Cluster Analysis of Simulated Gravitational Wave Triggers Using S-MEANS and Constrained Validation Clustering

Lappoon R. Tang† and Hansheng Lei† and Soma Mukherjee‡ and Soumya Mohanty‡§

† Computer Information Science Department, The University of Texas at Brownsville, Brownsville TX 78520, USA
‡ The Center for Gravitational Wave Astronomy, Department of Physics and Astronomy, The University of Texas at Brownsville, Brownsville TX 78520, USA

Abstract.

The fifth Science run of LIGO (S5) has been concluded recently. The data collected over two years of the run calls for a thorough analysis of the glitches seen in the gravitational wave channels, as well as in the auxiliary and environmental channels. The study presents two new techniques for cluster analysis of gravitational wave burst triggers. Traditional approaches to clustering treats the problem as an optimization problem in an "open" search space of clustering models. However, this can lead to problems with producing models that over-fit or under-fit the data as the search is stuck on local minima. The new algorithms tackle local minima by putting constraints in the search process. S-MEANS looks at similarity statistics of burst triggers and builds up clusters that have the advantage of avoiding local minima. Constrained Validation clustering tackles the problem by constraining the search in the space of clustering models that are "non-splittable" models in which centroids of the left and right child of a cluster (after splitting) are nearest to each other; the region of models that either over-fit or under-fit data (i.e. "splittable" models) can therefore be effectively avoided when assumptions about data are satisfied. These methods are demonstrated by using simulated data. The results on simulated data are promising and the methods are expected to be useful for LIGO S5 data analysis.

§ To whom correspondence should be addressed (lappoon.tang@utb.edu)

1. Introduction

The fifth Science run of the Laser Interferometric Gravitational Wave Observatories (LIGO) [1] came to an end in November 2007. This was the longest science run for the initial LIGO that lasted for about two years. The accumulated data spanned not only the gravitational wave (GW) channels in all the three interferometers (Hanford 4 kilometer (H1), Hanford 2 kilometer (H2) and Livingston 4 kilometer (L1)), but also several hundreds of auxiliary and environmental channels. The integrated data volume crossed several hundred terabytes over the duration of the run. Data are analyzed primarily with two goals : to look for presence of astrophysical signals in the data stream and to characterize the underlying noise. The second of these two goals leads to research in the LIGO Scientific Collaboration's (LSC) glitch group and detector characterization groups [2].

One of the main problems of the detector characterization research is to understand the source of the glitches seen in the GW channels. Typically, there are several thousands of glitches that show up in the GW channels. It thus becomes a daunting task to identify all of them manually. Several attempts are underway to analyze the glitches seen in the GW channels during S5, e.g. Q-scan [3], Block-Normal event display [4], Multidimensional classification analysis [5]. While partial success has been achieved, a lot more still remains desired.

Given the data size, use of data mining techniques is necessary in solving such problems. In the recent past, multidimensional hierarchical classification analysis has been applied to LIGO science data [6]. A wavelet-based event trigger generator (ETG) called the kleineWelle algorithm [7] generates burst-like events in the LIGO data stream in all channels. These data are stored in a protected database. Each trigger is characterized by GPS start and stop times, a central time, central frequency (estimated from the wavelet scale), duration, weighted and unweighted energy values and a significance parameter that indicates how strong the signal is. In the hierarchical classification analysis, metrics are constructed in the higher dimensional space. If there are *N* independent parameters describing a signal, this results into $N \times N$ metrics. Thus, to some extent, the accuracy of the result remains restricted by the number of parameters that can be used. In case of the kleineWelle database, this figure amounts of three - duration, central frequency and signal-to-noise ratio (snr) which is calculated from the energy values. However, given the richness of the data and the wide repertoire of signals that the GW channel is seen to contain, it is quite likely that a lot more structure is present in the multidimensional data space than revealed by the kleineWelle discreet database. Under this perspective, development for S5 burst classification algorithms has been undertaken that utilize not just the discreet parameters that the database offers, but rather the information contained in the actual waveform of the burst signal [8].

This paper describes two algorithms that we have been developing to address needs for data mining viz. S-MEANS and CV CLUSTER. The paper is organized as follows: Section 2 presents a short review of clustering algorithms, Sections 3 and 4 describe the algorithms S-MEANS and CV CLUSTER respectively, Section 5 shows the experimental results on simulated data, and finally Section 6 gives the conclusions and future directions.

