

LINEX K-Means: Clustering by an Asymmetric Dissimilarity Measure

Narges Ahmadzadehgoli

Department of Statistics, Science and Research Branch, Islamic Azad University, Tehran, Iran
n_ah652@yahoo.com

Adel Mohammadpour¹

*Department of Statistics, Faculty of Mathematics & Computer Science,
Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez Ave., Tehran, Iran.*
adel@aut.ac.ir

Mohammad Hassan Behzadi

Department of Statistics, Science and Research Branch, Islamic Azad University, Tehran, Iran
behzadi@srbiau.ac.ir

Received 5 October 2016

Accepted 6 March 2017

Abstract

Clustering is a well-known approach in data mining, which is used to separate data without being labeled. Some clustering methods are more popular such as the k-means. In all clustering techniques, the cluster centers must be found that help to determine which object is belonged to which cluster by measuring the dissimilarity measure. We choose the dissimilarity measure, according to the construction of the data. When the overestimation and the underestimation are not equally important, an asymmetric dissimilarity measure is appropriate. So, we discuss the asymmetric LINEX loss function as a dissimilarity measure in k-means clustering algorithm instead of the squared Euclidean. We evaluate the algorithm results with some simulated and real datasets.

Keywords: LINEX loss function, dissimilarity measure, k-means clustering

1. Introduction

Clustering is a popular approach in data mining, which is used to separate data without being labeled [10]. Using a clustering algorithm, a dataset is partitioned into groups with the largest similarity within a group, in comparison with others [12]. In all clustering methods, each object belongs to a cluster according to the cluster center, by measuring the dissimilarity. The optimal centers of a clustering algorithm depend on the loss function. The most popular partitioning algorithm is the k-means. It partitions a dataset into some different class of similar entities, which minimizes the dissimilarity between the entities and their related cluster centroids [3]. The dissimilarity measures have an important role in clustering. Many loss functions such as Euclidean squared, City-Block,

¹ Corresponding author. Tel.: +98 2164542533; Fax: +98 2166497930; E-mail: adel@aut.ac.ir.

Minkowsky and Relative Entropy have been used to measure the dissimilarity in the clustering and the Euclidean squared measure is the widely used one. Clustering with the Bregman divergence was introduced by Banerjee et al. [4] in which the loss of Bregman information is minimized. The minimum is always equal to the expected Bregman divergence of data and their relative cluster centroid [4]. The optimal point in a symmetric loss function is the conditional mean of the variable.

Suppose that $\theta \in \Theta$ be an unknown parameter and $\delta(X)$ be an estimator of θ based on a random observation X . Granger (1999) stated some required properties for a loss function $L(\Delta)$ where Δ is the estimated error, $\delta(X) - \theta$, and $L(\Delta)$ is the loss of estimating θ by the value $\delta(X)$, as the following [7].

- a) $L(0) = 0$,
- b) $\min_{\Delta} L(\Delta) = 0$, so $L(\Delta) \geq 0$,
- c) $L(\Delta)$ is monotonic non-decreasing.

He also stated that a loss function might have some of the following properties:

- a) Symmetry, i.e., $L(\Delta) = L(-\Delta)$,
- b) Homogeneous, i.e., $L(a\Delta) = Q(a)L(\Delta)$, where $Q(a)$ is a positive function,
- c) Differentiability in order p .

The convex k-means clustering algorithm was proposed by Modha and Spangler [14]. They stated that the loss function, which is used as the dissimilarity measure, should be non-negative, convex and symmetric. They generalize k-means to the convex k-means algorithm [14]. When the positive and negative errors are of the same importance, the symmetric loss functions are used to evaluate the dissimilarities. However, in many situations, we need asymmetric measures. In 2005, Kummamuru et al. [13] proposed the class of asymmetric dissimilarity measures and referred them as the context sensitive learnable asymmetric dissimilarity measures.

