Identifying Storms in Noisy Radio Frequency Data via Satellite: an Application of Density Estimation and Cluster Analysis

Tom Burr, Angela Mielke, Abram Jacobson
Mail Stop E541 Los Alamos National Laboratory
Los Alamos, NM 87545

Abstract

The FORTE (Fast On-Orbit Recording of Transient Events) satellite collects records of radio frequency events that exceed a threshold. Here we consider data for the observed phenomena of "recurrent-emission storms." Each data point represents the total electron content (TEC) of the emanation of a 409.6 μ-s collection window. Some data records contain well-defined storm events which consist of many data points in a specialized cluster. We present a method involving noise rejection and cluster analysis to identify well-defined storms from the data records. We first remove noise using density estimation and then apply hierarchical clustering to the higher-density points. For each identified cluster of points, we fit TEC as a quadratic function of time (a quadratic shape is anticipated from atmospheric physics), and find more points that belong to the cluster using a careful extrapolation. The overall performance of finding each storm and identifying which points belong to which storm is assessed by comparing our results to test data produced by a human analyst. We also give results for three other mixture-fitting methods: (1) principal curve clustering, (2) a method using the integrated squared error norm, and (3) a method using the expectation-maximization algorithm.

1 Background and Data Description

The FORTE (Fast On-Orbit Recording of Transient Events) satellite collects records of radio frequency (RF) events that exceed a threshold ((Moore, 1995) and (Jacobson, 1999)). Here we consider data for the observed phenomena of "recurrent-emission storms." This data emanates from a single RF source that radiates while the satellite passes overhead. Each data point of a micro event represents the total electron content (TEC) of the emanation of a 409.6 μs collection window. Each data record contains approximately 100 to 400 micro events and is processed with a dechirping step that corrects for the frequency-dependent transit time through the atmosphere. The total record time is approximately 15 minutes which is the time for the satellite to pass over a region of the earth. A data record in our study contains 100 to 400 micro events ("points"). As defined here, "data records" were produced from a database query for at least 100 events and over a certain region of the earth.

Most data records from our query contain storms which consist of many data points (micro events) in a specialized cluster shaper. Atmospheric dispersion models suggest that a fully viewed storm will consist of data points in a bowl shape. Because most of the bowl is concave up, and some false positives are concave down, we will restrict attention to concave up feature clusters. We have empirically determined that a quadratic fit is adequate to describe most clusters of interest.

We present a method involving noise rejection and cluster analysis to identify well-defined storms from the data records. We first remove noise using density estimation and then apply hierarchical clustering to the higher-density micro events. For each identified cluster of micro events, we fit TEC as a quadratic function of time, and find more micro events that belong to the cluster using a careful extrapolation. Because some data records contain false alarms having many points generated over a very short time, a final inspection of the found clusters includes diagnostic checks of the time duration and of the residual variance of the points around the quadratic shape. We also reject concave down clusters. The overall performance of finding each storm and identifying which micro events belong to which storm is assessed by comparing our results to test data produced by a human analyst.

1.1 Examples

We plot examples using the four data sets D6, D7, D18 and D16 (of 30 analyzed) in Figure 1. Figure 1a is the easiest example in the sense that we expect to find the one cluster, probably without any false positives. There is only one storm in Figure 1a so the number of clusters is $K = 1$. In all cases, we use the following labeling convention. The first cluster (storm) is labeled with a 1, the second with a 2, etc., and the noise points are all labeled with the integer $K + 1$. Figure 1b is somewhat harder because there are two clusters quite close together. Figure 1c is harder still, and we might expect to have several false positives and/or false negatives in Figure 1d which is the hardest.

1.2 Performance Measures

Our goal is to find quadratic-shaped upward clusters and to do so with a small false positive rate. We can accept a relatively large false negative rate because we expect thousands of event records from which we need to extract well-characterized storms. The data from identified storms (clusters) will be linked to ground-based data for further analysis as described in Jacobson et. al. (1999) and to be extended in future work.

1.3 Our Approach

After some initial trial and error with several methods, we chose an iterative procedure that we call method 1 with the following steps.

1. Reject noise points (all rejected points are candidates for inclusion with a cluster later).

2. Get cluster result A with optimal values, and result B with near optimal values.

3. For each cluster, extrapolate using a quadratic fit and a "zone of ownership" to avoid ambiguous points. Each cluster center defines a zone of ownership using uncertainty bands that widen as the extrapolation distance increases. If a point "belongs" to two or more zones of ownership, then it is ambiguous and assigned to the noise class.

4. Compare A and B results. For each cluster in A that is confirmed in B, accept the cluster as a storm. We also apply diagnostic checks to reject found clusters with very short time duration or a concave down shape or a very large residual variance around the quadratic shape. For example, D10 contains many points over a very short time range that appear to any algorithm to be a cluster but are the result of calibration or data corruption.

Figure 2 illustrates method 1 without the initial noise removal for data sets D1, D3 and D8. Note that we do not find the 2 clusters in D1. Figure 3 illustrates method 1 with the initial noise removal for the same three data sets and note that we do find the 2 clusters in D1.

We now describe steps 1-4 in more detail.

1. Noise rejection. Do an initial noise rejection (all points removed are candidates to be added back in step 3) using the distance to the nearest $k_1, k_2, \ldots, k_5$ neighbors. We also tested a density estimation scheme that counted the number of neighbors within distances $r_1, r_2, \ldots, r_5$ of each point. Several composite measures for identifying noise were then implemented and tested. For example, we used the first 2 principal components (PCs) of the 10-dimensional data (5 k-nearest neighbor results and 5 density estimation results). However, the most basic k-nearest neighbor method with $k = 3$ to 5 worked nearly as well as more involved methods. For example, using $k = 5$, the false positive rates ranged from 0.03 to 0.33 for a range of thresholds while the associated false negative rates ranged from 0.79 to 0.36. A method based on the PCs of the 10-dimensional data had false positive rates ranging from 0.02 to 0.32 with associated false negative rates of 0.79 to 0.35. Because these rates are essentially the same, we use the distance to the 5th nearest neighbor to identify noise. In all cases, it was better to scale TEC and time to unit variance so we assume TEC and time have been rescaled in the remainder of our discussion.

2. Cluster results A and B. We used hierarchical clustering with single linkage (the distance between clusters is the minimum distance between a point in the first cluster and a point in the second cluster). This is called method = "connected" in Splus, and it favors long and thin clusters. We cut the hierarchical tree at a high percentile (approximately the 0.99 percentile) of all between-group distances to select the number of clusters. Six parameters are involved: 2 noise rejection thresholds (relative to the scene f_1 and absolute parameter f_2), 3 factors related to hierarchical clustering: cut the hierarchical tree at some high percentile f_3, reject clusters with small relative number of observations f_4, reject clusters with small relative number of observations f_5, and 1 factor f_6, related to extrapolation from original cluster. The factor f_6 is the fraction of the range of the original cluster to allow in extrapolation to identify new feature points. We identified good nominal values of these 6 parameters using data set D1. D1 is a good training example because it is moderately difficult, having 2 distinct but somewhat close clusters. We then used a low value of half the nominal and a high value of twice the nominal and searched over 3^6 runs using all 30 data sets to find good values. Good values had the lowest false negative rates subject to having a very low false positive rate. The result is that the optimal values over 30 data sets are approximately the same as those chosen from D1. Therefore, we use the values selected from exploratory analysis with D1 as the "result A" values and values that differ slightly from these are the "result B" values.

