Hyperspectral Data Clustering Using Hellinger Divergence

E Myasnikov

1Geoinformatics and Information Security department, Samara National Research University, 34 Moskovskoye shosse, Samara 443086, Russia

E-mail: mevg@geosamara.ru

Abstract. Clustering is an important task in hyperspectral image processing. Despite the existence of a large number of clustering algorithms, little attention has been paid to the use of non-Euclidean dissimilarity measures in the clustering of hyperspectral data. This paper proposes a clustering technique based on the Hellinger divergence as a dissimilarity measure. The proposed technique uses Lloyd's ideas of the k-means algorithm and gradient descent-based procedure to update clusters centroids. The proposed technique is compared with an alternative fast k-medoid algorithm implemented using the same metric from the viewpoint of clustering error and runtime. Experiments carried out using an open hyperspectral scene have shown the advantages of the proposed technique.

1. Introduction

Hyperspectral data clustering is one of the central tasks in hyperspectral data analysis. Clustering techniques are used for thematic interpretation of remote sensing data, form the basis for the segmentation of satellite and aerial photographs [1, 2].

Today there are a large number of data clustering techniques, which can be divided into hierarchical agglomerative clustering methods (single-linkage, complete-linkage, average linkage, etc. [3, 4]), methods based on estimating the parameters of statistical models (for example, EM-algorithm [5]), nonparametric density estimation (for example, DBSCAN [6]), etc.

Probably the most popular clustering technique in many fields, including the analysis of hyperspectral data, remains the k-means algorithm [7], which is based on the proximity to cluster centroids. Unfortunately, the work of this popular method is closely tied to the use of Euclidean distance. The use of the k-medoids algorithm as an alternative technique makes it possible to work with arbitrary dissimilarity measures. But this technique is associated with significantly higher computational costs and a loss of efficiency.

The issue of using alternative dissimilarity measures in the analysis of hyperspectral images was considered when solving problems of classification, search in hyperspectral data, etc. [8, 9]. It has been shown that alternative dissimilarity measures, such as spectral angle [10] and spectral information divergence [11], often have advantages over Euclidean distance when solving these problems.

In a number of cases, the metric properties of dissimilarity measures become crucial when choosing a measure. It turns out, for example, that the solution of the search problem can be performed much faster if the chosen dissimilarity measure possesses the properties of the true metric. Thus, it was shown [9] that the use of Hellinger divergence [12] instead of spectral information divergence [11]...
allows the use of multidimensional indexing structures for fast search of hyperspectral data, which provides a multiple growth in the classification speed with practically indistinguishable quality.

This paper shows how the Hellinger divergence [12] can be used in the construction of a clustering algorithm similar to the k-means Lloyd’s algorithm, as well as how its metric properties make it possible to use it in the fast k-medoid algorithm [13].

The work has the following structure. Section 2 describes the methods used and the developed clustering algorithm. Section 3 is devoted to the experimental study of the developed algorithm and its comparison with the alternative fast k-medoid algorithm. The work ends with a conclusion and a list of references.

2. Methods

2.1. Base k-means algorithm
In the classical algorithm of k-means, the total square deviation of the points contained in the clusters from the centroids is minimized:

$$E = \sum_{k=1}^{K} \sum_{x \in C_k} ||x - c_k||^2$$

where $C_k$ is the set of points included in the k-th cluster, $c_k$ is the centroid of the k-th cluster.

The most common method for solving such a problem, leading to a local minimum of the above error functional, is Lloyd's algorithm [7]. The algorithm starts by initializing the centroids $C_k$, followed by a sequence of iterations, each iterating as follows:

Step 1. The entire set of points is divided into $k$ disjoint subsets $C_k$ in accordance with the proximity to the centroids:

$$C_k = \left\{ x \mid k = \arg\min_{j} \| x - c_j \| \right\}.$$  

Step 2. Cluster centroids are updated

$$c_k = \frac{1}{|C_k|} \sum_{x \in C_k} x.$$  

The algorithm terminates when the clusters stop changing.

As you can see, the above algorithm is built to work in vector spaces with the Euclidean metric, and a simple replacement of the Euclidean distance with another dissimilarity measure when splitting points into clusters will not be correct.