We choose the loss functions according to the construction of the data. Parsian and Kirmani [15], stated that when the overestimation and the underestimation are not equally important, the symmetric loss functions are not appropriate. For example, overfilling the containers in food-processing industries is more undesirable than under filling them [9]. The underestimation of water level is much more important than the overestimation when constructing a barrier [20]. Varian [19] proposed an asymmetric loss function, called linear exponential (LINEX) loss function. A LINEX loss function is convex and asymmetric. On one side of zero, it is approximately exponential and on the other side, it is linear. To study more about the properties of LINEX loss function, we refer to [20].

In this work, first, we review k-means and k-median algorithms and some required concepts. Then we propose k-means clustering algorithm based on the asymmetric loss function, LINEX. We use some benchmarked dataset that their labels are available to check if the k-means clustering with LINEX loss function (LINEX k-means) could separate data well and to show its application. The available labels are useful to evaluate the accuracy of the algorithm. We also use the normalized variation information and the Davies-Bouldin index to evaluate the clustering algorithm results. Some simulated datasets are generated to complete evaluations.

2. The k-means clustering algorithm

Suppose we have a dataset $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n)'$, which contains n entities $\mathbf{X}_i = (X_{i1}, \dots, X_{im})$ with m features. The k-means is an algorithm that used to partition \mathbf{X} into K disjoint groups $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_K\}$ of similar entities, which

minimizes the dissimilarity between the entities and their related cluster centroids [3]. The cost (error) function between \mathbf{X}_i in group k and its respective centroid, \mathbf{C}_k , which is randomly selected, is as follows:

$$J = \sum_{k=1}^K \sum_{\mathbf{X}_i \in S_k} L(\mathbf{X}_i, \mathbf{C}_k),$$

where L is the Euclidean distance (or loss) between $\mathbf{X}_i \in S_k$ and \mathbf{C}_k . Each entity assigns to the nearest centroid \mathbf{C}_k , and the function J is computed. The algorithm stops if it is less than a determined threshold. Otherwise, the cluster centers are updated to the mean of S , until the new centroid does not differ from the previous one. The implementation of this algorithm depends on the number of clusters and their initial centroids. Therefore, the algorithm must be performed many times with different initial centroids. Performing centers continue until there are no non-clustered entities. The optimal centers of a clustering algorithm mostly depend on the dissimilarity measure that is used. In k-means clustering, if we use the Manhattan distance which is the absolute difference between two points as the dissimilarity measure, then the centers are updated at their median rather than their means and the outliers have less influence on the centroids [6]. This algorithm is named ‘k-median’.

3. LINEX loss function

The LINEX is one of the interesting non-negative, convex and asymmetric loss function that is differentiable in order $p \geq 1$ and takes the following form [15],

$$L(\Delta) = \exp(a\Delta) - a\Delta - 1, \quad (1)$$

where a is a scalar and Δ denotes the estimation error of using $\delta(X)$ in order to estimate θ . If $a > 0$ LINEX is exponential for positive Δ and if $a < 0$ it is linear in negative Δ . When $a = 1$, $L(\Delta)$, is quite asymmetric and the overestimation is costlier than the underestimation. For small $|a|$ (close to zero), the loss function is nearly (but not completely) symmetric and it is similar to the squared Euclidean loss function. So the optimal estimates which obtained by the squared and LINEX loss functions are not so different when $|a|$ is small [15], but for larger values of $|a|$, the optimal points are quite different. In Figure 1, for $a = -1$, $a = 0.001$ and $a = 1$, we plot $L(\Delta)$ versus Δ .