2. Background on Clustering: Overview of K-means

Two clustering algorithms are most popularly used: hierarchical clustering and K-means. Hierarchical clustering produces a nested hierarchy of clusters according to a pairwise distance matrix of all the given points. The hierarchy gives intuitive visualization. A user does not need to have prior knowledge on the data since no parameter excepts distance measure is needed in hierarchical clustering. However, the distance matrix limits its application to small data sets (both time complexity and space complexity are $O(n^2)$ or higher).

K-means[9] basically divides a given data set into K clusters via an iterative refining procedure. The procedure simply consists of three steps:

(i) initialize K centroids (c_i, $1 \leq i \leq K$) in the vector space.

(ii) Calculate the distances from every point to every centroid. Assign each point to group i, if c_i is its closest centroid.

(iii) Update centroids. Each centroid is updated as the mean of all the points in its group.

(iv) If no point changed its membership or no centroid moved, exit, otherwise, go to step (ii).

The iterative procedure uses hill climbing to minimize the objective function:

$$J = \sum_{i}^{K} \sum_{j}^{N} \|x_j^{(i)} - c_i\|^2 \tag{1}$$

where $\|x_j^{(i)} - c_i\|^2$ denotes Euclidean distance between point x_j to corresponding centroid c_i. The Euclidean distance can be substituted by any distance measure.

Although the procedure will always terminate, K-means might converge to a local minima. K-means is a simple algorithm that has been employed in many data mining or data analysis tasks. However, one of the major problems of K-means is that we do not know the right number of clusters in advance. There is no existing theoretical solution to find the optimal number of clusters for any given data set. A common approach is to score the results of multiple runs with different K values according to a given criterion. The criterion might incur new risk and parameter setting problems. We propose to use a similarity driven approach to clustering that does not require specification of K.

3. S-MEANS: Similarity Driven Clustering

The clustering problem we need to solve is: *given N data points, group them into clusters such that within each cluster, all members have similarity $\geq$ T, a user-defined threshold, with the centroid.* Similarity is a central notion in classification problem. The definition of cluster also implies that the cluster members should have high similarity with each other. The most popular Euclidean distance is a dissimilarity measure, which can be converted to a similarity measure in Gaussian form: $k(x_i, y_j) = exp(-\gamma\|x_i - y_j\|^2)$. This is also called the Radial Basis Function (RBF kernel) in kernel machines. Kernel methods all use similarity measures instead of dissimilarity. Similarity value is usually normalized to between 0 and 1; a confidence threshold in [0, 1] also makes intuitive sense to users where 0 represents the extreme that there is absolutely no similarity between two items and 1 the other extreme. There are a large number of similarity measures available beside the RBF, such as correlation r, R-squared (the square of r) [10]. The similarity measure used by S-MEANS as the "default" is R-squared:

$$R^2 = \frac{\{\sum_{i=1}^{n}(x_i - \overline{X})(y_i - \overline{Y})\}^2}{\sum_{i=1}^{n}(x_i - \overline{X})^2 \sum_{i=1}^{n}(y_i - \overline{Y})^2}$$

where $X = (x_1, X_2, \ldots, x_n)$, and $Y = (y_1, y_2, \ldots, y_n)$ are two time series sequences.

However, any kernel function can be considered a similarity measure. Therefore, the clustering problem, if defined in terms of similarity, is more user-friendly and will likely gain more popularity due to the increasing amount of interests in kernel methods.

3.1. Algorithm description

S-MEANS starts from $K = 1$ by default and a user can specify any starting K. Note that the starting K is only an optional parameter in S-MEANS. First, same as in K-means, we initialize K centroids. Second, calculate the similarities from every point to every centroid. Then, for any point, if the highest similarity to centroid c_i is $\geq T$, group it to cluster i, otherwise, add it to a new cluster (i.e. the $(K+1)^{\text{th}}$ cluster). Third, update each centroid, using the mean of all member points by default. If one group becomes empty, remove its centroid and reduce K by 1. Repeat the second and third step until no new cluster is formed and none of the centroids moves.

Note that S-MEANS is somewhat similar to K-means but with significant differences. The major difference lies in the second step, which basically groups all the points to a *new* cluster whose highest similarity to *existing* centroids is below the given threshold . In K-means, all points must go to one of the existing K groups, which is unfair for some points when their similarities to corresponding closest centroid are very low. This simple difference makes big impact on the output of clusters. Also, we can let K starts from 1 and it will converge to a value, which eliminates the need of specifying a fixed K value. Also, there is a minor difference in the third step. While K is incremented by 1 if a new cluster is formed, it is decremented when some groups become empty. It is not unusual that as K keeps increasing, some old groups would disappear (as points in existing clusters could change membership as new clusters are formed). This way, K will not go beyond control.

