ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2019.0265

ROOT MAPPING BASED NEIGHBOUR CLUSTERING IN HIGH-DIMENSIONAL DATA

M.D. Dithy[1] and V. KrishnaPriya[2]

[1]Department of Computer Science, Sri Ramakrishna College of Arts and Science for Women, India
[2]School of Computing, Sri Ramakrishna College of Arts and Science, India

Abstract

High-dimensional data arise naturally in a lot of domains, and have regularly presented a great confront for usual data mining techniques. This work, take a novel perspective on the problem of data points (data in the orientation of contain points) in clustering large-dimensional data. The planned methodology known as root mappings and neighbor clustering, that takes as input measures of correspondence between pairs of information points. Real-valued data points are exchanged between data points until a high-quality set of patterns and corresponding clusters gradually emerges. To validate our theory by demonstrating that data points is a high-quality measure of point centrality within a high-dimensional information cluster, and by proposing several clustering algorithms, showing that main data points can be used effectively as cluster prototypes or as guides during the search for centroid-based cluster patterns. Experimental results demonstrate the good performance of our proposed algorithms in manifold settings, mainly focused on large quantities of overlapping noise. The proposed methods are modified mostly for detecting approximately hyper spherical clusters and need to be extended to properly handle clusters of arbitrary shapes.

Keywords:

Clustering, High-Dimensional, Nearest Neighbours, Data Points, Root Mapping

1. INTRODUCTION

Clustering, in general, is an unsupervised process of grouping elements together, so that elements assigned to the same cluster are more similar to each other than to the remaining data points [1]. This goal is often difficult to achieve in practice. Over the years, various clustering algorithms have been proposed, which can be roughly divided into four groups: partitioned, hierarchical, density based [17] [18], and subspace algorithms. Algorithms from the fourth group search for clusters in some lower dimensional projection of the original data and have been generally preferred when dealing with data that are high dimensional [2]-[5].

The motivation for this preference lies in the observation that having more dimensions usually leads to the so-called curse of dimensionality, where the performance of many standard machine-learning algorithms becomes impaired. The difficulties in dealing with high-dimensional data are omnipresent and abundant. However, not all phenomena that arise are necessarily detrimental to clustering techniques. This paper that data points, which is the tendency of some data points in high-dimensional data sets to occur much more frequently in k-nearest neighbor lists of other points than the remainder of the points from the set, will indeed be used for agglomeration. To our data, this has not been antecedent tried. In a limited sense, data points in graphs have been used to represent typical word meanings in [6], which were not used for data clustering.

Our current focus was mostly on properly selecting cluster prototypes, with the proposed methods tailored for detecting approximately outlier spherical clusters. The objective of this work is to develop an efficient high dimensional data clustering with root mappings in synthetic and real world datasets.

The rest of the paper is organized as follows: Related work is detailed in section 2. In section 3, proposed methodologies perform an efficient feature selection and neighbor clustering algorithm process. In section 4 describes an experimental results and finally conclusion is in section 5.

2. RELATED WORK

2.1 DENSITY BASED CLUSTERING

Density primarily based agglomeration [8] differentiates regions that have a higher density than its neighborhood associate degree doesn't would like the number of clusters as an input parameter. Regarding a termination condition, two parameters indicate once the enlargement of clusters ought to be terminated: given the radius of the amount of information points to seem for a minimum the number of points for the density calculations must be exceeded. Local scaling may be a technique that makes use of the native statistics of the info once distinguishing clusters. This is done by scaling the distances around each point in the dataset with a factor proportional to its distance to its k^{th} nearest neighbor. Locally scaled density primarily based agglomeration algorithmic program clusters points by connecting dense regions of the area till the density falls below a threshold determined by the middle of the cluster. In high-dimensional spaces, this is often not easy to estimate, due to data being very sparse. There is also the issue of choosing the proper neighborhood size, since both small and large values of k can cause problems for density-based approaches [9].

2.2 K-MEANS++

The *K-means++* is a specific way of choosing centres for the *k*-means algorithm. The relationship between *k-means++* clustering and data points was briefly examined in [10], where it was observed that data points may not cluster well using conventional prototype-based clustering algorithms (*K-means* ++) [7], since they not only tend to be close to points belonging to the same cluster (i.e., have low intra-cluster distance) but also tend to be close to points assigned to other clusters (low inter-cluster distance). The demonstrable gains of k-means++ over random initialization is precisely in the constantly updated non-uniform selection. The algorithm that works in a small number of iterations, selects more than one point in each iteration but in a non-uniform manner, and has provable approximation guarantees. Data points can, therefore, be viewed as (opposing) analogues of outliers, which have high inter- and intra-cluster distance, suggesting that data points should also receive special attention [10].