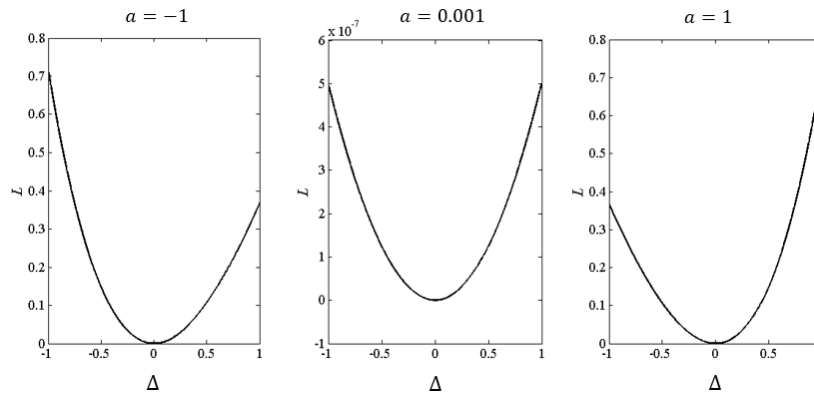


Fig. 1. A LINEX loss function, $L(\Delta)$, $\Delta \in (-1, 1)$, for different values of a

We can extend the LINEX loss function to estimate multi parameters [15]. Suppose $\delta_j(X)$ is an estimator of θ_j and $\Delta_j = \delta_j(X) - \theta_j$, for $j = 1, \dots, m$. We extend (1) to the m -parameter case as bellow,

$$L(\mathbf{\Delta}) = \sum_{j=1}^m \{\exp(a_j \Delta_j) - a_j \Delta_j - 1\}, \quad (2)$$

where $a_j \neq 0$ and $\mathbf{\Delta} = (\Delta_1, \dots, \Delta_m)$.

4. LINEX k-means clustering

In this section, we want to use the LINEX loss function as the dissimilarity measure in k-means clustering algorithm when the overestimating and the underestimating are not of the same importance. The procedures are the same as a k-means clustering algorithm. We define the number of clusters (according to any knowledge, which is related to the data) and the initial centroids are randomly chosen from the entities. All the entities are assigned to their nearest centroid, using a LINEX loss function as the dissimilarity distance. The procedure continues until there is no change in clusters. Now consider the following optimization problem,

$$G(\mathbf{H}, \mathbf{C}) = \sum_{k=1}^K \sum_{i=1}^n h_{ik} L_{LINEX}(\mathbf{X}_i - \mathbf{C}_k),$$

where $\mathbf{C} = (\mathbf{C}_1, \dots, \mathbf{C}_K)'$, $\mathbf{C}_k = (C_{k1}, \dots, C_{km})$, for $k = 1, \dots, K$ and L_{LINEX} is the LINEX distance in (2) and \mathbf{H} is an $n \times K$ matrix such that, for each $i = 1, \dots, n$,

1. $h_{ik} \in \{0, 1\}$,
2. $\sum_{k=1}^K h_{ik} = 1$.

Now, we want to minimize $G(\mathbf{H}, \mathbf{C})$ according to the above conditions similar to k-means [5, 17]. Therefore, we may solve the problem in the following two steps:

Step 1: Fix $\mathbf{C} = \hat{\mathbf{C}}$ and solve $G(\mathbf{H}, \hat{\mathbf{C}})$, it minimized if and only if

$$h_{ik} = \begin{cases} 1, & L_{LINEX}(\mathbf{X}_i - \mathbf{C}_k) < L_{LINEX}(\mathbf{X}_i - \mathbf{C}_t) \text{ for } 1 \leq t \leq K, \\ 0, & \text{otherwise.} \end{cases}$$

Step 2: Fix $\mathbf{H} = \hat{\mathbf{H}}$, then $G(\hat{\mathbf{H}}, \mathbf{C})$ is minimized if and only if for each component

$$C_{kj} = \frac{1}{a} \log \frac{\sum_{i=1}^n h_{ik} e^{a X_{ij}}}{\sum_{i=1}^n h_{ik}}, \text{ for } j = 1, \dots, m.$$

To prove, it is sufficient to minimize the following inner summand for fixed k ,

$$\sum_{i=1}^n h_{ik} \left(e^{a(X_{ij} - C_{kj})} - a(X_{ij} - C_{kj}) - 1 \right).$$

To do this, we differentiate with respect to C_{kj} , and the result is obtained. One can use the following algorithm to optimize $G(\mathbf{H}, \mathbf{C})$ [11].