Like K-means, S-MEANS also requires a parameter. However, it is a lower bound on the similarity of members in a cluster that is within 0% (members have no similarity to centroid) to 100% (members are identical to centroids). Such a parameter, that has semantics, is more meaningful than K. In a sense, one advantage of S-MEANS over K-means is the use of a parameter that "explains" K by describing property of data from which K is produced.

3.2. Time complexity and Termination

The termination of S-MEANS is guaranteed, because in the extreme case when K equals N every point has 100% similarity to itself. Of course, the extreme case is not desired. The result of K depends on threshold T. Intuitively, a high T produces more clusters. When

$T = 0$, S-MEANS is reduced back to K-means. In this sense, S-MEANS is a generalization of K-means.

If S-MEANS converges to K clusters, then time complexity is $O(N*(1+2+\cdots+K)) \approx O(N*K^2/2)$. Recall that the time complexity of K-means is $O(NKL)$, where L is the number of iterations, strongly related to K and the distribution of data points. If using model selection based method to try different K and choose the best one, then the time complexity is approximately $O(N*K^2/2*L)$, assuming K value varies from 1 to desired number of clusters. Besides avoiding the use of statistical tests (since both the number of data points and the data dimensionality could be high), S-MEANS has advantages in low time complexity. Readers are recommended to refer to [8] for more details.

4. Constrained Validation Clustering: The CV Cluster Algorithm

Another approach that tackles the problem of discovering relevant number of clusters is explored here. The motivation is that if one can categorize regions in the search space of clustering models, the search for a correct model can be constrained to specific regions.

4.1. Theoretical Intuition of the Algorithm

Before we proceed to describe the algorithm, let's first present the intuition of the algorithm that we call CV CLUSTER. Basically, the idea is that a "correct" cluster model is one such that it has a set of homogeneous and unique clusters; each cluster in the model contains data points generated by a single unique source, and no two clusters contain data points coming from the same source.

The theoretical principle behind the CV CLUSTER is the observation that suppose one has a set of data points D with the following property:
The centroid of cluster X is closer to that of Y than to that of Z if the data points in X and Y are produced by the same source but those in Z are produced by a different source.
Then, any cluster model M of D that consists of a set of homogeneous and unique clusters will have the following property:
For every cluster C in M, if we split C into two equal partitions X and Y, then the centroids of X and Y are closest to each other in M (i.e. C is a "non-splittable" cluster, and M is a "non-splittable" model).

The property about D is basically saying that centroids of a pair of clusters whose content is originated from the same source should be nearest to each other in a group of clusters. And, if this is the case, it follows intuitively that the centroids of a pair of homogeneous clusters (i.e. clusters whose content originated from the same source) should be nearest to each other in a group of clusters.

Therefore, the theoretical principle implies that if the assumption about D holds and a particular cluster model H under consideration is not a non-splittable model, one can conclude that H does not consist of a set of homogeneous and unique clusters. Hence, it cannot be a "correct" model for D.

This theoretical underpinning allows one to devise a strategy for the search of a "correct" clustering model given a set of data D. Suppose that the assumption about D holds. One can then narrow down the search space of clustering models to those that are non-splittable because a "correct" model cannot possibly be found outside the region of non-splittable models; any model discovered in the search process that is not non-splittable can be rejected without further consideration. Hence, we call our approach Constrained Validation CLUSTERing (CV CLUSTER).

However, it is unfortunately not the case that any non-splittable cluster model is a "correct" model for a data set D that satisfies the assumption. For example, any cluster model with only one cluster (i.e. the entire data set) is always trivially non-splittable but such a model cannot be correct for a data set with two or more real clusters. In other words, one still needs to provide a mechanism for determining if a cluster discovered in the search process is homogeneous and unique.

If a cluster model M consists of only unique clusters, to check if a particular cluster $C \in M$ is unique and homogeneous, obviously one only needs to check if C is homogeneous. A cluster C is homogeneous if and only if its two equal partitions C_x and C_y (i.e. $C_x \cup C_y = C$, $C_x \cap C_y = \emptyset$) contain data points produced by the same source. Hence, to check if a cluster $C \in M$ is homogeneous, one can compare the content of C_x to that of C_y to see if they may be produced by the same source.

