithms are inherently weak and are susceptible to manipulation.

II. Attack Model

We assume that the activities of an attacker necessarily cause records to be collected, and that it is difficult for the attributes in these records to be manipulated substantially. For example, given CCTV surveillance of a key building entrance, it is possible to wear a disguise, but not to avoid *some* image being captured.

We also assume that an attacker can cause other, chosen, records to be captured. The records needed are primarily outliers which, if detected using some other analysis, are likely to be discarded as errors; and bridging records that are even more typical than the attacker records to be concealed. In both cases, such records do not necessarily put those associated with them at risk, so the cost of generating them is low. We also assume that attackers do not have access to the rest of the data to be clustered, but can collect or infer data of a generally similar kind – we call this the *attacker's dataset*.

We focus on the situation where there is one large cluster, representing normal records, and a smaller cluster representing bad or undesirable records. The goal of subversion is to conceal the existence of the smaller cluster. The approach extends straightforwardly to the case of multiple normal clusters.

III. Subverting Distribution-Based Clustering

Expectation-Maximization (EM) [1] is a clustering algorithm that assumes that the attribute values of records belonging to a cluster are distributed around mean values that are the underlying reason for the cluster's existence. The distribution is usually assumed to be Gaussian. Given a set of records, and a set of k distributions to be fitted to them, there are two different unknowns: the parameters of the distributions (means and variances), and the allocations of records to clusters. The EM algorithm solves for these two unknowns alternately so as to maximize the likelihood of the clustering.

Two clusters, a normal one and a 'bad guy' cluster, can be made to look like a single one by adding a group of records (a mask) to the dataset so that either the records from both clusters and the mask appear to be a single cluster; or the two original clusters appear to be one cluster, and the mask another.

Hiding one cluster within another requires adding new records that make the overall distribution of attribute values appear more Gaussian. The surprising fact is that the number of records required is not very large. The number of records in the mask should be at least half the number of records in the smaller of the two clusters being covered (but in adversarial settings, one of these clusters will typically be small to begin with). The mean of the distribution for each attribute value for the records in the mask should be the midpoint of the means for the corresponding attribute in the two clusters. The

ISI 2008, June 17-20, 2008, Taipei, Taiwan
1-4244-2415-3/08/$20.00 ©2008 IEEE

standard deviation for each attribute value for the records in the mask should be approximately the difference between the mean attribute value of one cluster and the most extreme value of that attribute from the other cluster. The mask therefore introduces a new multivariate Gaussian overlaying those of the existing clusters, centered between them, and with a much wider spread.

The records in the mask are artificial in the sense that they do not represent normal behavior. In some cases, they may appear to be lone outliers, who may be discounted as eccentrics. In other cases, they may create the impression of a small outlying cluster and so act as a red herring.

The required properties of a mask are determined using the attacker's dataset. Attackers have several advantages: they know labels for the records, since they know what their own actions look like; they know their own strategy for making their records look normal; and they can generate typical 'bad guy' records as required.

The effectiveness of this attack against expectation-maximization will be demonstrated using a single example dataset for lack of space. The dataset has 4500 records and 30 attributes, with 4000 records in one cluster, and 500 in the other. The two clusters differ from one another in the values of the first three attributes. The attacker's dataset contains 450 records, one tenth the size of the actual dataset.

The larger cluster has attribute values Gaussian generated with mean 1 and standard deviation 1. The smaller cluster has Gaussian generated values for the first three attributes with means of -1, -0.5, and -1.5, respectively, and standard deviations of 0.2. All of the other attributes are Gaussian generated with mean 1 and standard deviation 1. This dataset is easily clustered by the EM algorithm.

Using the attacker's dataset, we are able to identify the attributes that best separate the two clusters The mean value for each attribute in the mask is set to the midpoint between the two cluster means for each attribute. As only the first three attributes have means that differ between the clusters, all other attributes of the mask have a mean of 1. The first attribute of the mask has mean $(1-1)/2 = 0$, the second attribute has mean $(1-0.5)/2 = 0.25$, and the third attribute has mean $(1-1.5)/2 = -0.25$. The standard deviation of the mask is approximately 5.5.