1. Set an initial \mathbf{C}^0 and solve $G(\mathbf{H}, \mathbf{C}^0)$ in order to obtain \mathbf{H}^0 (we set $t = 0$).
2. Put $\hat{\mathbf{H}} = \mathbf{H}^t$ to solve $G(\hat{\mathbf{H}}, \mathbf{C})$ and then obtain \mathbf{C}^{t+1} .
We stop the algorithm if $G(\hat{\mathbf{H}}, \mathbf{C}^t) = G(\hat{\mathbf{H}}, \mathbf{C}^{t+1})$, otherwise, we go to the next step.
3. This time, let $\hat{\mathbf{C}} = \mathbf{C}^{t+1}$ and solve $G(\mathbf{H}, \hat{\mathbf{C}})$ to achieve \mathbf{H}^{t+1} .

We stop the algorithm if $G(\mathbf{H}^t, \hat{\mathbf{C}}) = G(\mathbf{H}^{t+1}, \hat{\mathbf{C}})$, otherwise, put $t = t + 1$ and go to 2.

It can be proved that $G(\cdot, \cdot)$ is strictly decreasing and similar to [14], so after some iterations, this algorithm converges to a minimum point.

The k-means algorithm based on LINEX loss function (instead of Euclidean distance) is the same as k-means except in distance measure and the optimal cluster centers.

5. Evaluation

In this section, we want to evaluate LINEX k-means clustering to partition some simulated and real datasets. To do it, we use the NVI, which is an external clustering evaluation measure and an internal criterion DB index. If NVI value lies in $[0,1]$, the clustering performance considered as good. It decreases when the clustering is more homogeneous. Internal criterion is useful to evaluate a clustering algorithm, according to the inherited features and qualities of a dataset. DB criterion is based on the dispersion measure and the cluster similarity measure between each cluster. When DB index value is small, it means the clusters are separated better.

5.1. Results on simulated datasets

Gaussian dataset: We generated a Gaussian dataset of 500 observations, each with 6 features, a mixture coefficient equal to 0.2 and partitioned it into five clusters. This Gaussian model is generated using a NETLAB function and has spherical clusters of variance 0.1. The centers are generated independently from a $N(0,1)$ [1]. Now consider the following density functions and the relations between their independent variables [18]:

	Density, $f(\mathbf{x})$	Relations between independent variables
Log-Normal (LN)	$\frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\log(x)-\mu)^2}{2\sigma^2}\right), x > 0, \sigma > 0, \mu \in \mathbb{R}$	The product of n iid LN random variables is LN.
Cauchy	$\frac{1}{\pi\sigma(1+(\frac{x-\mu}{\sigma})^2)}, x, \mu \in \mathbb{R}, \sigma > 0$	The sum of n iid Cauchy, Gamma and Poisson random variables have the same distributions, respectively.
Gamma	$\frac{\beta^\alpha x^{\alpha-1} e^{-x\beta}}{\Gamma(\alpha)}, x \geq 0, \alpha, \beta > 0$	
Poisson	$\frac{e^{-\lambda} \lambda^x}{x!}, x = 0, 1, \dots$	

“iid” means independent and identically distributed

We generate four datasets according to the above density functions such that each dataset contains 100 dependent entities with m features and partitioned into two clusters. We state the procedure in Table 1.

Table 1: The procedure of generating some datasets of 100 dependent variables with k features that partitioned into 2 clusters.