To check if two clusters X and Y contain data points produced by the same source, for now, we use a heuristic test:
If $|\textit{avg-radius}(X) - \textit{avg-radius}(Y)| < \delta$, then X and Y are produced by the same source where *avg-radius*(C) = $\frac{\sum_{x \in C} d(x, mean(C))}{|C|}$, d is the Euclidean distance, and $\delta = 10^{-n}$ for some $n \geq 0$.

Our assumption is that if two clusters X and Y are produced by the same source, the sizes of the spheres X and Y should be similar (and thus so are their radii). Hence, if the gap between the radii of two clusters are "large", they would likely be produced by different sources. In that case, the cluster $C = X \cup Y$ is therefore not homogeneous. Otherwise, C is considered homogeneous. On the other hand, if C is not a non-splittable cluster, C is also not homogeneous assuming that the data set D satisfies the property mentioned above. Due to limitation in space, the proof for this theorem is left out.

How do we know if M consists of only unique clusters? We can ensure that M has only unique clusters if we choose only non-homogeneous clusters for splitting as non-unique clusters (i.e. clusters having data points produced by the same source) are only produced in the process by splitting a cluster that is already homogeneous. Ideally, each cluster is really unique and homogeneous. However, in practice, it is arguably acceptable that one only needs to ensure that a majority of the data points in a cluster are produced by the same source for that cluster to be considered practically homogeneous. Similarly, if a cluster contains very few data points produced by the same source as those contained in a different cluster, it can be considered practically unique.

The easiest way to think of δ is to treat it as a specification of the number of digits beyond the decimal point for which two numbers are required to be the same for them to be considered “equal”. For example, if $\delta = 0.1$, then a radius of 34.4xyz... and one of 34.4abc... are considered the same in size because they do not differ up to the first digit beyond the decimal point. A parameter is also used here, however, it is a specification of the level of precision required by a user for two clusters to be considered homogeneous. Therefore, although the system involves a parameter (like K-means), it is one such that a meaning (in this case, precision in numerical difference) can be attributed.

4.2. Description of the CV Cluster Algorithm

CV CLUSTER starts with one cluster (the entire set of data). It checks if all the clusters in the existing model M are homogeneous using the homogeneity test described above. If so, it terminates and returns M. Otherwise, the non homogeneous cluster whose splitting produces the child model that best optimizes the objective function in Equation (1) is chosen for splitting to produce a refinement of the existing model. If a model that is not “non-splittable” is generated in the process, it is rejected. This process repeats until a model with only unique and homogeneous clusters is found or the maximum number of clusters allowed has been reached.

5. Experiments on Mining Simulated LIGO Data

The artificial data set used in our experiments on mining simulated LIGO data has 20020 time series with a dimensionality of 1024. It was produced by Physics experimentalists. Each sequence is first generated by a single Gaussian modulated sinusoid signal. The amplitude is scaled such that the matched filtering signal to noise ratio (SNR) is 1 in white Gaussian noise with zero mean and unit variance ||. Then, a single Gaussian pulse is added to the signal in random position. The pulse amplitude is also scaled with SNR=1 in white Gaussian noise. Fig. 1 shows the typical shape of the each cluster. The simulated time series is a close representation of the actual triggers in Gravitational-wave Astronomy time series.

Although Gaussian process is used here, the possible clusters inside the data sets do not follow Gaussian distribution. As discussed in the introduction, GW events do not follow any known statistical distribution.

5.1. Results of S-MEANS

We applied S-MEANS to mine compact clusters in the data set of simulated LIGO time series. As S-MEANS could sometimes return a model with a cluster having only one member, we restart S-MEANS for a maximum number of times (100) if this happened. In addition, the upper bound on number of clusters allowed in a model is decremented by one each time it is restarted; the search space is progressively constrained in order to introduce a sufficient amount of bias in the search at some point so that S-MEANS can find a model that is small

|| White noise is used in the simulation data here as this was the first set of simulations that has been performed and our main goal was to understand the performance of the new algorithms in classifying burst triggers. More realistic noise models will be used in all future simulations.

