A new Distance Function for Prototype based Clustering Algorithms in High Dimensional Spaces

Roland Winkler and Frank Klawonn and Rudolf Kruse

Abstract High dimensional data analysis poses some interesting and counter intuitive problems. One of this problems is, that some clustering algorithms do not work or work only very poorly if the dimensionality of the feature space is high. The reason for this is an effect called distance concentration. In this paper, we show that the effect can be countered for prototype based clustering algorithms by using a clever alteration of the distance function. We show the success of this process by applying (but not restricting) it on FCM. A useful side effect is, that our method can also be used to estimate the number of clusters in a data set.

Key words: curse of dimensionality, distance concentration, prototype based clustering, fuzzy c-means

1 Introduction

The curse of dimensionality for clustering can be best described by means of distance concentration. Beyer et al. [1], introduced the effect of distance concentration for nearest neighbour queries. They showed that a nearest neighbour query is not meaningful if the relative variance of distances to other data objects converges to 0. In other words: the difference between the nearest and furthest data object becomes negligible with increasing dimensionality. Durrant and Kabán [5] expanded the argumentation by showing that the implication in Bayer et al.'s paper is indeed

Roland Winkler
German Aerospace Center Braunschweig e-mail: roland.winkler@dlr.de

Frank Klawonn
Ostfalia, University of Applied Sciences e-mail: f.klawonn@ostfalia.de

Rudolf Kruse
Otto-von-Guericke University Magdeburg e-mail: kruse@iws.cs.uni-magdeburg.de

an equivalence. Since clustering is the task to find meaningful structure solely by analysing the spacial distribution of data objects, the results of Beyer et al. and Durrant and Kabán are relevant for all clustering algorithms in high-dimensional feature spaces. Distance concentration is especially a problem if relations of distances are analysed as it is the case for FCM and other prototype based clustering algorithms.

In this paper, we present a distance function that counters the effect of distance concentration. Our approach does not only counter the effect of distance concentration, it also presents a solution for the problem of finding the correct number of clusters which is a specific problem for prototype based clustering algorithms.

The paper is structured as follows. In the next section, the effect of distance concentration is defined. In section 4 the new distance function is presented and in the following section 5 the clustering algorithm is developed. We apply the proposed algorithm and several others on a data set of aircraft movements in Section 6. Finally, this paper ends with the conclusions and references in Section 7.

2 Distance Concentration

Let $X \subset \mathbb{R}^m$ be a finite set of m-dimensional real data objects, i.i.d. sampled from some unknown probability distribution F_X in $\mathbb{R}^m$. Let $p > 0$ be a constant, $Q \in \mathbb{R}^m$ be an arbitrary sample point, $\|\cdot\| : \mathbb{R}^m \longrightarrow \mathbb{R}$ a metric and $D_X^p(Q) = \{\|x-Q\|^p \,:\, x \in X\}$ be the set of distances, from the viewpoint of Q. Let $\overline{E}(D_X^p(Q))$ be the mean (sample expectation value) and $\overline{V}(D_X^p(Q))$ the sample variance of $D_X^p(Q)$. Then

$$\overline{RV}(D_X^p(Q)) = \frac{\overline{V}(D_X^p(Q))}{\overline{E}^2(D_X^p(Q))}$$

is called the sample relative variance of $D_X^p(Q)$.

Formally, distance concentration occurs if for a sequence of probability distributions F_m and resulting sequences of data sets X_m and query points Q_m holds:

$$\lim_{m\to\infty} \overline{RV}(D_{X_m}^p(Q_m)) \to 0.$$

Or in other words, the relative variance of distances becomes negligible.

The occurrence of distance concentration depends on the norm $\|\cdot\|$ and the distribution F_X. Let here, $\|\cdot\|$ be one of the $\mathscr{L}_p$ norms with $p \geq 1$. A result from Hinneburg et. al [6] shows that distance concentration can only occur for norms with $p > 1$, which means only the Manhattan distance $\mathscr{L}_1$ norm is stable. If a norm is unstable, distance concentration can occur for a wide range of data set distributions [1]. For example for an m-dimensional normal distribution with i.i.d. dimensions: $F_m = (\mathscr{N}(1,0),\ldots,\mathscr{N}(1,0))^\perp$ with $\perp$ denoting the transposed vector and $\mathscr{N}(1,0)$ denoting the 1-dimensional standard normal distribution. Also more complex distributions like a uniform distribution on the hypercube surface (no pair of dimensions

are independent given any subset of other dimensions) is suffering from distance concentration.

