Indian Journal of Science and Technology, *Vol 9(26), DOI: 10.17485/ijst/2016/v9i26/93963, July 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Detection of Projected Outliers from the Higher Dimensional data sets using Extended Kalman Filter and Fuzzy K-Means

Kamal Malik[1*] and Harsh Sadawarti[2]

[1]IKGPTU, Jalandhar, Kapurthala - 144603, Punjab, India; kamal.malik91@gmail.com
[2]RIMTIET (Affiliated to Punjab Technical University), Godindgarh - 147301, Punjab, India, harshsada@yahoo.com

Abstract

Objectives: Curse of Dimensionality and the attribute relevance is the matter of great concern now these days while dealing with the higher dimensional data sets or Big Data, especially to detect the projected outliers. The objective of this research paper is to construct a Robust and a scalable model to prominently highlight the higher dimensional outliers in an effective and an efficient manner. **Methods/Analysis:** In order to detect the projected outliers, an algorithm EKFFK-Means with a hybrid approach is constructed using two important methodologies- Extended Kalman Filter (EKF) and Fuzzy K-Means. EKF is used to linearize the higher dimensional data by estimating the current mean and covariance by enhancing the Kalman gain and then fuzzy K-Means confirms the outlying property of each data instance and categorizes them in an effective and an efficient way using the membership label. **Findings**: A model EKFFK-Means is constructed that further creates 30 clusters from the complete data set to detect the projected outliers and various parameters like accuracy, cluster validity, True positive rate, False positive rate , robustness and cluster quality are calculated. **Improvements**: This algorithm is further compared with HPStream and CLUStream and is proved better against various parameters.

Keywords: Clustering, Projected Outliers, Robustness, Scalability, Unsupervised

1. Introduction

Due to the tremendous and voluminous amount of increasing data day by day, there is a prime requirement to advent the tools to deal with such data effectively and efficiently. Data Mining is one of the tools to discover the knowledge from such higher dimensional data. Outlier Mining is one of the very important techniques of data mining that highlights the anomalous and markedly inconsistent data instances from the normal data. Outliers are extreme observations fall away from normal observations. One source of outliers is transcription errors[1]. Uptil now, there are many methodologies evolved for outlier detection like distance based, Density Based, Neural Networks, and so on but still, among these, Clustering is the most prominent and effective technique to highlight the outliers. In context to the higher dimensional data, outliers are technically termed as projected outliers and the clustering is known as projected clustering. Projected clustering being an unsupervised technique effectively deals with the higher dimensional data without having any prerequisite training data set. More effectively and efficiently clustering is done, more prominently the outliers are highlighted and to detect, analyse and evaluate these projected outliers from the higher dimensional data streams and data is an important research problem.

An effective, efficient and a scalable clustering algorithm BIRCH[2] (Balanced Iterative Reduced Clustering using Hierarchies) was based on new-in-memory data structure called CF- tree and its performance is calculated under the parameters such as memory requirements, running time , clustering quality and stability. The research work[3] proposed a Robust Clustering algorithms named as CURE that employs the combination of random sampling

*Author for correspondence

and partition and identifies the non spherical shaped clusters with wide variances in size. CURE achieves this by representing each cluster by a certain fixed no. of points that are generated by selecting well scattered points from the cluster and then shrinking them towards the centre of the cluster by a specified fraction. Further this concept was extended in the similar study[4] in a way that all the points are ranked based on the outlier score by using the notion of nearest neighbourhood approach. Moreover, they provide the outlierness of each outlier and also specified the reason to call them as outlier. The author has worked on the higher dimensional outliers and technically named them as projected outliers. Moreover, the study[5] implemented a model named as SPOT i.e. Stream Projected Outlier Detection Technique to deal with the higher dimensional data. This model resolves the problem of curse of dimensionality according to which the outliers are embedded in some lower dimensional subspaces. The model spot has 100% detection rate and 0% false positive rate and the search workload of SPOT is significantly 35% to 43% higher than other existing methods. In the research[6] the scalable model for the aviation purposes is proposed and it worked on 60 dimensions and its experimental results are executed on the spatial and satellite data. The authors discussed two very important concepts Adaptive Mean Shift and Gaussian Mixture to help the pilots and co-pilots to detect and choose the particular buttons in abnormal situations. In the research work[7] kernel k-means presents a clustering method and kernel LOF based method to compute the likelihood values and then incorporated these generated values on SVDD (Support Vector Data Description) framework to build a more accurate classifier for global outlier detection and experimentally proved a better trade off between the detection rate and false alarm rate. In the similar study[8], an automatic and a non parametric method is proposed called an Outlier Detection Method (ODM) using Inter Quartile Range and estimated the no. of noise hyper sphere outliers. The study[9] implements the method, called Rough Outlier Set Extraction (ROSE) using rough set approximations i.e., lower and upper approximations and also introduced a kernel set to detect the outliers with a very less computational time as compared to the previous methods.