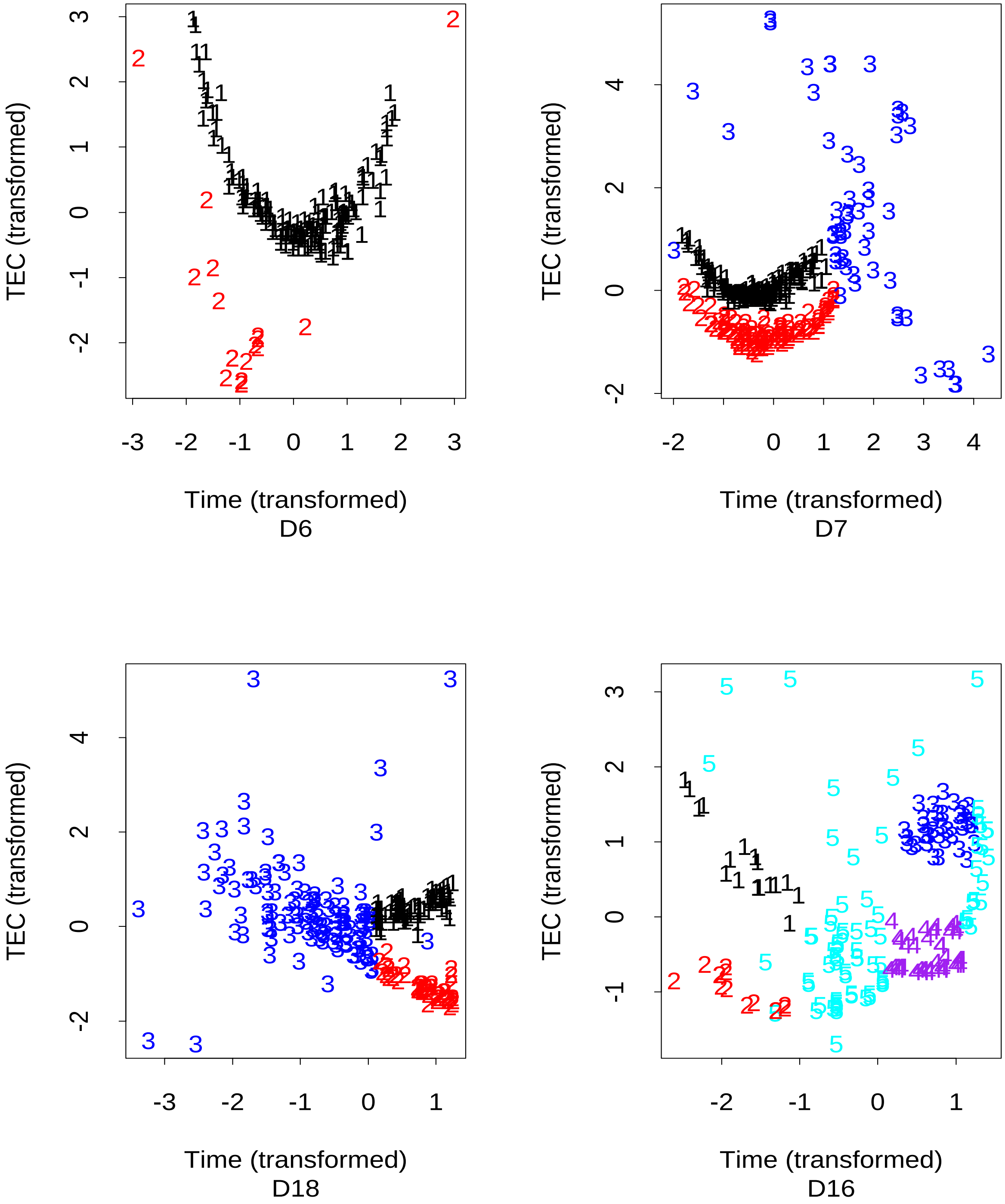

Figure 1. Examples in order of increasing difficulty, found in data sets D6, D7, D18, and D16.