There are two problems with this probability theory result in clustering applications. First, no data set is really going to have an infinite number of features. Second, distance concentration might not occur for the data set it self as it is supposed to be clumped up into several clusters, otherwise clustering would not make any sense in the first place. However, even if the relative variance of distances of a given data set is not 0, clustering algorithms still have their problems because the probability distribution F_X is not known in advance and the clumping effect of clusters might be too weak for the algorithm to recognise. Especially for fuzzy prototype based clustering algorithms this is a problem because they tend to evaluate relative distances in order to assign fuzzy values.

3 Distance Concentration and FCM type Clustering Algorithms

Let $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^m$ be a m-dimensional data set with n data objects, $Y = \{y_1, \ldots, y_c\} \subset \mathbb{R}^m$ a set of c prototypes, $\|\cdot\| = \mathscr{L}_2$ the euclidean metric, $1 < \omega \in \mathbb{R}$ the fuzzifier and $U \in [0,1]^{c \times n}$ the membership matrix with $u_{ij} \in [0,1]$ as elements subjective to $1 = \sum_{i=1}^{c} u_{ij}$. The symbol $d_{ij} = \|y_i - x_j\|$ denotes the distance between a data object and a prototype with $\|\cdot\| = \mathscr{L}_2$ being the euclidean distance. The fuzzy c-means algorithm [4, 2] is defined by minimizing the objective function with Lagrange multipliers $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$:

$$J_{\mathrm{FCM}}(X, Y, U, \Lambda) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{\omega} d_{ij}^2 - \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{c} u_{ij} - 1 \right). \tag{1}$$

The objective function is minimized using the alternative optimization algorithm which iteratively optimizes the prototype locations Y and membership values U. The update equations for defining the next iteration $(t+1)$ from the current iteration (t) with the time variable $t \in \mathbb{N}$ are

$$u_{ij}^{(t+1)} = \frac{\left(\frac{1}{d_{ij}^t} \right)^{\frac{2}{\omega - 1}}}{\sum_{k=1}^{c} \left(\frac{1}{d_{ij}^t} \right)^{\frac{2}{\omega - 1}}} \quad \text{and} \quad y_i^{t+1} = \frac{\sum_{j=1}^{n} \left(u_{ij}^{t+1} \right)^{\omega} x_j}{\sum_{j=1}^{n} \left(u_{ij}^{t+1} \right)^{\omega}}. \tag{2}$$

When FCM is applied on a high-dimensional data set, this update rule becomes problematic. It starts with the initialization, the initial positions $Y^0 = \{y_1^0, \ldots, y_c^0\}$ of prototypes must be somehow determined. A sample of prototype positions as a subset of the data set, $Y \subset X$, is usually not a good idea as this almost guaranties that not all clusters are found. Therefore, Y^0 is usually sampled from some distribution

F_{init} of the feature space, for example a uniform distribution on the smallest data set enclosing hyperrectangle. From the view point of the data object ($Q = x_j$), according to the last section, all distances to the members of a sample of a probability distribution F_{init}, like Y^0, becomes equal. Formally, let $Q = x_j \in X$, for an $1 \leq j \leq n$, then

$$d_j^* = \overline{E}(D_Y(x_j)) \approx \|y - x_i\|, \forall y \in Y. \tag{3}$$

This has very bad implications on the performance of FCM. Especially because the distances to the prototypes w.r.t. to a data object are not evaluated by their absolute value, but by their relative value to one another. Following from equation (3):

$$u_{ij} \approx \frac{\left(\frac{1}{d_j^*}\right)^{\frac{2}{\omega-1}}}{\sum_{k=1}^{c}\left(\frac{1}{d_j^*}\right)^{\frac{2}{\omega-1}}} = \frac{\left(\frac{1}{d_j^*}\right)^{\frac{2}{\omega-1}}}{c \cdot \left(\frac{1}{d_j^*}\right)^{\frac{2}{\omega-1}}} = \frac{1}{c}; \quad y_i \approx \frac{\sum_{j=1}^{n}\left(\frac{1}{c}\right)^{\omega} x_j}{\sum_{j=1}^{n}\left(\frac{1}{c}\right)^{\omega}} = \frac{\sum_{j=1}^{n}\left(\frac{1}{c}\right)^{\omega} x_j}{n \cdot \left(\frac{1}{c}\right)^{\omega}} = \frac{1}{n}\sum_{j=1}^{n} x_j.$$