Using the parameters determined, a mask is created for use with the attacker's dataset, using the same number of records as in the smaller cluster (50). The resulting distributions are shown in Figure 1, where each attribute appears to have a Gaussian distribution. The EM clustering algorithm on this dataset produces the desired clustering, with the smaller cluster being hidden in the larger cluster and the mask being detected as a new cluster.

IV. Subverting Decomposition-Based Clustering

Matrix decompositions provide ways of clustering data based on particular kinds of structures that each expects to find. We will consider two decompositions that have been used together to discover clusters in large and complex datasets.

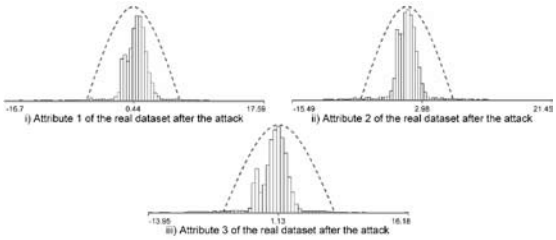

Fig. 1. Data attribute distributions after attack

The first is the Singular Value Decomposition (SVD) [2, 5]. Given an $n \times m$ matrix A, representing the records of a dataset, each row can naturally be interpreted as a point in m-dimensional space. However, m is typically quite large, so there is no straightforward way to visualize the records and their relationships using this geometry. The singular value decomposition of A is

$$A = USV'$$

where U is an $n \times m$ orthogonal matrix, V is an $m \times m$ orthogonal matrix (the dash indicates transposition), and S is a diagonal matrix with non-increasing diagonal entries called the singular values). An SVD implements a change of basis (the new basis captured by V) in which the rows of U correspond to the coordinates of the point associated with each record in the transformed space. However, its most useful property is that the first axis in the transformed space points in the direction of maximum variation in the data, the second axis in the direction of next greatest variation and so on. The SVD can be truncated at some number of dimensions, say k, thus: $A \approx U_k S_k V'_k$, where U is the $n \times k$ matrix of the first k columns of U, and S_k is the $k \times k$ upper triangle of S, and so on. This truncation is the most faithful representation of A in k dimensions. If k is chosen to be 2 or 3, the rows of U_k can be plotted, confident that any structure visible is genuinely present in the dataset, although some structures may not be representable. Clusters can often be detected in this visualization.

The semidiscrete decomposition (SDD) [3, 4] is a similar decomposition

$$A = XDY$$

where X is an $n \times k$ matrix, Y is a $k \times m$ matrix, the entries of X and Y are from the set $\{-1, 0, +1\}$, and D is a diagonal matrix. SDD expresses the clusters in the dataset in terms of rectilinearly-aligned regions of similar value in the data matrix. These are captured in terms of stencils expressed by the product of a column of X and a row of Y, with an average height given by the corresponding entry in D. However, for our purposes we need only observe that an SDD provides a ternary hierarchical clustering of the records, given by the entries in the first, and subsequent, columns of X. At the top level the records are divided into three clusters, those with a $+1$, -1, and 0 respectively in column 1 of the X matrix. Each of these clusters can be further subdivided according to the entries in column 2 of the X matrix, and so on. SDD produces the most accurate results when applied to the correlation matrix (AA') of the dataset, and so we use it in this way.