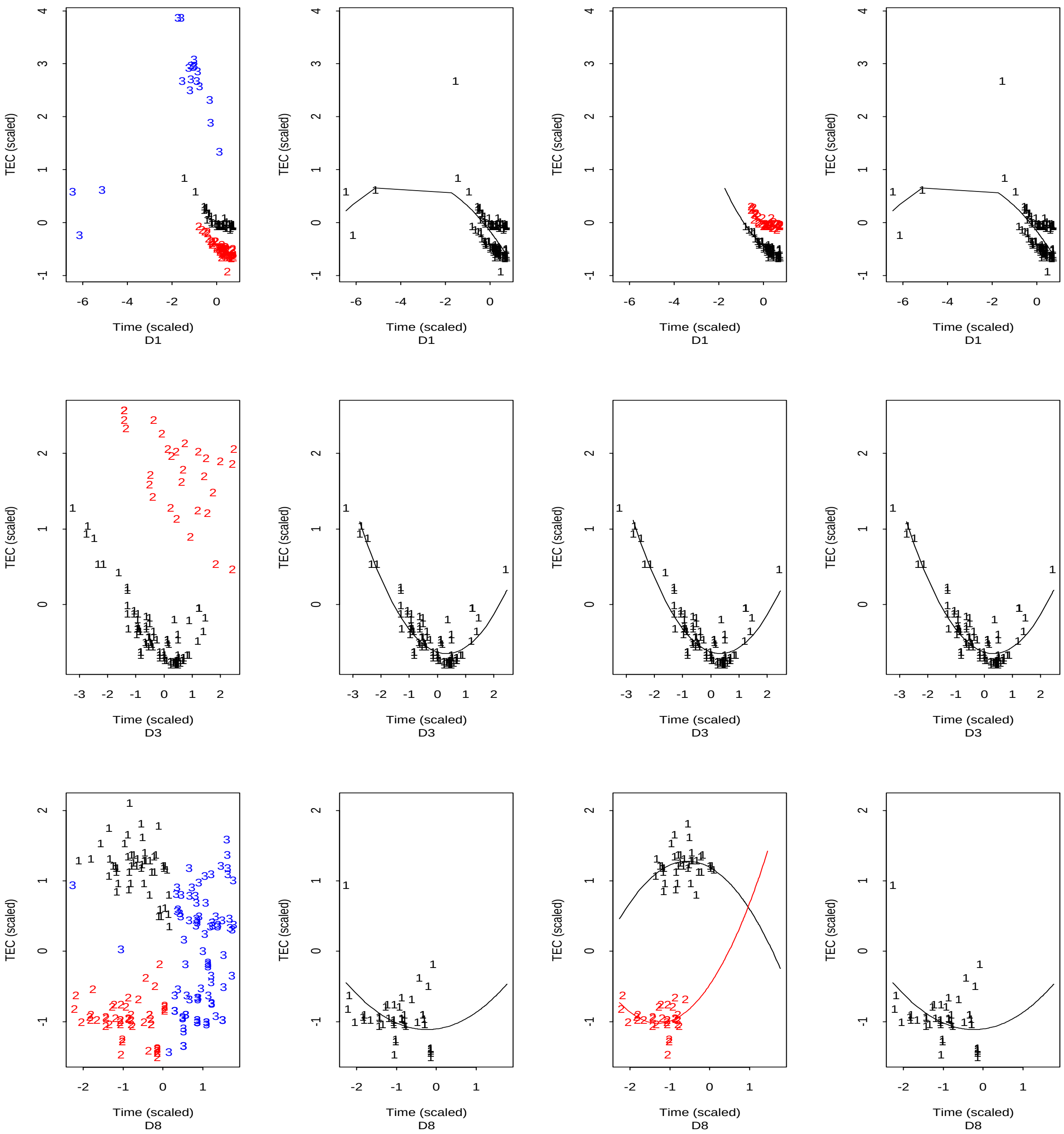

Figure 2. Example results using Method 1 without noise removal. The correctly labeled data is in column 1. The results using parameters A are in column 2. The results using parameters B are in column 3, and the final result is in column 4.

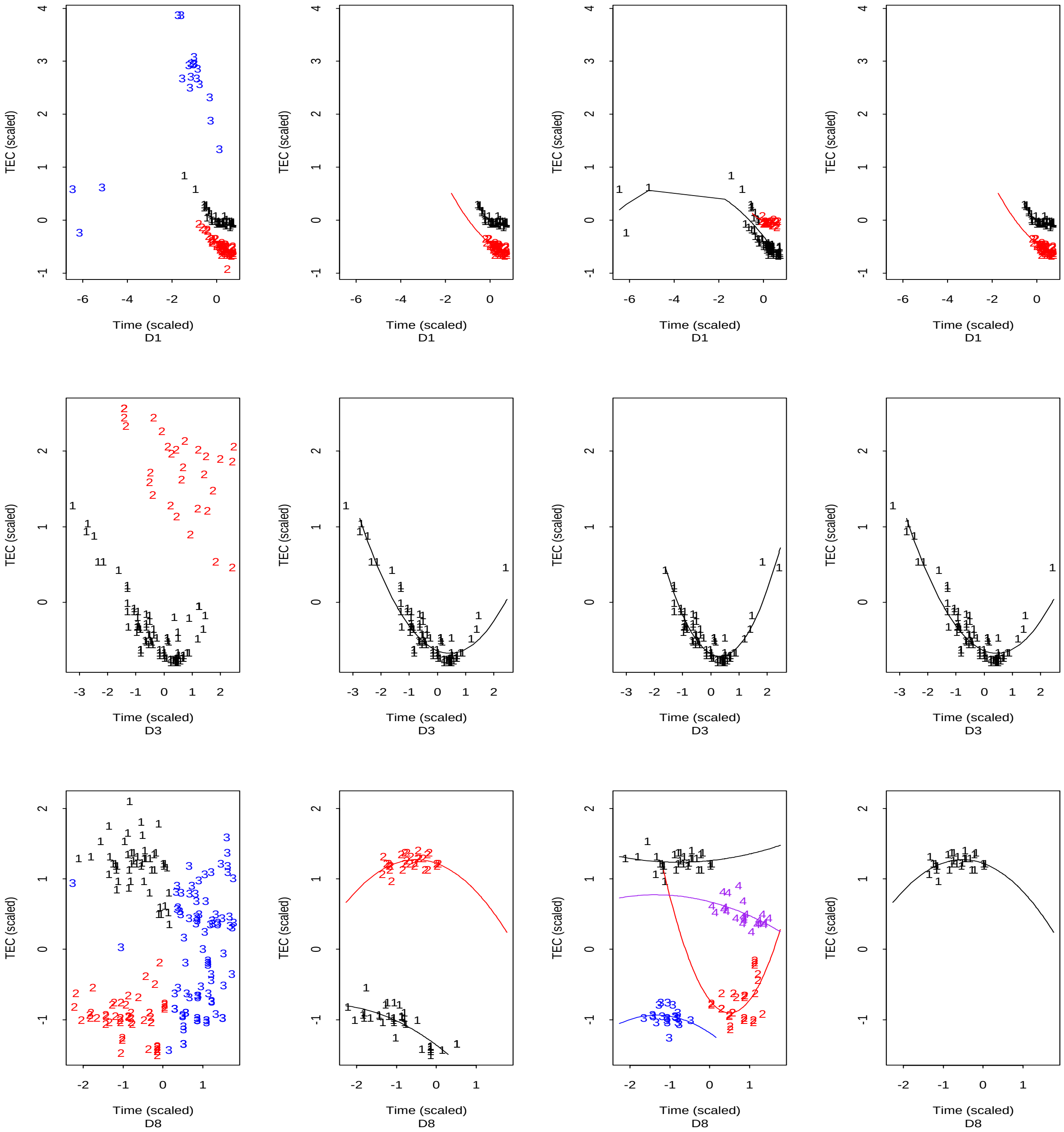

Figure 3. Same as Figure 2, except with noise removal.

3. Extrapolate outward from each cluster to identify new points. Some of the clusters have gaps so we anticipated the need to extrapolate outward from each central cluster in order to identify points that were clearly part of the storm. To do so, we fit TEC (scaled) as a quadratic function of time (scaled), and used the least squares estimates of uncertainty in the fitted function (the variance/covariance matrix of the fitted intercept, linear, and quadratic terms) to choose the "zone of ownership" of a cluster. If two or more clusters appeared to "own" a point then the point was ambiguous and labeled as a noise point. To control the false positive (FP) rate, we did not allow extrapolation to extend too far as a function of the range of the original core cluster. Otherwise, small, vague clusters would "own" too many noise points leading to a high FP rate.

4. Cluster stability check. We used two clustering results (A and B) to check for cluster stability and provide a final control on the FP rate. If and only if a cluster was found in both A and B and passed the three diagnostic checks (time duration, direction of curvature, and residual variance) would the final result include that cluster.

2 Results

We queried archived FORTE records and manually created 30 training cases, each having 0 to 5 clusters. Most cases had 1 or 2 clusters, but a few had 3, 4, or 5, and one case had 0 clusters.

We must define when a cluster has been found, which is somewhat subjective. Also, for each found cluster we define the false positive and false negative rates (fpr and fnr) as the number of false positive points divided by the true number of points in the class and the number of false negatives to be the number of missed points divided by the true number of points in the class.

Concerning whether a cluster has been found, consider the simulated data case in Figure 4. This simulated example is very challenging because of the nature of the overlap of the two clusters. The results from each of four methods (the other three methods are described in Section 3) are shown in Figure 5.