All prototypes are updated to a position, close to the centre of gravity of the data set X (for experimental proof, see section 6). Our previous work [8] shows, that this can only be prevented by initializing the prototypes near the clusters in X which would increase the variance in $D_Y(x_j)$ for all data objects of a cluster with a prototype nearby. The probability that all or at least most prototypes are initialized near a cluster is almost 0 because the hypervolume of the space near the clusters is very small, compared to the complete relevant feature space. This means that either the distribution of data objects F_X has to be known in advance, which is usually not the case. Or another clustering algorithm must be used to determine the initial location of the prototypes, which would make the application of FCM unnecessary. Also the question is, if there is an other, reliable clustering algorithm for high-dimensional data.

It should be noted that also the EM algorithm and other FCM related algorithms like noise clustering [3] and in fact most prototype based fuzzy type algorithms are effected by the same problem. Hierarchical and density based clustering algorithms are usually also not a good choice because the distances between clusters tend to be similar to the distances of data objects of one cluster which prevents a 'natural' choice of cutting the cluster hierarchy. Therefore, it is hard to determine the natural number of clusters or in general parameters necessary to run these algorithms.

4 Alternative Distance Function

In the last section, we determined that it is almost impossible to initialize FCM in a high-dimensional space in such a way that prototypes find a cluster. The idea is, to adjust the distance function according to the new circumstances. Hsu and Chen [7] proposed a new distance function (which is not a norm):

$$SDP(x,y) = \sum_{k=1}^{dim} \omega_k f_{s_{k1},s_{k2}}|x_k - y_k| \text{ and } f_{s_{k1},s_{k2}}(x) = \begin{cases} 0 & if \quad x < s_{k1} \\ x & if \quad s_{k1} < x < s_{k2} \\ e^x & if \quad s_{k2} < x \end{cases}$$

As this function is very useful and versatile, it contains many $(3 \cdot dim)$ parameters that are difficult to set. An other problem is, that the update equation for the prototypes in FCM must be solvable for the prototype location, which is not the case for most unusual norms as for example the SDP function. We propose an alternative distance function that is also useful for clustering purposes.

The reason FCM does not work very well is, that the distances have not enough contrast to be useful for assigning membership values. So the goal is to increase the contrast in distance values but leaving the update equation for the prototypes solvable for the prototype location. The *DCR* (**D**istance **C**oncentration **R**esistant) function is defined for a distance correction value $\delta \geq 0$:

$$DCR_\delta(x,y) = \|x-y\|^2 - \delta$$

This function is not a norm because its value can be less than 0. However, this function is very useful for replacing the distance function in FCM. With the parameter δ, it is possible to increase the contrast in distance values and $\nabla_y DCR_\delta(x,y) = \nabla_y \|x-y\|^2$ because δ is a constant value.

5 FCM with DCR as Distance Function

In the objective function of FCM, the distance function d_{ij} is replaced with $DCR_{ij} = DCR_{\delta_i}(x_j, y_i)$:

$$J_{\text{DCRFCM}}(X,Y,U,\Lambda) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{\omega} DCR_{ij} - \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{c} u_{ij} - 1 \right). \tag{4}$$

With the parameters δ_i, $i = 1,\ldots,c$ it is possible to adjust the distance values in such a way, that the effect of distance concentration in high dimensions is nullified. The cleanest approach would be, to set $\delta_i = \min(D_X^2(y_i))$, because this way, all distances would remain positive or equal to 0. However, practical tests have shown that this is not enough, the prototypes would get stuck on randomly scattered noise data objects.