2. Issues Related to the Higher Dimensional Data

Actually all the outlier detection problems seem to be very much similar to each other or when compared with the other problems of data mining literature due to which they lose their algorithmic proficiency for the higher dimensional data.

i. *Distance Concentration* – It deals with the distances of the data instances within the cluster. The distance between the data instances should not be very less as due to the very less distance, they will superimpose each other's effect and with a very large distance, they will become equidistant from each other and this hides the outliers as their prominence will not be highlighted[10]. So, it's very necessary that the concentration of data instances should be at an appropriate distance. This is technically known as Cluster Validation Problem[11] that must be resolved to have the good quality and appropriate clusters.
ii. *Ranking of Outliers* – This is the concept of giving the scores to the outliers based on their outlying properties which was first of all given by **Ramaswamy**. The notion of the nearest neighbourhood approach was applied to it. Ranking also provides the idea upto which extent an outlier can be called as an outlier.
iii. *Relevant and Irrelevant Attributes*- To find out the exact relevance of the attributes is itself a great issue in the clustering especially in the case of the contextual outliers where one data instance may be an outlier with respect to one context and may not be an outlier with respect to the another context. To resolve this problem to judge the appropriate and relevant attributes, fuzzy logic must be used because crisp learning do not provide the exact knowledge of the relevance of the data instances.
iv. *Curse of Dimensionality*- As soon as the higher dimensionality of the data is encountered the data instances become quiet equidistant from each other due to which their outlying property is lost and outliers are hidden in some lower dimensional subspaces and their prominence is not highlighted. This problem is usually known as subspace selection problem or problem of Curse of Dimensionality which is very common in very large databases and this problem must be resolved.

3. Extended Kalman Filter

In almost all the real life processes which are nonlinear must be required to be linearized before they can be estimated by means of Kalman Filter. By calculating

the Jacobian of f and h around the estimated state, this problem of nonlinearity can be very well solved by EKF[12]. The calculation of Jacobian yields a trajectory of the model function centred on the state. This Mathematical Formulation can be given as shown in Figure 1.

An Extended Kalman Filter[14] is the enhanced and the non linear version of Kalman Filter which linearizes an estimate of current mean and co-variance[16]. It can be given as following:

$$X(t) = f\{x(t), u(t)\} + K\{(t)(Z(t) - h(x(t)\}$$

$$\frac{dp}{dt} = F(t)P(t) + P(t)F(t) - K(t)H(t)P(t) + Q(t)$$

$$K(t) = P(t)\,H(t)^{T}\,R(t)^{-1}$$

$$F(t) = \frac{\partial f}{\partial x}, X(t), u(t)$$

$$H(t) = \frac{\partial h}{\partial x}, X(t)$$

Where K is the Kalman Filter, P is the Predictor; H is the Estimator F is the Function

Observed data instance $_{\text{current time}}$ = Normal data instance + Outlying Deviation.

4. Fuzzy K-Means

In Contrast to the crisp logic, fuzzy logic indicates many degrees of memberships either from 0 to 1. A membership function $\mu_A(x)$ is associated with a fuzzy set such that the function maps every element of the universe of discourse X to the interval [0, 1].

Mathematically,

$$\mu_A(x): X \rightarrow [0, 1]$$

A fuzzy set is usually treated as if X is the particular element of X, then a fuzzy set A is defined on X may be written as a collection of ordered pairs :

$$A = \{(x, \mu_A(x))\}, x \in X$$

where each pair $(x, \mu_A(x))$ is called a singleton. Membership function values are not required to be described by discrete value which often turns out to be described by continuous function. Just like K-Means, Fuzzy K-Means also works on those objects that can be represented in n- dimensional vector Space. Fuzzy K-Means is comparatively better than simple K-Means as later discovers the hard clusters whereas the fuzzy is more statistically more formalized method and discovers the soft clusters where a particular point can belong to more than one cluster with certain membership degree. The basic steps of Fuzzy K-Means are:

- Initialize k clusters
- Convergence will be done to:
  - Compute the probability of a point belong to a cluster for every pair
  - Recompute the cluster centres using above probability membership values of points to clusters.