This simulated example raises two questions related to performance measures:

(A) Should we define the effective number of clusters by using some notion of cluster separation?

(B) Should we consider our performance with method 1 in the simulated example to be one false positive and two false negatives?

Because the guessed cluster using method 1 is a hybrid of the two clusters, in our application, a better performance would be to find no clusters in this example. That is, large contamination of a cluster with members of other clusters is undesired, so we prefer to strongly penalize this performance. Our current performance measure works as follows. For the first true cluster (we order by true cluster size, so the first is the largest true cluster) define the guessed cluster by "majority rule," which means that the most common guess is defined to be "the guess." For the second largest true cluster, again define the guessed cluster by "majority rule," but only using previously unassigned cluster labels. Continue for all true clusters. This defines the number of false negatives. If the "majority rule" implies that the guess is the noise cluster, then accept the second-to-largest frequency class as the guessed class. This is a somewhat liberal choice. Reverse the roles of "guess" and "true" to define the number of false positives. Therefore, in Figure 5, method 1 and the L2E method both resulted in one false positive and one false negative with a large false positive rate for the one found cluster. The mixreg and princurve methods each have 2 false negatives.

To define a measure of difficulty for each case, we use the ratio of the within-to-between feature distances. We have experimented with using the median, mean, and minimum ratio for each pair of features and found that the median ratio is as good as any other choice and sometimes is much better. Therefore, currently the median ratio of the within-to-between feature distances is our measure of difficulty. We do not yet try to define a measure of difficulty for a case with no features.

The results on the 30 real data sets for method 1 using the false positive and negative definitions described above are:

false positives: 0 (found 24 of 59); fpr: 0.16, fnr:0.15; false negatives: 35/59.

The false negatives tended to appear for the more difficult cases as defined by the median ratio of the within-to-between class distances. We also evaluated three other methods, each described in the next section.

3 Other Approaches

Here we describe three other approaches and give results from each. These three methods treat this as a mixture fitting problem but the details differ among the three methods. Mixture fitting is known