2.2. K-medoids algorithm
In the k-medoids algorithm, data is processed in a similar way. However, instead of centroids, the sampling elements themselves (medoids) are used as representative objects. In a number of tasks, this is more natural. In addition, it allows the use of arbitrary dissimilarity measures and does not impose restrictions on the input space. The algorithm uses an absolute error as an objective function:

$$\varepsilon = \sum_{k=1}^{K} \sum_{x \in C_k} d(x, c_k).$$  \hspace{1cm} (1)$$

Here the cluster medoid $c_k$ is defined as the element that differs least of all from other elements of the cluster $C_k$ in the sense of the average dissimilarity:
Although the k-medoids problem itself is NP complex, the most common iterative Voronoi algorithm [14] is similar to Lloyd's algorithm for the k-means problem described above. An essential problem of the k-medoid algorithm is the actual choice of the medoid for each of the clusters. In contrast to the calculation of the centroid in the k-means algorithm, which is performed in time linear with respect to the number of elements, the direct selection of the medoid requires a quadratic time.

For this reason, several algorithms have been developed to speed up the computations, for example, TOPRANK, TOPRANK2 [15], TRIMED [13]. In the present paper, the last of the listed algorithms [13] is used to calculate the medoid in sub-quadratic time, but it imposes reasonable restrictions on the space: it must be metric.

2.3. Hellinger divergence
It was outlined earlier that classical k-means Lloyd's algorithm operates exclusively on vector spaces with Euclidean distances. Besides, a faster version of the k-medoid technique requires dissimilarity measure to be a metric. As in our study, we were particularly interested in utilizing information theoretical measures as a similarity measure in clustering, we turned our gaze on Hellinger divergence.

The reason behind this choice is that spectral information divergence, which is a well-known measure in hyperspectral image analysis, failed to satisfy the triangle inequality [16], and thus is not a true metric. Meanwhile, Hellinger divergence not only satisfies the requirements (positivity, identity, symmetry, and triangle inequality) but showed the practically indistinguishable quality of classification in our previous research [9] compared to spectral information divergence.

The Hellinger divergence is as follows [12]:

\[
HD(x_i,x_j) = \sqrt{1 - \sum_{k=1}^{M} p_k(x_i)p_k(x_j)},
\]

where \( p_k(x_i) = \frac{x_{ik}}{\sum_{l=1}^{M} x_{il}} \).

This metric (2) can be immediately used in the chosen k-medoid algorithm, but the k-means still needs modifications in order to be used with this metric.

2.4. Modified K-means algorithm for Hellinger divergence
Remind, Lloyd's algorithm consists of two iteratively repeated steps: splitting all the points into disjoint subsets according to the proximity to the centroids and updating the centroids.

Step 1. The modification of the first step is obvious: we just use Hellinger distance as a metric to split points into clusters:

\[
C_k = \left\{ x \mid k = \arg\min_j HD(x,c_j) \right\}
\]

Step 2. In the second step, it is necessary to build the appropriate procedure for updating the centroids of the clusters.

In order to construct the update procedure, we define the centroid \( c \) of the cluster \( C \) as a point in space, for which the total distance from the remaining points of the cluster is minimal in the sense of the chosen metric (2):

\[
c_k = \arg\min_{c} \sum_{x \in C_k} d(x,c).
\]
\[ c = \arg \min_{c'} \sum_{x \in C} HD^2(x, c') \]

In this case, to find the centroid \( c \) of cluster \( C \), we can apply the gradient descent technique consisting of the following steps:

Step 2.a. Initialization of the centroid with \( c(0) \).

Step 2.b. Iterative update the centroid \( c \):

\[ c(t) = c(t-1) - \alpha \nabla \sum_{x \in C} HD^2(x, c), \]

where \( t \) is the number of the iteration, \( \nabla \) is the gradient, \( \alpha \) is the coefficient of the gradient descent.