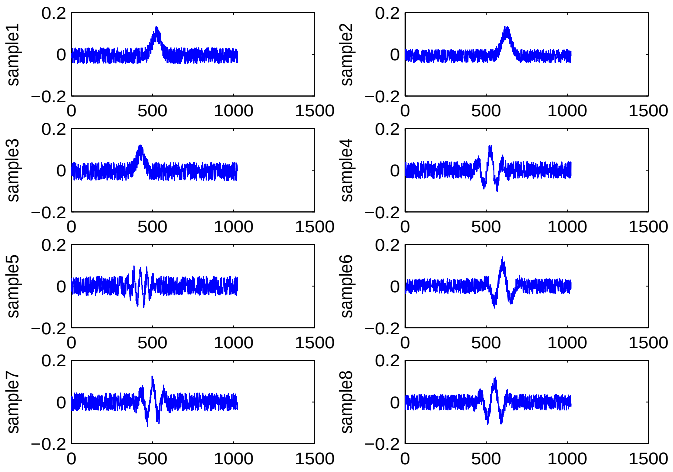

Figure 1: Samples of simulated Gravitational-wave time series.

Table 1: Number of clusters, average number of iterations, number of restart, execution time, average similarity change as similarity threshold T increases.

Threshold T	0.1	0.2	0.3	0.4	0.5	0.6
Number of clusters	10	11	13	60	96	97
Avg no. of iterations	54.00	15.00	17.03	89.43	104.4	106.25
Number of restart	0	1	28	39	4	3
Execution time (secs)	55.42	38.53	550.22	2596.99	355.66	237.24
Average similarity	0.56	0.58	0.63	0.65	0.66	0.67

enough to not overfit the data. Under this set up, S-MEANS was able to discover models without clusters having only one member. S-MEANS was allowed to start with five random initial centroids to facilitate the process of randomized restart. At the end, if S-MEANS did not find a model with clusters having more than one member in each, the model with the best average similarity is selected. Otherwise, the first model without a cluster having only one member is returned.

All the following experiments were performed in Matlab. The simulation is an example of a typical clustering task that arises in LIGO data cluster analysis and it also demonstrates the application of S-MEANS. As a part of GW data analysis, clustering time series based on “shapes” that can be matched by similarity measures is being experimented here. The simple similarity measure R-squared was used in S-MEANS.

S-MEANS was run on the data within a range of threshold values; this allows us to track its emerging behavior as the threshold T increases. For each threshold, the number of clusters found, the average number of iterations it took to discover the final model (i.e. total number of iterations divided by number of restart), number of times a restart is necessary, CPU seconds it took to find the final model, and average similarity of data points with respect to their corresponding centroids are reported. Results are summarized in Table 1.

Unlike what we expect, parameters related to computing resources needed to find a model (e.g. number of restart, execution time) do not always increase with respect to K. The most amount of computing resources seems to be needed at the "transition region" where K makes rapid jumps from small values to large values. For example, in our case here, this region lies in [0.3, 0.4]. In the transition region, S-MEANS would create a lot of clusters with only one member resulting in an explosion of number of clusters – when T approaches a "critical value". This is evidenced by the increase in the number of restart when T approaches 0.4.

When T is "sufficiently small", it is "easy" to put an arbitrary data point into a cluster because the criterion for membership is not strong. Likewise, when T is "sufficiently large" (so is the expected K), it is also "easy" to put an arbitrary data point into a cluster because the probability of membership increases with K. Thus, the most difficult time for S-MEANS to put a data point into a cluster occurs at the point when T is somewhere in between the two extremes.

Since clusters with only one member are created when it is most difficult to put a data point in any of the existing clusters, what is the implication for deciding on the K that is "correct"? When K is greater than the "correct" value, the number of restart should drop because probability of membership is larger. On the other hand, when K is too small, T is also less strong, and the number of restart should also drop. Thus, it follows that the "correct" K should occur at a value of T such that the number of restart is maximized – and this value has to be in the "transition region".

5.2. *Results of* CV CLUSTER

We also applied CV CLUSTER to discover if compact and well separated clusters exist in the simulated LIGO data. Since CV CLUSTER aims at discovering a model with unique and homogeneous clusters, one needs not specify K in advance for a given set of data although, similar to S-MEANS, a parameter δ (upper bound on absolute radii difference between two child clusters) that relates to the similarity of the sources of two clusters has to be specified. As *avg-radius(C)* $\leq$ *avg-radius(D)* if $C \subseteq D$, the radii gap can range from 0 to *avg-radius(D)* where D is the entire set of data. However, δ was set to 1.0 in our experiment.