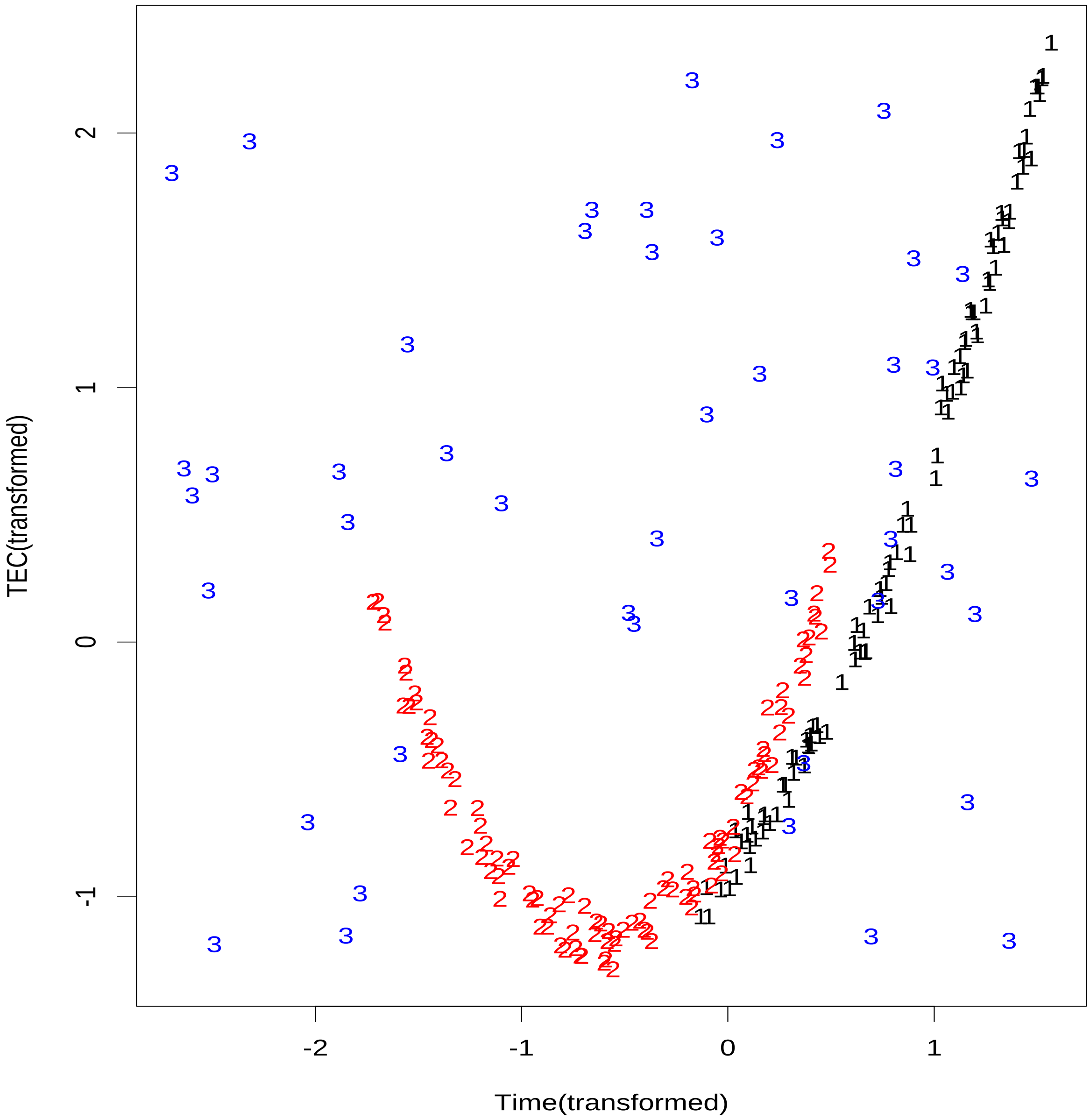

Figure 4. Simulated example with overlapping features

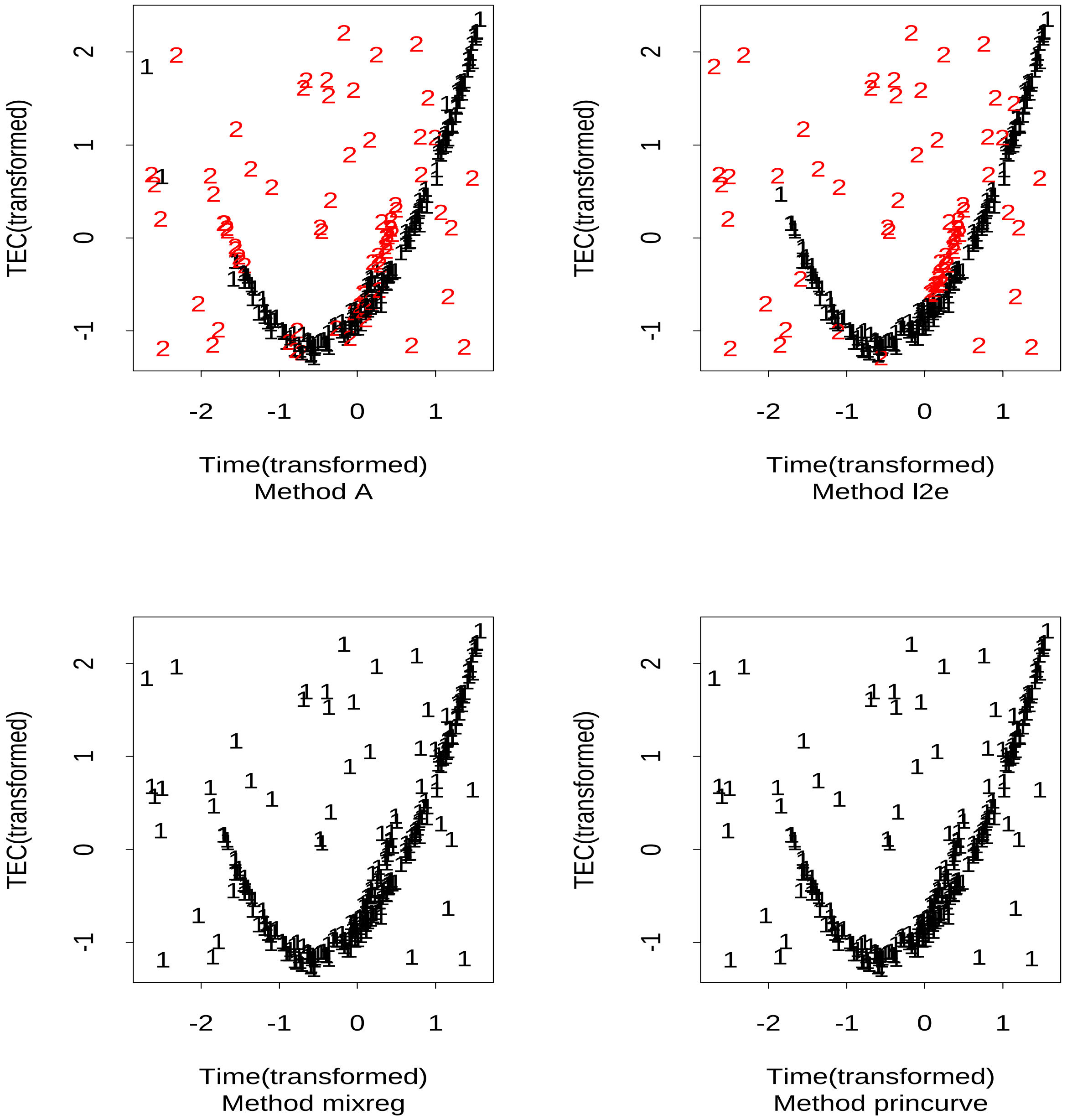

Figure 5. Results with each of the 4 methods for the simulated data in Figure 4.

to be difficult. In all cases the mixture model for the cluster classes assumes that the errors around the quadratic fit are Gaussian. The stochastic component of each model for each observation x is therefore some version of $\sum_{k=1}^{K} \pi_k \phi(x|\mu_k \sigma_k^2)$, where π_k are the relative fractions in each cluster, μ_k is the mean of cluster k and σ_k^2 is the variance of cluster k, and ϕ is the Gaussian probability density. We use the notation of the cited references for each method below, so both the approach and the notation differ among the three methods.

Principal curve clustering

Principal curve clustering with noise was developed by Stanford and Raftery (2000) to locate principal curves in noisy spatial point process data. Although our data is time series, the technique is a viable candidate when we view the predictor as time and the response as TEC. A principal curve is a smooth curvilinear summary of p-dimensional data (Hastie and Stuetzle, 1989). It is a nonlinear generalization of the first principal component line observations. Stanford and Raftery developed an algorithm that first uses hierarchical principal curve clustering (HPCC, which is a hierarchical and agglomerative clustering method) and next uses an iterative relocation (reassign points to new clusters) based on the classification estimation-maximization (CEM) algorithm. A probability model uses the principal curve probability model for the feature clusters and a homogeneous Poisson process model for the noise cluster. Because our features are approximately quadratic in shape, a principal curve with 2 to 5 degrees of freedom should be an adequate fit. Future work will consider using a probability model that uses a quadratic fit using HPCC, although we do not anticipate that this will have much impact.

The noise in most cases appears to be a non homogeneous Poisson process with a strong tendency for the noise cluster to appear in one or a few regions of the scene. Therefore, we modified the noise model by assuming Poisson-distributed noise only within the range of the observed noise. Because the procedure must identify noise, this implies that we should adaptively choose the noise regions of the scene, and also perhaps the noise model. At present, we use a noise model from the initially identified noise points.

Let X denote the set of observations, $x_1, x_2, \ldots, x_n$ and C be a partition considering of clusters $C_1, C_2 \ldots, C_{K+1}$ where the cluster C_j contains n_j points. The noise cluster is C_{K+1} and assume feature points are distributed uniformly along the true underlying feature so their projections onto the feature's principal curve are randomly drawn from a uniform $U(0, \nu_j)$ distribution, where ν_j is the length of the jth curve. An approximation to the probability for $0, 1, \ldots, K$ clusters is available from the Bayesian Information Criterion (BIC), which is defined as $BIC = 2\log(L(X|\theta) - M\log(n)$, where L is the likelihood of the data X, and M is the number of fitted parameters, so $M = K(DF + 2) + K + 1$ For each of K features we fit 2 parameters (ν_j and σ_j defined below) and a curve having DF degrees of freedom. There are K mixing proportions (π_j defined below) and the estimate of scene area is used to estimate the noise density. The likelihood L satisfies $L(X|\theta) = \prod_{i=1}^{n} L(x_i|\theta)$, where $L(x_i|\theta) = \sum_{j=0}^{K} \pi_i L(x_i|\theta, x_i \in C_j)$ is the mixture likelihood (π_j is the probability that point i belongs to feature j), $L(x_i|\theta, x_i \in C_j) = (1/\nu_j)(1/\sqrt{2\pi}\sigma_j)\exp(-\frac{||x_i - f(\lambda_{ij})||^2}{2\sigma_j^2})$ for the feature clusters, and $||x_i - f(\lambda_{ij})||$ is the Euclidean distance from x_i to its projection point $f(\lambda_{ij})$ on curve j, and $L(x_i|\theta, x_i \in C_j) = 1/Area$ for the noise cluster.