The partial derivative of the Hellinger divergence (2) with respect to the centroid coordinates is written as

\[ \frac{\partial}{\partial c_n} HD(c, x) = \frac{1}{4 HD(c, x)} \sum_{t=1}^{M} \left( \sum_{k=1}^{M} \frac{p_k(c) p_k(x)}{\sqrt{p_k(c) p_k(x)}} - \frac{p_n(x)}{\sqrt{p_n(c) p_n(x)}} \right). \]

This gives the following recurrence relation for finding the centroid \( c \):

\[ c_n(t) = c_n(t-1) - \alpha \sum_{x \in C} \frac{1}{4 HD(c(t-1), x)} \sum_{t=1}^{M} \left( \sum_{k=1}^{M} \frac{p_k(c(t-1)) p_k(x)}{\sqrt{p_k(c(t-1)) p_k(x)}} - \frac{p_n(x)}{\sqrt{p_n(c(t-1))}} \right). \]

It is worth mentioning that we use the same approach as in the kmeans++ algorithm [17] to initialize the centroids at the beginning of the modified algorithm.

In our implementation of the proposed technique, we used the gradient descent algorithm with error control, which adaptively reduced the coefficient \( \alpha \) by the factor of 2 in the case of raised error. We used the absolute difference of the error as a stopping criterion and stopped the optimization when this difference fell below the small predefined value (10⁻⁸). Besides, we limited the maximum number of iterations with the \( \text{MaxIter} \) value, which was a parameter of the algorithm.

3. Experiments

To evaluate the algorithms described in the previous section, we used probably the most well-known open hyperspectral Indian Pines scene [18]. All algorithms were implemented in C++. The research was carried out on a laptop based on an Intel Core i7-6500U @ 2.3 GHz processor.

Due to the limited scope, we compare the proposed technique with its k-medoids alternative from the viewpoint of the clustering error (3) and time characteristics:

\[ E = \sum_{k=1}^{K} \sum_{x \in C_k} HD^2(X, c_k). \]

In our first experiment, we examined how the maximum number of iterations \( \text{MaxIter} \) in the gradient descent algorithm affects the final clustering quality (3). To accomplish this we varied the maximum number of iterations in the centroid update algorithm (Step 2.b) and measured the quality of clustering after updating all centroids. The results are shown in Figure 1.

Each graph the Figure 1.a correspond to a particular value of the \( \text{MaxIter} \) parameter (here we present only results for 3, 20, 50, and 100) and shows the dependence of the clustering quality \( E \) on the number \( t \) of the iterations in the modified Lloyd’s algorithm described in Section 2.4. As it can be seen, greater values of \( \text{MaxIter} \) leading to more precise centroid updates provided lower errors and
better clustering quality what is expected. The average iteration time of the modified Lloyd’s algorithm grows with MaxIter almost linearly (see Figure 2.b).

Figure 1. The influence of the MaxIter parameter on the clustering quality $E$ (a) and iteration time (b) for the proposed technique (averaged over 20 runs).

In our second experiment, we compared the proposed clustering technique with the k-medoids technique based on the TRIMED algorithm and operating with the Hellinger divergence (see Section 2.2). Figure 2 shows the dependence of the clustering quality $E$ on the number $t$ of the iterations in Lloyd’s algorithm for both the proposed and k-medoids technique. In this figure, the left part shows the error value averaged over 30 runs of the algorithms, while the right part shows the best (minimum) values over 30 runs. In both cases, we can see that the proposed approach provided better results regardless of the number of iterations.

In terms of time, our TRIMED implementation of the k-medoids required about 0.55 sec. per iteration that roughly corresponds to the MaxIter=20 setting of the proposed technique (0.67 sec). At this setting, the proposed algorithm does not provide the best quality but it still outperforms the k-medoids algorithm.

Figure 2. The dependence of the clustering quality for the k-medoid (Trimed) and proposed technique on the number of iterations: averaged error (a) and minimum error (b) over 30 runs.

It is worth noting that the k-medoids uses objective function (1) different from (3) but all said above is true with respect to the objective (1). In particular, after 100 iterations the objective (1) was equal to $e_{k\text{-med}} = 0.0015$ and $e_{prop} = 0.0012$ for the k-medoids and proposed technique correspondingly.
In all cases, we reported results for ten clusters due to volume limitations. As an example, we provide the results for twenty clusters after 100 iterations: $E_{k,med} = 0.0331$ and $E_{k,med} = 0.00137$ for k-medoids against $E_{prop} = 0.0295$ and $E_{prop} = 0.00105$ for the proposed technique.