We use a more radical approach. For a parameter $\alpha \in \mathbb{R}$, $\alpha > 0$, set $\delta_i = \max\{0, \overline{E}(D_X^2(y_i)) - \alpha \cdot \overline{V}(D_X^2(y_i))\}$. So the distance reduction value δ_i is set to the mean of distances (from the point of view of the prototype), reduced by α times the sample variance of the distances. A value of $\alpha = 3$ is usually a good choice because the Cantelli inequality (one sided Chebyshev's inequality) guarantees that at most 10% of the data objects are closer to y_i than δ_i. That however implies that there might be negative distance values. That is not a problem for the objective

function as its actual value is not important. For updating the membership values however, the condition of $u_{ij} \geq 0$ must be ensured using the Karush-Kuhn-Tucker multiplier. Because that is computationally difficult, the condition is satisfied manually: if $DCR_{ij} < 0$, the corresponding membership value is set to $u_{ij} = 1$. If there are k prototypes with negative DCR value, the corresponding membership values are set to $\frac{1}{k}$.

If two or more prototypes are coming close to a cluster, they tend to move very close together due to the equal sharing of membership values of nearby data objects. This multiple representation of clusters can be resolved by simply removing all redundant prototypes. Therefore, the algorithm can end with less prototypes than it started, which means that it has to be initialised with an overestimation of prototypes. This also solves the problem of defining the number of clusters in a data set, which is often not known and hard to do, especially for high-dimensional data sets. Due to the overestimation of prototypes in the beginning, some prototypes end up covering very small cluster or random noise that created a denser area by chance. These prototypes usually represent only very small number of data objects which can easily be detected at the end of the update process. By removing all prototypes which have a sum of membership values below a predefined threshold: $\sum_{j=1}^{n} u_{ij} < \xi$ with threshold $0 < \xi \in \mathbb{R}$, these unecessary prototypes are remuved.

6 Application in S.O.D.A.

In this section we want to demonstrate the problems of FCM and similar algorithms as well as demonstrate the advantages of the proposed algorithm. We use two examples, one real world example in cooperation with Fraport AG and and one artificial generated example to demonstrate that the problems are not induced by the specific data set. The Fraport AG develops an analysis tool called S.O.D.A. (Surveillance Data Analysis Tool) to analyse the movement patterns of aircraft on the airfield of the Frankfurt airport. The database contains approx. 700.000 aircraft tracks and the goal is to find groups of aircraft that move similar routes.

Due to the large number of tracks in the database and the complexity of comparing two tracks directly, we decided to simplify the task by transforming the data. A set of 457 reference points is added to the airport structure, for each track, the closest distance to each reference point is computed. To simplify the data further, the distance values are transformed using a simple, trapezoid fuzzy rule: let $d \in \mathbb{R}$ be the minimal distance of a reference point to an aircraft track, then $f(d) = 1$ for $d \leq a$, $f(d) = \frac{d-a}{b-a}$ for $a < d < b$ and $f(d) = 0$ if $d \geq b$ with $a = 25m$ and $b = 50m$. This rule simply states that for $f(d) = 1$, it is sure that the aircraft passed over this point, for $f(d) \in (0,1)$ the case is unsure and for $f(d) = 0$ it is sure the aircraft did not pass over the reference point. Each fuzzified distance value corresponds to one dimension, the resulting dataset is therefore 457-dimensional. In Figure 1 (leftmost subfigure), 10'000 transformed aircraft tracks are presented as grey points, projected on two of the 457 dimensions and with some jitter for demonstration purposes. In

the second left subfigure of Figure 1, an artificial dataset with 50 dimensions, 100 uniform distributed clusters which have in turn are sampled from a 100-dimensional normal distribution with the location of the cluster as expectation vector. Also the artificial data set contains 10% uniform distributed noise.