Briefly, the HPCC-CEM steps are:

(1) Make initial estimate of noise points and remove then;

(2) Form an initial clustering with at least seven points in each cluster (we use clara for fast clustering in R);

(3) Fit a principal curve to each cluster;

(4) Calculate a clustering criterion $V = V_{About} + \alpha V_{Along}$, where $V_{About} = \sum_{j=1}^{n} ||x_i - f(\lambda_{ij})||^2$ and $V_{Along} = 1/2 \sum_{j=1}^{n} ||\epsilon_j - \overline{\epsilon}||^2$ and $\epsilon_j = f(\lambda_j) - f(\lambda_{j+1})$, so that V_{About} measures the orthogonal distances to the curve ("residual error sum of squares") and V_{Along} measures the variance in arc length distances between projection points on the curve. Minimizing V (the sum is over all clusters) will lead to clusters with regularly spaced points along the curve, and tightly grouped around it. Large values of α will cause the method to avoid clusters with gaps and small values of α favor thinner clusters. Clustering (merging clusters) continues until V stops decreasing.

Because we believe that data set D1 is a good training set, we evaluated the BIC for D1 for DF ranging from 2 to 6, a range of candidate numbers of clusters, and a range for the parameter α (around the nominal value of 0.4). Conventionally, differences of 2-6 between BIC values represent positive evidence for the model having larger BIC. The goal was to find the α and DF values that caused the BIC to maximize at the correct number of clusters. Unfortunately, with D1 the "maximize BIC" algorithm always chose 3 or 4 clusters for any values of α and DF. Therefore, we altered the probability model for the noise cluster slightly to reflect the fact that the noise tended to locate near the features rather than be randomly scattered throughout the scene. We did this by adjusting the area for the noise cluster to agree closely with the observed area of the noise. The results remained the same which indicates either that the BIC approximation is not particularly effective or that the groups have been inconsistently defined by the human expert. In a similar experiment, we calculated the BIC for the correctly partitioned data and for an estimate of the partition. BIC was maximum for the correct labels in only 10 of the 30 cases with the scene-wide Poisson noise model and in only 11 of the 30 cases with the local Poisson noise model. It might be possible to use the BIC to define a degree of difficulty of each case. For example, if the BIC for the true labels is very close to the BIC for a fairly major rearrangement of the labels, then the case would be considered difficult. Recall, we currently define the measure of difficulty as the median ratio of ratio of the within-to-between feature distances.

The results for the HPCC-CEM method are:

false positives: 4; fpr: 0.11, fnr: 0.08; false negatives: 34/59.

We also modified the results of the HPCC-CEM method by including our diagnostics (direction of curvature, residual variance for each cluster, and duration of each cluster) and the strategy of choosing two sets of slightly different parameters (each set nearly optimal for D1) to get two clustering results, and then accepting only those clusters that were found by both clustering results.

The results for this modified HPCC-CEM method are

false positives: 1; fpr: 0.02, fnr: 0.04; false negatives: 50/59.

We note that this modification results in a very large false negative rate: We detect only 9 of 59 clusters and find one false positive. Although the "per found cluster" false positive and negative rates are impressively low, it appears that this modification is too cautious in conjunction with the principal curve clustering method.

Minimum Integrated Square Error (ISE) Method

The minimum integrated squared error (or L2 distance) appears to be a good approach to fit mixture models, including mixtures of regression models. (Scott, 2002 and Scott and Szewczyk, 2002). The minimum L2 distance method tries to find the largest portion of the data that matches the model. In this case, we seek feature 1 having the most points, regard the remaining points as noise, remove the feature and then repeat the procedure in search of the feature 2, and so on until a stop criterion is reached. It should also be possible to estimate the number of components in the mixture in the first evaluation of the data but we will not consider that approach here. To motivate the form of the L2 estimator, consider estimating the smoothing parameter h in density estimation with $\hat{h} = \arg\min \hat{\int} [f_h(x) - f(x)]^2 dx$. By interpreting terms and ignoring which does not depend on h, we find that $\hat{h}$ should satisfy $\hat{h} = \arg\min_h [\hat{\int} \hat{f}_h^2(x)dx - 2\hat{\int} \hat{f}_h(x) f(x) dx]$. The key quantity to estimate is the second term, which is the expected height of the density estimate (the first integral can be evaluated exactly for any h so it does not require estimation). It follows that a reasonable estimator is $\hat{h} = \arg\min_h [\frac{2}{(n-1)h} - \frac{n+1}{n^2(n-1)h} \sum_k \nu_k^2]$, where ν_k is the bin count of bin $B_k = (x_0 + kh, x_0 + (k+1)h)$ and x_0 is the bin origin. Methods to estimate h arose from nonparametric density estimation but Scott (2002) has shown that in the parametric setting with model $f(x|\theta)$, instead of estimating h we estimate θ using $\arg\min_\theta \hat{\int} [f_h(x|\theta) - f(x|\theta_0)]^2 dx$ where the true parameter θ_0 is unknown. Again the key quantity to estimate is the expected height of the density, $\int f(x|\theta) f(x|\theta_0) dx$ and again by interpreting terms it follows that a reasonable estimator minimizing the parametric ISE criterion is $\hat{\theta}_{L2E} = \arg\min_\theta [\int f(x|\theta)^2 dx - 2/n \sum_{i=1}^n f(x_i|\theta)]$.

This assumes that the correct parametric family is used. To achieve robustness the concept can be extended to include the case in which the assumed parametric form is incorrect. If we focus on the distribution of the residuals in regression then the L2E criterion can fit mixtures of regression models which is a good model for our data. Our data is a mixture of 0 to 5 features plus feature noise and scene noise. Each feature consists approximately of a quadratic feature for which regression of TEC on time and time^2 should provide a good fit. David Scott kindly provided Splus code for fitting a concave up quadratic using the L2E method. The results of this method on data set D1 and D3 are shown in figures 6 and 7 respectively. The top left plot is the original data D1 (transformed). The top right plot is the first feature found ("feature 1"). All points in feature 1 are removed and the procedure is repeated. The middle left plot is the data with feature 1 removed. The middle right plot is the next feature found on the remaining data ("feature 2"). The bottom left plot is the remaining data with both features removed. We chose stop criteria (involving the number of points in each feature and residual sum of squares in each feature) that correctly stopped finding features after these 2 features. The bottom right plot indicates the final result which is qualitatively in good agreement with the human analyst's labels (both features were found). However, not all feature points agreed with those of the human analyst. The results of this same procedure for data set D3 (figure 7) indicate that feature 1 was judged to be 2 features. Therefore, the qualitative performance is not as good with data set D3. Overall, using stop criteria that were optimized for D1, we found 33 of 59 clusters with 26 false positives.

The results for the L2E method are:

false positives: 26; fpr: 0.33, fnr: 0.17; false negatives: 26/59.

Because of this high false positive rate, we also implemented the L2E method by insisting on finding at most 2 or at most 1 cluster.

The results for L2E with "at most 2" clusters imposed are:

false positives: 17; fpr: 0.35, fnr: 0.16; false negatives: 26/59.

The results for L2E with "at most 1" cluster imposed are:

false positives: 1; fpr: 0.34, fnr: 0.16; false negatives: 32/59.

We also modified our results for L2E with "at most 1" cluster imposed, using the same diagnostics and "comparison of two cluster results" as in the other two methods. The modified results are:

false positives: 0; fpr: 0.26, fnr: 0.18; false negatives: 40/59.