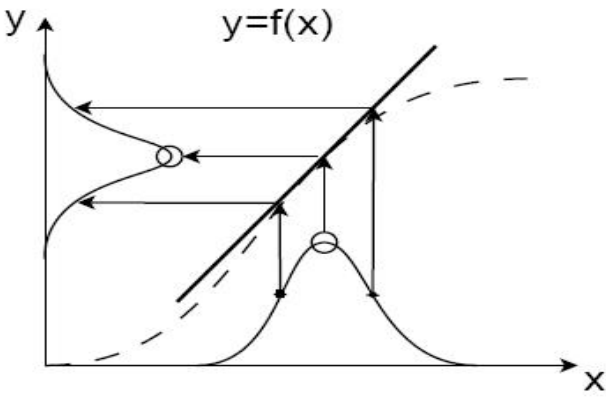

Figure 1. EKF linearizing a non-linear function around the mean of a Gaussian distribution[13]

5. Proposed Algorithm (Extended Kalman Filter and Fuzzy K-Means — EKFF-K Means)

The proposed algorithm is based on Extended Kalman Filter that is used to achieve and an approximate maximum likelihood estimation of the data instances .The data set taken to apply this algorithm is KDD-CUP'99 which is non linear data set. EKF firstly linearizes the estimate of the current mean and Co-variance and gives the appropriate value of the maximum likelihood of the data instances. Then Fuzzy K-Means is applied to provide the membership function and weight ranging from 0 to 1 to each of the data instances. The algorithm is as follows:

For each data instances (i, j, k ...etc,)

- Consider initially the value of k=30 implies that 30 clusters are to be performed.

- Extended Kalman Filter is applied along with the ScalerMin-Max function that will linearize the non linear dataset of KDD-CUP'99 using the following equations—

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \quad (1)$$

$$z_k = h(x_k) + v_k \quad (2)$$

where w_{k-1}, v_k are the observed extra data instances of different types

- In this step, the central centroid of each of the clusters is found by collecting the maximum likelihood values using the Predict and Update Equations
  - ***Predict***

$$x_k = f(x_{k-1},_{k-1}\, u_k) \quad (3)$$

$$P_{k,k-1} = F_{k-1}\, P_{k-1,k-1} + Q_k \quad (4)$$

  - ***Update***

Measurement of Outliers

$$Y_k = Z_k - h\,(x_k,_{k-1}) \quad (5)$$

Outlier Co-variance

$$Sk = H_k\, P_{k,k-1}\, H_k + R_k \quad (6)$$

Optimal Kalman Gain

$$K_k = P_{k,k-1}\, H_k\, S_k \quad (7)$$

Updated Centroid Estimate

$$X_{k,k} = x_{k,k-1} + K_k y_k \quad (8)$$

where F_{k-1} and H_k are state transition and observation metrics of EKF and are defined as

$$F_{K-1} = \frac{\partial F}{\partial x} \quad (9)$$

$$H_k = \frac{\partial H}{\partial x} \quad (10)$$

In the first step, centroid value is predicted and in the second one, the centroid is tried to refine upto much extent using update equations and optimal Kalman gain and Updated Centroid Estimation.

- Fuzzy output function f(μ) is computed as

$$f(\mu) = \text{fuzzy output function} = \Sigma f_{ik}(\mu) \quad (11)$$

where $f_{ik}(\mu)$ is defined as the consequent fuzzy output function when input 1 is in class i and an input 2 is in class k

$$f_{ik}(\mu) = w_{ik}\, f_{ik}(\mu) \quad (12)$$

w_{ik} is the activation level of that centroid

- Every time the centroid is refined and the value of w_{ik} will be given as

$$w_{ik} = \min[f_{i1}(\text{input 1}), f_{k2}(\text{input2})] \quad (13)$$

- Fuzzy K-Means is applied to partition a finite collection of n-elements X= { x_1, x_2,x_3, ...x_n) into a collection of k fuzzy clusters with respect to the above criteria of EKF . Fuzzy K-Means basically aims to minimize the objective function

$$\arg\min \sum_{i=1}^{n} \sum_{\substack{0\le i\le m \\ 0<j<n}} wij(xi - kj) \quad (14)$$

- Repeat step iii of EKF to again apply the maximum likelihood function on the data set by taking the new values of centroid until the minimum and the most appropriate value of the centroid is not found out.