4. Conclusions
In this paper, we proposed the clustering technique based on the Hellinger divergence as a dissimilarity measure. The proposed technique uses Lloyd's ideas of the k-means algorithm and gradient descent-based procedure to update clusters centroids.

The proposed technique is compared with an alternative fast k-medoid algorithm TRIMED implemented using the Hellinger divergence and exploiting its metric properties. Experiments carried out using the Indian Pines hyperspectral scene have shown the advantages of the proposed technique in terms of clustering error.

In the future, it is planned to conduct a study of clustering algorithms built with alternative metrics.

5. References
[1] Myasnikov E V 2017 Hyperspectral image segmentation using dimensionality reduction and classical segmentation approaches Computer Optics 41(4) 564-72
[2] Zimichev E A, Kazanskiy N L and Serafimovich P G 2014 Spectralspatial classification with k-means++ particlional clustering Computer Optics 38(2) 281-6
[3] Spáth H 1980 Cluster Analysis Algorithms (Chichester: Ellis Horwood)
[4] Everitt B 2011 Cluster analysis (Chichester, West Sussex, U.K: Wiley)
[5] Dempster A P, Laird N M and Rubin D B 1977 Maximum likelihood from incomplete data via the EM algorithm Journal of the Royal Statistical Society B 39(1) 1-38
[6] Ester M, Kriegel H-P, Sander J and Xu X 1996 A density-based algorithm for discovering clusters in large spatial databases with noise Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96) (AAAI Press) pp 226-31
[7] Lloyd S P 1982 Least squares quantization in PCM IEEE Transactions on Information Theory 28(2) 129–37, doi:10.1109/TIT.1982.1056489
[8] Van der Meer F 2006 The effectiveness of spectral similarity measures for the analysis of hyperspectral imagery International Journal of Applied Earth Observation and Geoinformation 8(1) 3-17
[9] Myasnikov E 2021 Nearest neighbor search in hyperspectral data using binary space partitioning trees IEEE Xplore Digital Library, Proc. 11th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS) pp 1-4, doi: 10.1109/WHISPERS52202.2021.9484041
[10] Kruse F A, Boardman J W, Lefko A B, Heidebrecht K B, Shapiro A T, Barloon P J and Goetz A F H 1993 The Spectral Image Processing System (SIPS) - Interactive Visualization and Analysis of Imaging Spectrometer Data Remote Sensing of Environment 44 145-163
[11] Chang C-I 2013 Hyperspectral Data Processing: Algorithm Design and Analysis (John Wiley & Sons)
[12] Hellinger E 1909 Neue begründung der theorie quadratischer formen von unendlichvielenveränderlichen Journal für die reine und angewandte Mathematik 136 210–71
[13] Newling J and Fleuret F 2017 A Sub-Quadratic Exact Medoid Algorithm PMLR 54 185-93
[14] Hastie T J, Tibshirani R J and Friedman J H 2001 The elements of statistical learning : data mining, inference, and prediction. Springer series in statistics (New York: Springer)
[15] Okamoto K, Chen W and Li X-Y 2008 Ranking of closeness centrality for large-scale social networks Proceedings of the 2nd Annual International Workshop on Frontiers in Algorithmics (FAW ’08) pp 186-95 (Berlin, Heidelberg: Springer Verlag)
[16] Abou-Moustafa K T, Ferrie F P 2012 Note on metric properties for some divergence measures: the gaussian case JMLR 25 1-15
[17] Arthur D and Vassilvitskii S 2007 K-means++: the advantages of careful seeding. Proceedings
of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'07) pp 1027-35. DOI: 10.1145/1283383.1283494

[18] Baumgardner M F, Biehl L L and Landgrebe D A 2015 220 band AVIRIS hyperspectral image data set: june 12, 1992, Indian pine test site 3 Purdue University Research Repository