Although we found zero false positives, we detect only 19 of 59 clusters. It appears that this modification is too cautious (false negative rate is too high) in conjunction with the L2E method. However, L2E does well at finding the single largest cluster.

Mixreg method

Rolf Turner (2000) implemented a method to handle a mixture of regression models via the well-known EM (estimation - maximization) algorithm (Dempster, Laird, and Rubin, 1977) The function mixreg is available from statlib (http://lib.stat.cmu.edu) for use in Splus (1999).

The likelihood L satisfies $L(X|\theta) = \prod_{i=1}^{n} L(x_i|\theta)$, where $L(x_i|\theta) = \sum_{j=1}^{K} \pi_i L(x_i|\theta, x_i \in C_j)$ is the mixture likelihood (π_j is the probability that point i belongs to feature j) and $L(x_i|\theta, x_i \in C_j) = f_{ij} = (1/\sigma_j)\phi(\frac{y_i - X_i\beta_j}{\sigma_j})$ where ϕ is the standard Gaussian distribution.

Introduce indicator variable z_i of which component of the mixture generated observation y_i and iteratively maximize $Q = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ik} \ln(f_{ik})$ with respect to θ where θ is the complete parameter set of the model consisting of vectors of regression coefficients, the variances σ_i^2, and the mixing probabilities π_j for each class. The γ_{ik} satisfy $\gamma_{ik} = \pi_k f_{ik} / \sum_{i=1}^{K} \pi_k f_{ik}$. Then Q is maximized with respect to θ by weighted regression of y_i on $x_1, \ldots, x_n$ with weights γ_{ik} and each σ_k^2 is given by
$\sigma_k^2 = \sum_{i=1}^{n} \gamma_{ik}(y_i - x_i\beta_k)^2 / \sum_{i=1}^{n} \gamma_{ik}$. In practice the components of θ come from the value of θ in the previous iteration and $\pi_k = 1/n \sum_{i=1}^{n} \gamma_{ik}$.

A difficulty in choosing the number of components in the mixture, each with unknown mixing probability, is that the likelihood ratio statistic has an unknown distribution. Therefore, the mixreg function suite includes a bootstrap strategy (implemented in the function bootcomp) to choose between 1 and 2 components, between 2 and 3, etc. The strategy to choose between K and $K+1$ components is: (a) calculate the log-likelihood ratio statistic Q for a model having K and for a model having $K+1$ components; (b) simulate data from the fitted K component model; (c) fit the K and $K+1$ component models to each simulated data set and calculate the corresponding bootstrap log-likelihood ratio statistic Q^*; (d) compute the p-value for Q as the $p = 1/n \sum_{i=1}^{n} I\{Q \geq Q^*\}$.

The bootcomp function actually implements a semi parametric bootstrap described in Turner (2000). However, bootcomp requires a very long run time and did not lead to an improvement over a slightly unfair strategy involving using the correct data partition in the initial guess for mixreg.

We also include mixreg results using the correct partition with fitted quadratic relations as starting values. Such excellent starting values are not available in practice but allow us to assess how well this version of mixreg could be in the best case.

The mixreg results (with "excellent" starting values) are:

false positives: 4; fpr: 0.20, fnr: 0.06; false negatives: 35/59.

4 Summary

The results presented suggest that either of the 4 approaches presented can find the single dominant feature (storm) in most examples with a low false positive rate. However, all methods have difficulty finding the other features, if any, in most examples. Our current application can accept a relatively high false negative rate; therefore, we have been at least partially successful. Although we had zero false positives among the 29 real test cases (we regard D1 as a training case), as discussed, we had a false

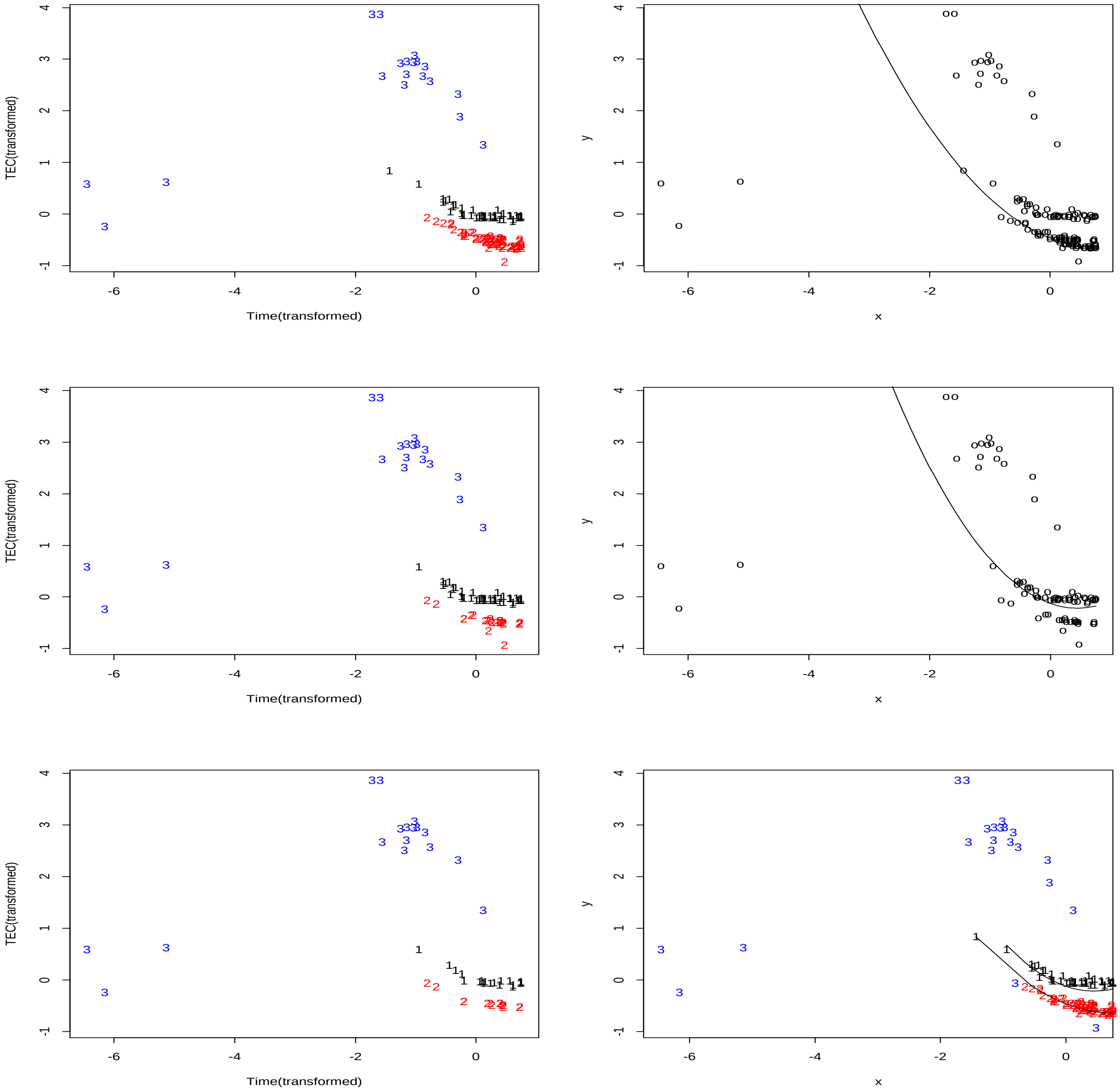

Figure 6. L2E method results on D1.