3. PROPOSED SYSTEM

The proposed method identifies the patterns among data points and forms clusters of data points around these patterns. It operates by at the same time considering all information as potential patterns and exchanging messages between knowledge points till an honest set of patterns and clusters emerges. The root mapping and neighbor cluster are used to find the fitness value data points are exchanged between data points until a high-quality set of patterns and corresponding clusters gradually emerges.

3.1 FEATURE SELECTION

A “feature” or “attribute” or “variable” refers to a portion of the data points. Typically before collecting data, features are specified or preferred. Features can be discrete, continuous, or insignificant. Feature selection for high-dimensional data clustering is the task of disregarding irrelevant and redundant terms in the vectors that represent the data points, aiming to and the smallest subset of terms that reveals “natural” clusters of data points. Searching for the small subset Fig.1 of relevant terms will speed up the clustering process while avoiding the curse of dimensionality.

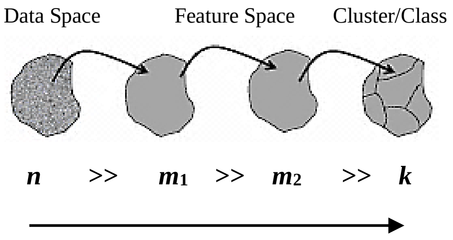

Fig.1. Dimensionality Reduction

The unconnectedness filter removes extraneous options employing a changed kind of the relief algorithmic program that assigns relevancy values to options by treating coaching samples as points in feature house. For each sample, it finds the nearest “hit” (another sample of the same class) and “miss” (a sample of a different class), and adjusts the significance value of each features in step with the square of the feature distinction between the sample and therefore the hit and miss. Irrelevance Filter feature selection methods evaluate attributes prior to the learning process, and without specific reference to the clustering algorithm that will be used to generate the final result. The filtered dataset may then be used by any clustering algorithms.

3.2 CORRELATION OF ROOT MAPPING TO DATA CLUSTERS

A correlation between low data points elements and outliers was also observed. A low-points score indicates that a point is on average far from the rest of the points and hence probably an outlier. In high-dimensional spaces, however, low data point elements are expected to occur by the very nature of these spaces and data resource. The root mapping can be applied using more general notions of similarity, and the similarities may be positive or negative. The output of the algorithmic program is unchanged if the similarities area unit scaled and/or offset by a relentless (as long because the preferences area unit scaled and/or offset by constant). To compute fitness measure over the set of possible clusters and then chooses among the set of cluster candidates points those that optimize the measure used. To identify the cluster of a specific vertex or to group all of the vertices into a set of clusters, and then present possible cluster fitness measures that serve for ways that turnout the bunch by scrutiny totally different groupings and choosing one that meets or optimizes a particular criterion. The ratio of the cluster is to minimum sums of degrees either within the cluster or outside it. A fitness function is evaluated for all neighbours and the outcome is used to choose to which neighbour the search will proceed.

3.3 NEIGHBOUR CLUSTERING ALGORITHM

The neighbour clustering algorithm works message passing among data points. Each data points receive the availability from other data points (from a pattern) and send the responsibility message to others data points (to a pattern). Sum of responsibilities and availabilities for data points identify the cluster patterns. The high-dimensional data point availabilities *A* (*i*, *k*) are zero: A(*i*, *k*) = 0, *R* (*i*, *k*) is set to the input similarity between point *i* and point *k* as its pattern, minus the largest of the similarities between point *i* and other candidate patterns.