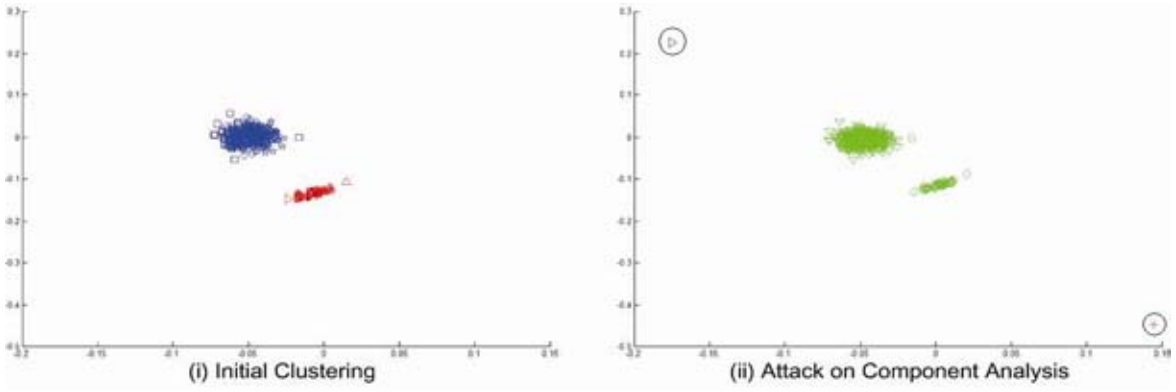

Fig. 2. Subverting component analysis by adding two outlying clusters (circled)

As the SVD and SDD have different but complementary roles, clustering by decomposition may be considered a two-part process. It is necessary to consider both these parts when attempting to subvert decomposition-based clustering.

Clustering by decomposition relies heavily on the distribution of records in a dataset. Our method of attack again requires adding records to a dataset so that a smaller cluster is hidden by making it seems as if it is part of a larger cluster. The attack against decomposition-based clustering is easier than the EM attack. Only a small number of records need to have extreme values.

The attack against decomposition-based clustering focuses on the labelled clustering generated from the SDD. The basic strategy is to add records to the dataset that SDD will interpret as extreme clusters, leaving the original two clusters seeming like a single cluster. An example of this is shown in Figure 2.

Graph (*i*) of Figure 2 shows the clustering of a dataset prior to the attack. In this graph, the smaller cluster is red while the larger cluster is blue. The records of one cluster are labelled $+1$ in column 1 of the X matrix, and those of the other cluster are labelled -1, so it is obvious from the decomposition that there are two clusters. Two new clusters, actually only two new records, are added; and when the decomposition is applied to the modified dataset, the result is as shown in graph (*ii*) of the figure. Each of the two added records is detected as a cluster of size 1 (the blue triangle and the red cross), while all of the original records are considered to belong to a single cluster. (Of course, column 2 of the X matrix will separate the central cluster into its two original pieces, but the same strategy can be applied with two closer records to create new structure at the second level, and so on.) The attribute values of the records in the added clusters are determined by the relationship between the two original clusters. The idea is to place the added clusters far from the original clusters along the line joining them.

It is still easy to detect, by human inspection, that the central cluster is really two clusters, although only after applying an SVD. Since visual analytics is an alternate methodology, a bridge of extra records that lie in the gap between the existing clusters can be inserted to make the separation less obvious. This is illustrated in Figure 3, where we have inserted 20% of the number of records in the smaller cluster between the two original clusters. The mean of each attribute for records in this bridge is the midpoint of the means of the two original clusters, and the standard deviation is two-thirds of the distance between the two original cluster means. This ensures that the records mix well into both clusters.

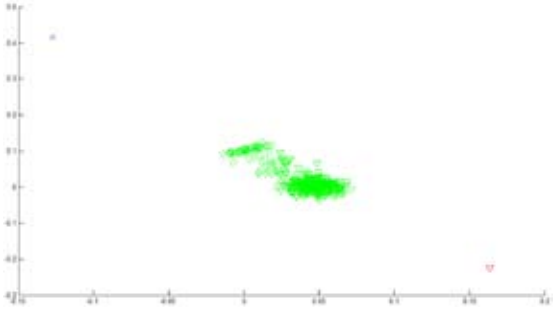

Fig. 3. Bridging the two clusters

V. Conclusion

In most adversarial settings, the records corresponding to 'bad guys' are likely to be fringes on larger clusters of normal records, rather than single outliers or outlying clusters. This has implications for the kinds of clustering algorithms that can be used in such settings. We show that even those algorithms that might be expected to perform well are easily subverted by the additional of only a small number of carefully-chosen records.

This provides further evidence that knowledge discovery in adversarial settings requires the development of new algorithms, rather than adapting mainstream algorithms as if this were simply a different application domain.

References

[1] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:138, 1977.

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

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

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

[5] D.B. Skillicorn. *Understanding Complex Datasets: Data Mining with Matrix Decompositions*. CRC Press, 2007.