6. Results and Discussions

The Implementation of the above algorithm EKFFK-Means is done in Python and on UBUNTU Operating System. The data set that is used is KDD'CUP'99 which is a non linear data set. A ScalerMin-Max function is used to normalize the data; moreover, two libraries of BIG data i.e., Pandas and Spark are used as KDD'CUP'99 is a data set of approximately 5-Lakh instances and is considered as a very higher dimensional data set. Then Extended Kalman Filter is applied and the value of k is given as 30, it means that the 30 clusters are to be considered ranging from 0 to 29. An optimal Kalman Gain is calculated on them and then fuzzy K-Means is applied on these 30 clusters to give them a proper weight through the membership label. Results are as shown in Table 1 that represents both the normal data as well as the outliers for each and every cluster as shown in Table 1. Various parameters of EKFFK-Means like Accuracy, Cluster Validity, True Positive rate, False Positive rate, Robustness in terms of Standard Deviation, Cluster Quality are calculated and are further compared with HPStream and CLUStream in Table 2 and the results obtained through the proposed algorithm are comparatively better than the HPStream and CLUStream.

Table 1. Normal data and the outliers of 30 clusters are shown

Cluster 1 labels:	**Cluster 2 labels:**	**Cluster 3 labels:**	**Cluster 4 labels:**
normal. 2996	smurf. 280606 normal. 32	neptune. 37320 portsweep. 17 normal. 2	neptune. 20456 portsweep. 58 satan. 10 normal. 4
Cluster 5 labels:	**Cluster 6 labels:**	**Cluster 7 labels:**	**Cluster 8 labels:**
normal. 8879 back. 23	normal. 1869 pod. 42 ipsweep. 7 satan. 6 smurf. 5 nmap. 3 neptune. 2 teardrop. 1 imap. 1 portsweep. 1	normal. 11431 smurf. 7	normal. 18744
Cluster 9 labels:	**Cluster 10 labels:**	**Cluster 11 labels:**	**Cluster 12 labels:**
normal. 4239 satan. 24 portsweep. 7	normal. 616 ipsweep. 3 back. 1	normal. 3551 guess_passwd. 49 back. 6 portsweep. 2 neptune. 1	ipsweep. 813 normal. 141 pod. 138 nmap. 99 land. 12 multihop. 1
Cluster 13 labels:	**Cluster 14 labels:**	**Cluster 15 labels:**	**Cluster 16 labels:**
normal. 3506 warezclient. 55 rootkit. 4 satan. 3 perl. 3 ftp_write. 2 ipsweep. 2 loadmodule. 1 spy. 1 imap. 1	neptune. 48382 portsweep. 5 dtype: int64	normal. 2778 back. 430 buffer_overflow. 14 loadmodule. 2 guess_passwd. 1	normal. 23782 back. 1491 phf. 3 satan. 1
Cluster 17 labels:	**Cluster 18 labels:**	**Cluster 19 labels:**	**Cluster 20 labels:**
satan. 1222 portsweep. 8	normal. 2572 back. 19	satan. 172 portsweep. 3	normal. 1237 satan. 109 teardrop. 1 portsweep. 1
Cluster 21 labels	**Cluster 22 labels**	**Cluster 23 labels**	**Cluster 24 labels**

Table 1 (*continued*)

normal. 1790 pod. 60 smurf. 38 satan. 22 teardrop. 7 rootkit. 3 nmap. 2 spy. 1	normal. 2097 back. 155 phf. 1 warezclient. 1	normal. 1244 ipsweep. 339 smurf. 134 pod. 24 nmap. 24 satan. 19 warezmaster. 18 imap. 6 land. 3 ftp_write. 2 rootkit. 2 guess_passwd. 2 multihop. 2 loadmodule. 1 portsweep. 1 dtype: int64	teardrop. 970
Cluster 25 labels:	**Cluster 26 labels:**	**Cluster 27 labels**	**Cluster 28 labels:**
normal. 1193 warezclient. 657 buffer_overflow. 16 loadmodule. 5 ftp_write. 4 back. 3 multihop. 2 rootkit. 1 dtype: int64	portsweep. 934 ipsweep. 83 normal. 3 dtype: int64	normal. 369 warezclient. 306 multihop. 2 warezmaster. 2 dtype: int64	normal. 3007 back. 75 satan. 1 dtype: int64
Cluster 29 labels:	**Cluster 30 labels:**		
normal. 1187 dtype: int64	neptune. 1040 nmap. 103 normal. 9 land. 6 imap. 4 portsweep. 3 warezclient. 1 guess_passwd. 1		

7. Comparison of EKFFK-Means with CLUStream and HPStream

In this section, this is elaborated that how the present algorithm is outperforming the already existing competitive techniques. A comparison of our Algorithm EKFFK-Means is done with very prominent projected clustering algorithms HPSTREAM[15] and CLUSTREAM[15] and various important parameters are calculated and shown below:

7.1 Scalability

The major issue of the algorithms available for projected outliers is Scalability and our implementation resolves it upto much extent. The algorithm is run for 3 conditions. In the first scenario, complete dataset is considered with 492452 samples. The second case uses 10% of the total dataset and the third case considers 1% of the data. It is found that for the first case, the accuracy comes out to be 98.2% and for the second case it decreases to 97.6%

Table 2. Comparison of EKFFK-MEANS with HPSTREAM and CLUSTREAM

PARAMETRES	HPSTREAM	CLUSTREAM	EKFFK-MEANS (*Proposed Method*)
ACCURACY	97%	96.5%	98.002%
CLUSTER VALIDITY	97.3%	96.5%	98.2%
TRUE +VE RATE	92%	73%	95.7%
FALSE +VE RATE	3%	1%	1.3%
Standard Deviation (Robustness)	0.6%	1.3%	0.7%
Cluster Quality	97.6%	97%	98.4%

while for the third case it is 95.1%. The slight decrease in the accuracy with reduction in data samples can be attributed to the reduced training sample. However the results are still above than 95% all the time which is better than HPSTREAM and CLUStream which has an accuracy of around 90% in all cases.

7.2 Robustness

The algorithm is run and executed for 50 Monte Carlo Simulations and the variance in results is calculated. The results in each cluster are found to deviate with a standard deviation of 0.7% which is quite low as compared to that of HPSTREAM and CLUStream.

8. Conclusion

The algorithm EKFFK-Means exploits the concept that while dealing with the higher dimensional data, crisp learning is itself not sufficient rather to resolve the various issues of Higher Dimensional Data and to obtain the accurate and precise projected outliers, fuzzy techniques must be utilized. Extended Kalman Filter is embedded with Fuzzy K-Means to obtain the desired results with an optimum Quality against various parameters.

9. Acknowledgement

Authors acknowledge the opportunity and support provided by I.K Gujral Punjab Technical University, Jalandhar to conduct the present work.

10. References

1. Ahmed SK, Naidu MM, Subha Rami Reddy C. Outliers/ Most Influential Observations in Variable Returns to Scale Data Envelopment Analysis. Indian Journal of Science and Technology. 2016 Jan; 9(2). DOI: 10.17485/ijst/2016/v9i2/80361
2. Zhang T, Ramakrishnan R, Livny M. Birch: A new data Clustering algorithm and its Applications. Data Mining and Knowledge Discovery. 1997; 1141–82.
3. Guha S, Rastogi R, Shim K. CURE: An efficient clustering algorithm for large databases. SIGMOD Rec. 1998; 27(2):73–84.
4. Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets. 2000; 427–38.
5. Zhang J. PhD Thesis by entitled as Towards Outlier Detection for higher Dimensional data streams using projected outlier analysis strategy. 2009.
6. Zvitia O, Mayer A. Co-registration of White matter Tractographies by Adaptive Mean shift and Gaussian Mixtures. 2001.
7. Liu B, Xiao Y, Yu PS, Hao Z. An Efficient Approach for outlier detection with imperfect data labels. IEEE Transactions on Knowledge and Data Engineering. 2014 Jul; 26(7).
8. Wu S, Wang S. Information- Theoretic Outlier Detection for Large Scale categorical data. IEEE Transaction of knowledge and Data Engineering. 2013 Mar; 25(3).
9. Andreou C, Karathanassi V. Estimation of the Number of End members using Robust Outlier Detection Method. IEEE journal of selected topics in Applied earth observations and Remote sensing. 2014 Jul; 7(1).
10. Aggarwal CC. A Human-Computer Interactive Method for Projected Clustering. IEEE Transactions on Knowledge and Data Engineering. 2004; 16(4):448–60.
11. Aggarwal CC, Procopiuc C, Wolf J, Yu PS, Park JS. Fast algorithms for projected clustering. ACM SIGMOD Conference, 1999
12. Available from: http://en.wikipedia.org/wiki/Kalman_filter, 2016 Apr14.
13. Madhumita M. PhD thesis entitled as Study Of Kalman, Extended Kalman And Unscented Kalman Filter NIT Rourkela in 2010.
14. An Introduction to the Kalman Filter , Greg Welch and Gary Bishop, TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill-, NC-27599-3175, 24th July, 2006
15. Aggarwal CC, Han J, Wang J, Yu P. A Framework for Clustering Evolving Data Streams. VLDB Conference, 2003.
16. Babcock B, Babu S, Datar M, Motwani R, Widom J. Models and Issues in Data Stream Systems, ACM PODS Conference, 2002.