CV CLUSTER discovered the existence of twenty unique and homogeneous clusters in the simulated LIGO data. We also compared the quality of the model discovered by CV CLUSTER to that of K-means, G-means, and S-MEANS. S-MEANS could sometimes produce a cluster with only one data point due to algorithmic randomness (depending on the initial seeding centroids). To avoid producing a trivial model, S-MEANS is allowed to restart if such a cluster is produced. T was set to 0.35. The number of restart was 45. G-means [11] is another clustering system that is capable of discovering the number of significant clusters in a set of data. However, it employs the assumption that homogeneous clusters obey a Gaussian distribution and thus a normality test such as the Anderson-Darling test is used to check if a cluster is homogeneous. Unfortunately, there is no guarantee that two Gaussian clusters cannot be produced by the same source. G-means could potentially overfit the data;

Table 2: The performance of CV Cluster, K-means, G-means, and S-means on clustering simulated LIGO data

Metrics Systems	CV Cluster	K-means	G-means	S-means
Number of clusters found	20	20	2882	55
K-means objective function	10939540.31	11380425.92	13136128.84	**3244124.10**
DB validation index	**2.30**	3.75	2.30	56.51

CV Cluster avoids this problem by not assuming that Gaussian clusters are produced by different sources. K in K-means was set to 20 for a meaningful comparison. K-means basically serves as a “baseline” for comparison in the experiment. The K initial centroids were selected using the seeding method in K-means++ [12] to help K-means overcome local minima. To quantitatively measure model quality, we employed two standard metrics – one is the objective function used by K-means and the other is the Davies-Bouldin (DB) index [13] that aims at identifying clusters that are compact and well separated. It is a cluster validation index commonly used in the clustering community for comparing model qualities. More precisely, a small value of the DB index indicates that a model has clusters that are compact and whose centers are far away from each other. Likewise, smaller K-means objection function values are more optimal.

Overall, CV Cluster found the correct number of clusters while clusters found by S-means are the most compact. CV Cluster outperformed both K-means and G-means in optimizing the K-means objective function and it produced a model with the best DB index value. S-means produced a model that best optimized the K-means objective function; each cluster in the model has the smallest total intra-cluster distance on average (and hence, each cluster is very compact). According to experimentalists, G-means produced too many clusters. The results can be found in Table 2.

6. Conclusions and future work

Two clustering algorithms have been presented, and applied on mining simulated LIGO data. Both were demonstrated to address problems with existing approaches such as hierarchial and K-means clustering. Also, both clustering algorithms discovered existence of significant clusters in the simulated LIGO data, which has been confirmed by Physics experimentalists. Despite that there could be disagreement between S-means and CV Cluster, clusters produced by both algorithms are interesting to experimentalists for further examination; they produce two independent views on the data. Perhaps the best way to use both clustering tools is to treat all the clusters produced by either one as candidates from which experimentalists can finalize to a “correct” list of clusters by tracing the clusters to their physical origins. One future work is to apply both approaches on clustering simulated LIGO data generated by more realistic noise models, and to use experimental results for algorithmic improvement.

7. Acknowledgment

This document is assigned LIGO DCC number LIGO-P080035-00-Z. The work is supported by NSF CREST grant 0734800 (2007).

References

[1] A. Abramovici et al. Ligo: the laser interferometer gravitational wave observatory. In *Science*, volume 256, 1992.

[2] gallatin.physics.lsa.umich.edu/keithr/lscdc/home.html.

[3] S. Chatterji. S5 glitch overview. In *LIGO Technical document G070138-00-Z*, 2007.

[4] S. Desai. New glitches seen/studied in block normal. In *LIGO Technical document G070105-00-Z*, 2007.

[5] S. Mukherjee et al. Preliminary results from the hierarchical glitch pipeline. In *Classical and Quantum Gravity*, volume 24, 2007.

[6] S. Mukherjee. Multi-dimensional classification of kleine welle triggers from ligo science run. *Classical Quantum Gravity*, 23:1–12, 2006.

[7] L. Blackburn et al. Glitch group s5 activities. In *LIGO Technical document G-060407-00-Z*, 2006.

[8] H. Lei, L. Tang, J. Iglesias, S. Mukherjee, and S. Mohanty. S-means: Similarity driven clustering and its application in gravitational-wave astronomy data mining. In *Proceedings of the International Workshop on Knowledge Discovery from Ubiquitous Data Streams (IWKDUDS 2007)*, Warsaw, Poland, 2007.

[9] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–297, January 1967.

[10] H. Lei, S. Palla, and V. Govindaraju. ER2: An intuitive similarity measure for on-line signature verification. In *the 9th International Workshop on Frontiers in Handwriting Recognition*, pages 191–195, 2004.

[11] Greg Hamerly and Charles Elkan. Learning the k in k-means. In *Advances in Neural Information Processing Systems*, volume 17, 2003.

[12] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

[13] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 1(2):224–227, 1979.