Dataset		$Y = (Y_1, \dots, Y_{100})'$
Multivariate with 3 features		$Y_i = (X_{i1} + X_0, X_{i2} + X_0, X_{i3} + X_0) = (Y_{ij})$ for $j = 1, 2, 3$
	Cauchy	$X_0, X_{ij} \stackrel{iid}{\sim} \text{Cauchy}(0, 1), Y_{ij} \stackrel{id}{\sim} \text{Cauchy}(0, 2)$ for $i = 1, \dots, 50$ $X_{ij} \stackrel{iid}{\sim} \text{Cauchy}(4, 1), Y_{ij} \stackrel{id}{\sim} \text{Cauchy}(4, 2)$ for $i = 51, \dots, 100$
	Gamma	$X_0, X_{ij} \stackrel{iid}{\sim} \text{Gamma}(1, 3), Y_{ij} \stackrel{id}{\sim} \text{Gamma}(2, 3)$ for $i = 1, \dots, 50$ $X_{ij} \stackrel{iid}{\sim} \text{Gamma}(6, 3), Y_{ij} \stackrel{id}{\sim} \text{Gamma}(7, 3)$ for $i = 51, \dots, 100$
Univariate	LN	$Y_i = (X_{i1}X_0, X_{i2}X_0, X_{i3}X_0) = (Y_{ij})$ for $j = 1, 2, 3$ $X_0, X_{ij} \stackrel{iid}{\sim} \text{LN}(1, 1), Y_{ij} \stackrel{id}{\sim} \text{LN}(2, 2)$ for $i = 1, \dots, 50$ $X_{ij} \stackrel{iid}{\sim} \text{LN}(20, 1), Y_{ij} \stackrel{id}{\sim} \text{LN}(21, 2)$ for $i = 51, \dots, 100$
	Poisson	$Y = (Y_1, \dots, Y_{100}), Y_i = X_i + X_0$ $X_0, X_i \stackrel{iid}{\sim} \text{Poisson}(1), Y_i \stackrel{id}{\sim} \text{Poisson}(2)$ for $i = 1, \dots, 50$ $X_i \stackrel{iid}{\sim} \text{Poisson}(10), Y_i \stackrel{id}{\sim} \text{Poisson}(11)$ for $i = 51, \dots, 100$

Now we run k-means, k-median, and LINEX k-means algorithms 500 times and compute the average accuracy results, i.e., an average of one minus percent of misclassification observations for 500 iterations. We note that, in LINEX k-means algorithm, the initial centers are chosen randomly from the dataset members similar to the k-means. Then, all the n entities are assigned to their relative centers according to the minimum LINEX rule. The results are presented in Table 2. They illustrate that the LINEX k-means clustering algorithm is sometimes more accurate in comparison with the traditional k-means and k-median. In our simulated dataset, the overestimation and the underestimation are not important for us, so we choose the parameter a close to zero. Therefore, the LINEX k-means performance is not so different from the k-means. If we choose $a \geq 0.1$ the accuracies decrease, and some of the data are pushed to the wrong clusters because of overestimation.

In some cases, an event might be near the border of two clusters and it is hard to assign it to the right cluster. For example, in each cluster of the above Poisson dataset, some events are of the same values. However, they are generated from two different Poisson distribution with different parameters. Suppose we like to assign these events to the cluster, which is generated from Poisson with $\lambda = 1$. Using the above algorithms (without assuming the over or underestimation) they might be misclassified. But if we use the LINEX k-means algorithm with $a > 0$, it means that the overestimating is more important, these mentioned entities are guided to be placed in the second cluster and the results are improved, see Table 3. When assuming overestimating, we choose a in step of 0.1 from 0.1 to 2. The lower values of NVI and DB helped us to choose a . When there are no overestimating and underestimating assumptions, a is chosen to be closed to zero. Now we want to see whether if the skewness of the entities might affect the value of a . We generate three datasets from Exponential distribution and classify them into two classes.

Table 2: The accuracy percentage, NVI, and DB values for k-means, k-median, and LINEX k-means clustering algorithms in the simulated datasets.