The cluster responsibilities are computed using the equation,

$$R(i,k) \leftarrow S(i,k) - \max_{k's\cdot t\cdot k' \neq k} \left\{ A(i,k') + S(i,k') \right\} \tag{1}$$

In later iterations, when some data points are effectively assigned to other patterns, their availabilities will drop below zero. These negative availabilities will decrease the effective values of some of the input similarities S (*i*, *k*′) in the above rule, removing the corresponding candidate from the competition. The above responsibility in equation (1) is updated that let all data point patterns to get competed for ownership of a data point, the following availability update gathers confirmation from data points as to whether each data would make a good pattern:

$$A(i,k) \leftarrow \min\left\{ 0, R(k,k) + \sum_{i's\cdot t\cdot i' \in \{i,k\}} \max\left\{0, R(i',k)\right\} \right\} \tag{2}$$

The data links are sent from cluster members (data points) to candidate patterns (data points), indicating how well-suited the data point would be as a member of the candidate pattern cluster. The rot mapping and neighbour clustering iteratively computes data responsibilities and data availabilities to overcome the outlier points. The algorithm terminates if decisions for the patterns and the cluster boundaries are unchanged for convict’s iterations, or if maximum iterations are reached. The responsibilities and availabilities are messages that provide evidence for whether or not each data point should be in data points and if not to what outlier that data point should be assigned.

Algorithm 1: Neighbour Clustering Algorithm

Input: *A*, *R*, *i*, *k*

Step 1: Initialize $A(i,k) = 0$, $R(i,k) = 0$, $k = 0$, and $S(i,k) = 0$ randomly

Step 2: repeat

Step 3: Update the data point responsibility by Eq.(1) where $S(i,k)$ is the similarity of data points and root map pattern k.

Step 4: Update the data point availabilities by Eq.(2)

Step 5: Update self-availability by using Eq.(3)

Step 6: Compute $sum = A(i,k) + R(i,k)$ for data point i and find the value of k that maximizes the sum to identify the data points.

Step 7: If outlier points do not change for a fixed number of iterations go to Step 7 else go to Step 1.

4. EXPERIMENTAL RESULTS

The proposed root mapping with neighbour clustering algorithm on Real-world data is usually much more complex and difficult to cluster; therefore such tests are of a higher practical significance. As not all data exhibit data points, the algorithms are tested both on intrinsically high-dimensional, high-data points and intrinsically low-to-medium dimensional, low-data. There were two different experimental setups. In the first setup, a single data set was clustered for many different K-s (number of clusters), to see if there is any difference when the number of clusters is varied. In the second setup, 20 different data sets were all clustered by the number of classes in the data (the number of different labels).

The agglomeration quality in these experiments was measured by 2 quality indices, the silhouette index and also the isolation index [11]-[16], that measures a proportion of k-neighbour points that area unit clustered along. In the experimental setup, the two-part Miss-America data set (cs.joensuu.fi/sipu/datasets/) is used for evaluation. Each part consists of 6,480 instances having 16 dimensions. Results were compared for numerous predefined numbers of clusters in formula calls. Each algorithm was tested 50 times for each number of clusters. Neighbourhood size was 5. The highest level of noise for which we tested was the case when there was an equal number of actual data instances in original clusters and noisy instances. At every noise level, RMNC (root map with neighbour cluster), KM++, GHPC, and Global Hubness-Proportional *k*-Means (GHPKM) were run 50 times each.

Table.1. Clustering Quality of Silhouette index on the Miss-America Data Set

K	2	4	6	8	10	12	14	16
RMNC	0.59	0.42	0.31	0.28	0.19	0.17	0.13	0.1
GHPC	0.38	0.29	0.25	0.21	0.15	0.10	0.10	0.09
KM++	0.14	0.12	0.09	0.08	0.07	0.07	0.07	0.07
GHPKM	0.28	0.18	0.17	0.14	0.13	0.11	0.10	0.08

The results for both parts of the data set are given in Table.1 and Table.2. The root map and neighbour cluster (RMNC) is clearly outperformed GHPC, KM and other data-based methods. This shows that hubs will function smart cluster center prototypes.

Table.2. Clustering Quality of Isolation index on the Miss-America Data Set

K	2	4	6	8	10	12	14	16
RMNC	0.94	0.92	0.79	0.58	0.51	0.49	0.36	0.29
GHPC	0.91	0.89	0.71	0.53	0.42	0.33	0.30	0.26
KM++	0.62	0.46	0.34	0.23	0.19	0.16	0.13	0.12
GHPKM	0.85	0.54	0.45	0.38	0.29	0.26	0.24	0.23

5. CONCLUSION

The proposed method of RMNC method had proven to be more robust than the GHPKM and K-Means++ baseline on both synthetic and real-world data, as well as in the presence of high levels of artificially introduced noise. The root map with neighbour clustering can easily be extended to incorporate additional pair-wise constraints such as requiring points with the same label to come into view in the same cluster with simply anadditional layer of performing of performing hubs. The model is versatile enough for data aside from express constraints like two points being in several clusters or perhaps higher order constraints (e.g., 2 of 3 points should be in the same clusters.