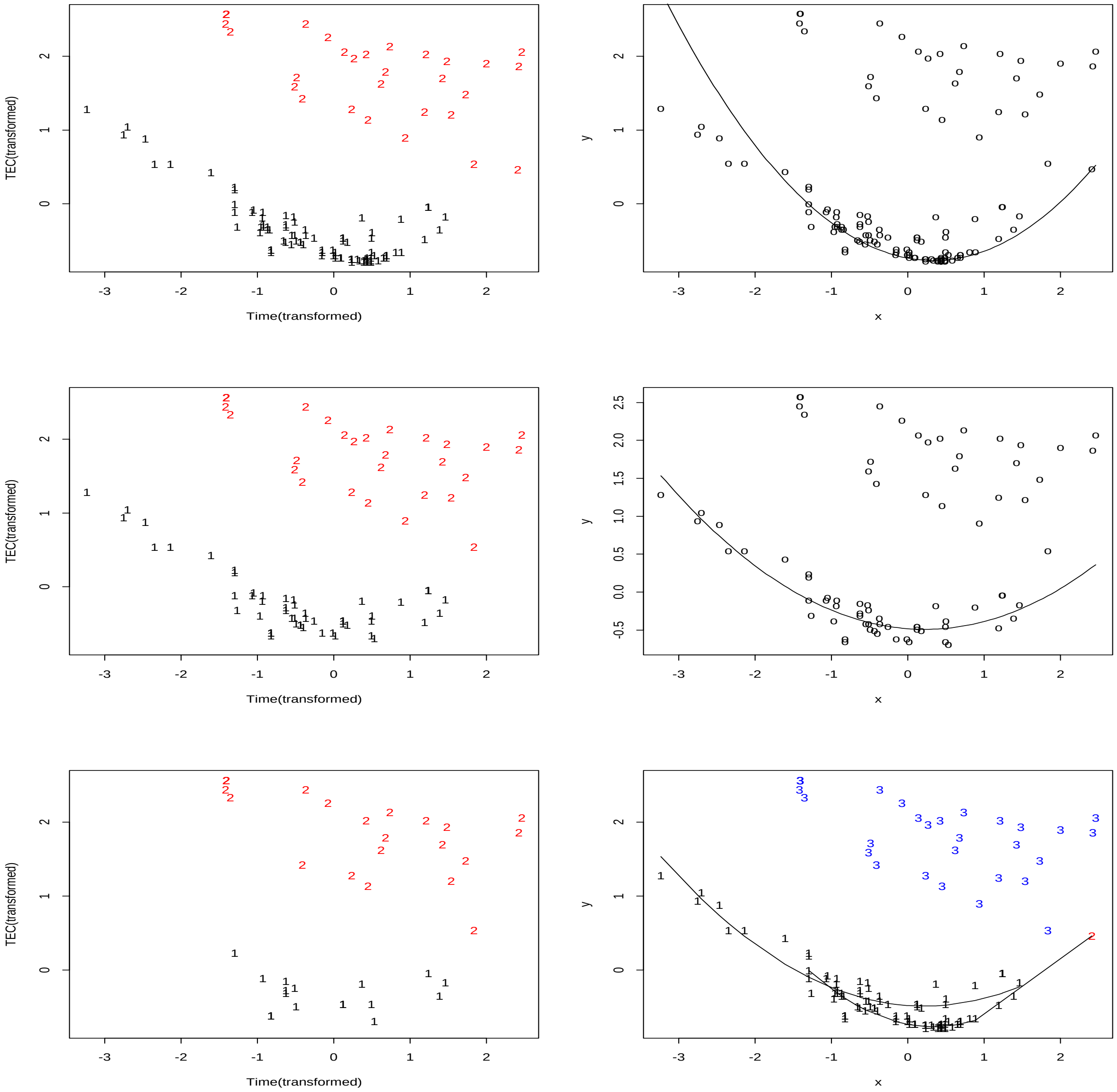

Figure 7. L2E method results on D3.

positive in a simulated case (if we use one particular definition of false positive) or at least a high false positive rate in the found cluster. This is undesirable in our intended use for the identified storms.

We believe that this data set provides rich opportunity for evaluating methods for fitting mixture models. It could be argued that because the BIC criterion did not indicate a clear signal in favor of the "correct" model that there is some inherent ambiguity in the analyst's labels. Therefore we think a useful next step would be to consider the data partition (into features and noise) having the highest posterior probability to be the "true model." This also has drawbacks because there is no guarantee that the highest posterior probability would belong (in practice) to the true model.

At present, we plan to use our conservative method 1 (low false positive rate method) involving noise rejection, hierarchical clustering, and possible relabeling of some noise points as feature points if they fall in a cluster's zone of ownership. However, we also plan a more extensive evaluation of the three alternatives presented here because we anticipate that some improvements could be made with modifications to the basic approach. Perhaps a blended approach using some information from each of the four methods applied to each new case would be effective.

5 References

Dempster, A., Laird, N., and Rubin, D. (1977), "Maximum Likelihood for Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Soc Series B* 39, 1-38.

Hastie, T., and Stuetzle W. (1989), "Principal Curves," *Journal of the American Statistical Assoc* 84, 502-516.

Jacobson, A., Knox, S., Franz R., and Enemark, D. (1999), "FORTE Observations of Lightning Radio-Frequency Signatures: Capabilities and Basic Results," *Radio Sci*, 34(2), 337-354. See also http://nis-www.lanl.gov/nis-projects/forte.

Moore, K., Blain, P., Briles, S., and Jones, R. (1995), "Classification of RF Transients Using Digital Signal Processing and Neural Network Techniques," *Applications and Science of Neural Networks, Proceedings SPIE*, 2492, 995-1006.

Stanford, D., and Raftery, A. (2000), "Finding Curvilinear Features in Spatial Point Patterns: Principal Curve Clustering with Noise," *IEEE Trans on Pattern Analysis and Machine Intelligence*, 22(6) 601-609.

Splus5 for Linux, Insightful Corporation, Seattle Washington, 1999.

Scott, D. (2002), "Parametric Statistical Modeling by Minimum Integrated Square Error," *Technometrics* 43(3), 274-285.

Scott, D., and Szewczyk, W. (2002), "From Kernels to Mixtures," *Technometrics* 43(3), 323-335.

Turner, R. (2000), "Estimating the Propagation Rate of a Viral Infection of Potato Plants via Mixtures of Regressions," *Applied Statistics* 49, part 3, 371-384.