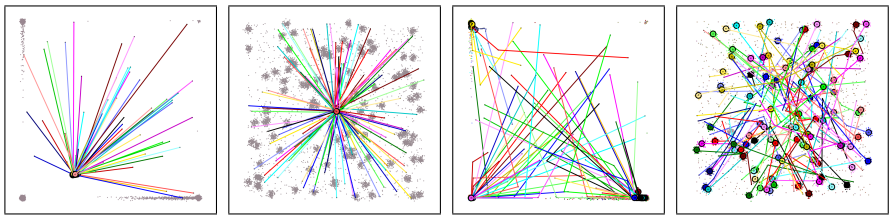

Fig. 1 High-dimensional data sets, projected on 2 dimensions

The effect of FCM on this data set is demonstrated by the colourful circles which represent the prototypes. The lines represent the ways, the prototypes took from their initial position to their their final position. It is clearly visible that FCM is not working in both cases. The colour of the data objects indicate that they are shared equally by all clusters as their colour indicate their cluster membership. The third and fourth subfigure of Figure 1 present the same datasets, but instead of the euclidean distance, DCR is used. The algorithm was initialised with 200 prototypes in both cases. In the S.O.D.A. dataset, 62 clusters were found and on the artificial dataset, 99 out of 100 clusters which is almost perfect. In Figure 2, 8 out of the 62 clusters of the S.O.D.A. dataset are presented. To reduce the overlapping effect of fuzzy clustering for this figure, only tracks with a membership value of at least 0.85 to their respective cluster are shown.

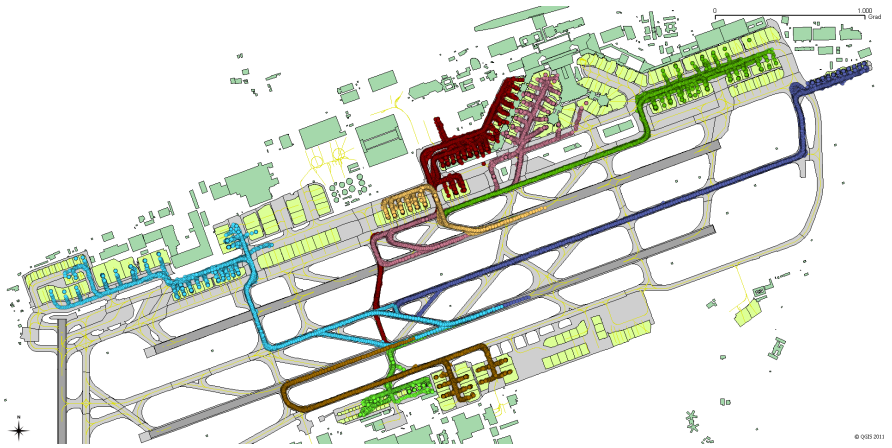

Fig. 2 8 Clusters in the S.O.D.A. dataset

7 Conclusions

We presented a very simple alteration to the distance function that is very effective in countering effect of distance concentration on a prototype based clustering algorithm. The alteration provides additionally the chance of estimating the number of clusters in a data set by overestimating the number of prototypes needed and removing unnecessary ones. The process has been shown for FCM in particular but is not restricted to it, the distance function can also be useful for EM, NC and similar algorithms. To prove our point, we have applied the algorithm on aircraft movement data and on an artificial data set.

Acknowledgements

We like to thank the FRAPORT AG for providing the data for scientific analysis, represented by Steffen Wendeberg, Thilo Schneider and Andreas Figur. We also would like to thank the engineers of DLR Braunschweig for setting up the database system, namely Hans Kawohl and his staff.

References

1. Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is nearest neighbor meaningful? In *Database Theory - ICDT'99*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235. Springer Berlin / Heidelberg, 1999.
2. James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
3. Rajesh N. Dave. Characterization and detection of noise in clustering. *Pattern Recogn. Lett.*, 12(11):657–664, 1991.
4. J.C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Cybernetics and Systems: An International Journal*, 3(3):32–57, 1973.
5. Robert J. Durrant and Ata Kabán. When is 'nearest neighbour' meaningful: A converse theorem and implications. *Journal of Complexity*, 25(4):385 – 397, 2008.
6. Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What is the nearest neighbor in high dimensional spaces? In *VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases*, pages 506–515, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
7. Chih-Ming Hsu and Ming-Syan Chen. On the design and applicability of distance functions in high-dimensional data space. *Knowledge and Data Engineering, IEEE Transactions on*, 21(4):523 –536, 2009.
8. Roland Winkler, Frank Klawonn, and Rudolf Kruse. Fuzzy c-means in high dimensional spaces. *IJFSA*, 1(1):1–16, 2011.