REFERENCES

[1] J. Han, M. Kamber and J. Pei, "*Data Mining: Concepts and Techniques*", 2nd Edition, Morgan Kaufmann, 2006.

[2] C.C. Aggarwal and P.S. Yu, "Finding Generalized Projected Clusters in High Dimensional Spaces", *Proceedings of 26th ACM International Conference on Management of Data*, pp. 70-81, 2000.

[3] K. Kailing, H.P. Kriegel, P. Kroger and S. Wanka, "Ranking Interesting Subspaces for Clustering High Dimensional Data", *Proceedings of 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 241-252, 2003.

[4] K. Kailing, H.P. Kriegel and P. Kroger, "Density-Connected Subspace Clustering for High-Dimensional Data", *Proceedings of 4th SIAM International Conference on Data Mining*, pp. 246-257, 2004.

[5] E. Muller, S. Gunnemann, I. Assent and T. Seidl, "Evaluating Clustering in Subspace Projections of High Dimensional Data", *Proceedings of International Conference on Very Large Data Base Endowment*, Vol. 2, pp. 1270-1281, 2009.

[6] E. Agirre, D. Martinez, O.L. De Lacalle and A. Soroa, "Two Graph-Based Algorithms for State-of-the-Art WSD", *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, pp. 585-593, 2006.

[7] D. Arthur and S. Vassilvitskii, "K-Means++: The Advantages of Careful Seeding", *Proceedings of International Conference on* ACM-SIAM *SIAM International Conference on Discrete Algorithms*, pp. 1027-1035, 2007.

[8] E. Bicici and D. Yuret, "Locally Scaled Density Based Clustering", *Proceedings of International Conference on*

Adaptive and Natural Computing Algorithms, pp. 739-748, 2007.

[9] S. Hader and F.A. Hamprecht, "Efficient Density Clustering using Basin Spanning Trees", *Proceedings of International Conference on Data Science and Applied Data Analysis*, pp. 39-48, 2003.

[10] M. Radovanovic, A. Nanopoulos and M. Ivanovic, "Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data", *Journal of Machine Learning Research*, Vol. 11, pp. 2487-2531, 2010.

[11] G. Frederix and E.J. Pauwels, "Shape-Invariant Cluster Validity Indices", *Proceedings of 4th International Conference on Data Mining*, pp. 96-105, 2004.

[12] Y. He, H. Tan, W. Luo, H. Mao, S. Feng and J. Fan, "MR-DBSCAN: An Efficient Parallel Density-based Clustering Algorithm using MapReduce", *Proceedings of International Conference on Parallel and Distributed Systems*, pp. 473-480, 2011.

[13] C. Cassisi, A. Ferro, R. Giugno, G. Pigola and A. Pulvirenti, "Enhancing Density-Based Clustering: Parameter Reduction and Outlier Detection", *Information Systems*, Vol. 38, No. 3, pp. 317-330, 2013.

[14] D. Moulavi, P. A Jaskowiak, R.J.G. B. Campello, A. Zimek and J. Sander, "Density-Based Clustering Validation", *Proceedings of 4th SIAM International Conference on Data Mining*, pp. 839-847, 2014.

[15] R. Guidotti, R. Trasarti and M. Nanni, "Tosca: Two-Steps Clustering Algorithm for Personal Locations Detection", *Proceedings of International Conference on Advances Geographic Information Systems*, 18-38, 2015.

[16] Y. Lv, T. Ma, M. Tang, J. Cao, Y. Tian, A.A. Dhelaan and M.Z. Rodhaan, "An Efficient and Scalable Density-based Clustering Algorithm for Datasets with Complex Structures", *Neurocomputing*, Vol. 171, pp. 9-22, 2016.

[17] J. Gan and Y. Tao, "On the Hardness and Approximation of Euclidean DBSCAN", *ACM Transactions on Database Systems*, Vol. 42, No. 3, pp.1-14, 2017.

[18] Avory Bryant and Krzysztof Cios, "RNN-DBSCAN: A Density-Based Clustering Algorithm Using Reverse Nearest Neighbor Density Estimates", *IEEE Transactions on Knowledge And Data Engineering*, Vol. 30, No. 6, pp. 1109-1121, 2018.