Dataset	Algorithm Evaluation index	k-means	k-median	LINEX k-means	
		Results	Results	Results	α
Gaussian	Accuracy	82.90	83.01	87.37	10^{-2}
	NVI	0.243	0.239	0.221	
	DB	0.785	0.813	0.637	
Log-normal	Accuracy	71.00	81.00	79.04	10^{-3}
	NVI	0.831	0.720	0.733	
	DB	0.737	0.812	0.862	
Cauchy	Accuracy	97.00	98.00	97.00	10^{-2}
	NVI	0.322	0.217	0.247	
	DB	0.779	0.786	0.779	
Gamma	Accuracy	98.00	98.00	98.00	10^{-2}
	NVI	0.217	0.217	0.217	
	DB	0.626	0.626	0.626	
Poisson	Accuracy	94.00	96.01	95.80	10^{-2}
	NVI	0.525	0.455	0.457	
	DB	0.449	0.448	0.448	

In the LINEX k-means algorithm, we assume that there is no overestimation or underestimation, so the parameter α is closed to zero.

Table 3: The accuracy percentage, NVI, and DB values for LINEX k-means clustering algorithms with different α in the simulated Poisson datasets.

Dataset	Evaluation Index	α :	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Poisson	Accuracy		94.14	94.26	94.38	94.44	96.12	96.24	96.64	92.72
	NVI		0.447	0.447	0.447	0.447	0.318	0.225	0.203	0.504
	DB		0.448	0.448	0.448	0.448	0.449	0.449	0.450	0.476

We compute the skewness of them and run the k-means and the LINEX k-means algorithms with different values of α . The results are presented in Table 4. According to the results, it seems that the skewness affects the value of α , and in the skewed datasets, the accuracies of LINEX k-means algorithm with $\alpha > 0$ are better than k-means algorithm. Therefore, we suggest scrutinizing how choosing the value of α depends on the skewness of the frequency curve of a dataset.

Table 4: Evaluating the result of clustering three Exponential datasets with LINEX k-means and k-means algorithm. The first row for each skewness parameter is the accuracy percentages and the second row is the DB index.

Skewness Parameter	Algorithm Validation Index	k-means	LINEX k-means										
		α :	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.65	Accuracy	91.00	91.00	92.35	93.25	93.17	94.15	93.70	93.70	92.80	92.80	93.25	93.25
	DB	0.431	0.431	0.453	0.423	0.421	0.420	0.421	0.421	0.429	0.429	0.423	0.423
1.50	Accuracy	95.00	95.00	95.00	95.00	95.00	95.00	95.00	99.00	99.33	98.66	98.33	97.33
	DB	0.445	0.450	0.450	0.450	0.450	0.450	0.450	0.466	0.468	0.463	0.462	0.455
2.00	Accuracy	93.00	94.46	94.73	94.60	94.86	94.86	94.73	94.60	94.60	84.00	80.26	79.20
	DB	0.375	0.389	0.391	.390	0.393	0.393	0.391	0.390	0.390	0.492	0.533	0.556

5.2. Evaluation in some real datasets

Here we compare the results of LINEX k-means with k-means and k-median algorithms on some real datasets. We run the algorithms 500 times and compute the average of accuracies due to a random selection of centers for each algorithm. These datasets are all available in the UC Irvine Machine Learning Repository [2]. All the data are labeled so that we can evaluate the algorithms. We normalize the data before running the algorithm. The results are summarized in Table 5.

- The Iris data set is a collection of 150-flower specimen each with four features and is divided into three clusters.
- The Wine dataset consists of 178 chemical specimens of wine with thirteen features that partitioned into three clusters.
- The Hepatitis dataset contains 155 specimens with 19 categorical and numeric features, which is partitioned into two clusters.

Table 5: The accuracy percentage, NVI, and DB values for k-means, k-median, and LINEX k-means clustering algorithms in different datasets.

Dataset	Algorithm Evaluation Index	k-means	k-median	LINEX k-means
Iris	Accuracy	85.33	78.39	84.87
	NVI	0.485	0.528	0.495
	DB	0.618	0.630	0.539
Wine	Accuracy	98.87	94.83	99.77
	NVI	0.100	0.302	0.075
	DB	0.489	0.508	0.474
Hepatitis	Accuracy	68.90	64.98	75.74
	NVI	0.766	0.850	0.549
	DB	1.078	1.620	0.669

In the LINEX k-means algorithm, we assume that there is no overestimation or underestimation. So we consider the parameter a close to zero ($a = 1 \times 10^{-3}$).

The results in Table 5 shows that LINEX k-means algorithm acts as well as the others and in some cases, it leads to better results. Normalizing the data improves the accuracy.

As we state before, sometimes the overestimating or underestimating is of different importance (For example, estimating the number of earthquakes with high magnitude during a short period in an Earthquake-prone area) and classifying an event in a specific cluster is worse (or better). Now consider three datasets: Seismic bumps, Haberman’s Survival and Magic Gamma Telescope.

- The seismic bumps dataset includes 2584 data with 18 categorical and numeric features, which describe the problem of forecasting high energy (higher than 104 J) seismic bumps in a coal mine and is partitioned into two clusters, “high energy seismic bump” and “no high energy seismic bumps”. Clustering a high-energy seismic bump as a no high-energy seismic bump is very hazardous.
- Haberman’s Survival includes 306 data with three integer features from a study on the survival of breast cancer patients who had tolerated surgery. It is partitioned into two groups, “the patient survived five years or more” and “the patient died less than five years”.
- Magic Gamma Telescope includes 19020 data with 11 real features, which use to describe high-energy gamma particles with the imaging technique in a ground-based atmospheric Cherenkov Gamma Telescope (The evidence of energy depositions is usually asymmetric along the major axis.). It is partitioned into two

clusters, single and background. Classifying a background event as a signal is worse than clustering a signal event as the background. Therefore, the simple k-means algorithm is not appropriate for this dataset.

We use k-means and LINEX k-means algorithm with different values of a (as explained in the previous part) to partition data. We choose a according to the accuracy results and the NVI and DB values. The results in Table 6 shows that LINEX k-means algorithm is more accurate, especially when the overestimating and the underestimating are not of the same importance. The k-means and k-median algorithms that are based on symmetric dissimilarity distances are appropriate for data, which the negative and positive errors are of the same importance. However, the LINEX dissimilarity measure is useful in both symmetric and asymmetric cases.

Table 6: The accuracy percentage, NVI, and DB values for k-means, k-median, and LINEX k-means clustering algorithms in Seismic Bumps, Haberman's Survival, and Magic Gamma Telescope datasets.

Dataset	Algorithm	k-means Results	k-median Results	LINEX k-means	
				Results	a
Seismic Bumps	Accuracy	90.91	89.62	93.42	10^{-1}
	NVI	0.985	0.981	1.000	
	DB	0.419	0.501	0.401	
Haberman's Survival	Accuracy	51.18	57.03	72.79	5×10^{-1}
	NVI	0.999	0.983	0.974	
	DB	0.965	1.050	1.401	
Magic Gamma Telescope	Accuracy	0.589	58.91	70.72	10^{-1}
	NVI	0.989	0.998	0.957	
	DB	1.412	1.412	1.742	

6. Conclusion

Sometimes we need an asymmetric dissimilarity measure to calculate the distance between data. We use the LINEX loss function as the dissimilarity measure in k-means clustering algorithm instead of the common measures such as squared Euclidean and Manhattan. So, the centers in each cluster are not the mean (or median) of their entities. To evaluate the accuracy of our algorithm, we partition some real and simulated datasets that their labels and their number of clusters are available and the NVI and the DB indexes help us both to compare the results and to choose a . In some of these datasets, the misclassifying the data into a special cluster may lead to a high loss. Therefore, it is important to use an asymmetric dissimilarity measure to make the different between overestimating and underestimating. Comparing the results, we conclude that LINEX k-means algorithm acts in almost cases well and partitioned the data accurately, especially when classifying an event in a specific cluster is worse (or better).

Acknowledgements

The authors would like to show gratitude to the editor and the reviewers of the article for their comments and suggestions on an earlier version of the manuscript which helped to improve the quality of content of the article.

References

- [1] R. C. Amorim and P. Komisarczuk, On Initializations for the Minkowski Weighted k-means, *Lecture Notes in Computer Science* 7619 (2012), 45-55.
- [2] A. Asuncion and D. J. Newman, UCI Machine Learning Repository <http://www.ics.uci.edu/~mllearn/MLRepository.html>], Irvine, CA: University of California, School of Information and Computer Science, 2007.
- [3] G. H. Ball, D. J. Hall, A Clustering Technique for Summarizing Multivariate Data, *Behavioral Science* 12 (1967), 153-155.
- [4] A. Banerjee and S. Merugu, Clustering with Bregman Divergences, *Journal of Machine Learning Research* 6 (2005), 1705-1749.
- [5] L. Bobrowski and J. C. Bezdek, C-means Clustering with the LL_1 and LL_∞ Norms, *IEEE Transactions on Systems, Man and Cybernetics* 21 (1991), 545–554.
- [6] P. S. Bradley, O. L. Mangasarian and W. N. Street, Clustering via Concave Minimization, *Advances Neural Information Processing Systems*, M. C. Mozer, M. Jordan, and T. Petsche (Eds.) (1997), 368-374.
- [7] C. W. J. Granger, Outline of Forecast Theory Using Generalized Cost Functions, *Spanish Economic Review* 1 (1999), 161-173.
- [8] M. Halkidi, Y. Batistakis and M. Vazirgiannis, Clustering Validity Checking Methods: Part II. *ACM SIGMOD Record*, (2002), 31(3).
- [9] T. J. Harris, Optimal Controllers for Nonsymmetric and Nonquadratic Loss Functions, *Technometrics* 34 (1992), 298–306.
- [10] J. A. Hartigan, *Clustering Algorithms*, Wiley, New York, 1975.
- [11] Z. Huang, Extensions to the k-means Algorithm for Clustering Large Data sets with Categorical Values, *Data Mining and Knowledge Discovery* 2 (1998), 283–304.
- [12] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Prentice-Hall New York reference series, 1988.
- [13] K. Kumamuru, R. Krishnapuram and R. Agrawal, On Learning Asymmetric Dissimilarity Measures, *ICDM '05 Proceedings of the Fifth IEEE International Conference on Data Mining* (2005), 697-700.
- [14] D. S. Modha and W. S. Spangler, Feature Weighting in k-means Clustering, *Machine Learning* 52 (2003), 217-237.
- [15] A. Parsian and S. N. U. A. Kirmani, Estimation under LINEX Loss Function, *Handbook of Applied Econometrics and Statistical Inference* 165 (2002), 53-76.
- [16] R. Reichart and A. Rappoport, The NVI clustering evaluation measure. *In proceedings of the Thirteenth Conference on Computational Natural Language Learning* (2009), 165-173.
- [17] S. Selim and M. Ismail, K-means-type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 (1984), 81-87.
- [18] J. Shao, *Mathematical Statistics*, second edition, Springer, 2003.
- [19] H. R. Varian, A Bayesian Approach to Real Estate Assessment. *Studies in Bayesian Econometrics and Statistics in Honor of Leonard J. Savage* (Eds S. E. Fienberg and A. Zellner), (1975), 195-208.
- [20] A. Zellner, Bayesian Estimation and Prediction Using Asymmetric Loss Functions, *Journal of the American Statistical Association* 81 (1986), 446-451.