Constrained Clustering with Minkowski Weighted K-Means

Renato Cordeiro de Amorim
Department of Computer Science and Information Systems
Birkbeck University of London
Malet Street, London WC1E 7HX
Email: renato@dcs.bbk.ac.uk

***Abstract*—In this paper we introduce the Constrained Minkowski Weighted K-Means. This algorithm calculates cluster specific feature weights that can be interpreted as feature rescaling factors thanks to the use of the Minkowski distance. Here, we use an small amount of labelled data to select a Minkowski exponent and to generate clustering constrains based on pair-wise must-link and cannot-link rules.**

We validate our new algorithm with a total of 12 datasets, most of which containing features with uniformly distributed noise. We have run the algorithm numerous times in each dataset. These experiments ratify the general superiority of using feature weighting in K-Means, particularly when applying the Minkowski distance. We have also found that the use of constrained clustering rules has little effect on the average proportion of correctly clustered entities. However, constrained clustering does improve considerably the maximum of such proportion.

Keywords: Minkowski Weighted K-Means; Constrained K-Means; Minkowski metric; semi-supervised learning; feature weighting.

I. Introduction

The aim of any clustering algorithm is to partition a dataset Y into K groups of homogeneous entities $y \in Y$ creating a partition $S = \{S_1, S_2, ..., S_K\}$. These algorithms are rather popular and they have been used to tackle the most diverse problems, such as: summarizing data [1], detecting anomalous patterns [2], clustering mental tasks [3], clustering malware by its behaviour [4] and etc.

Clustering algorithms are traditionally divided into hierarchical and partitional. While the former iteratively merges smaller (or divide larger) clusters, producing enough information to generate a dendogram, the latter produces a single set of clusters. Among the partitional algorithms we have K-Means [5, 6], as probably the most popular. K-Means iteratively minimizes the within-cluster dissimilarity between entities and their respective centroids producing a non-overlapping partition, in other words, a given entity y can belong to a single partition S_k. The K-Means criterion is given below:

$$W(S, C) = \sum_{k=1}^{K} \sum_{i \in S_k} d(y_i, c_k) \qquad (1)$$

where K represents the number of clusters, c_k the centroid of cluster S_k and d a function returning the distance between an entity $y_i \in S_k$ and its respective centroid c_k. The algorithm outputs a clustering S and a set of centroids $C = \{c_1, c_2, ..., c_K\}$.

We present the iterative minimisation algorithm of the K-Means criterion below:

1) Select K entities at random as initial centroids, put them in C. Initialize $S = \{\}$
2) Assign each entity y_i to the closest centroid c_k creating the clustering S'. The algorithm supports a variety of distance measures, represented as d in Equation (1). If $S = S'$ stop.
3) Move each centroid $c_k \in C$ to the centre of gravity of S_k. This centre is given by the distance measure, for Euclidean distance this would be the mean. return to Step 2.

Visibly, the above iterative minimization of Equation (1) has its weaknesses, among them: (i) it requires the user to know the number of clusters in the dataset beforehand; (ii) there is no guarantee the minimization will reach a global minima; (iii) the final clustering depends heavily on the initial centroids, normally found at random; (iv) distance measures create a bias, for instance using Euclidean distance would bias the algorithm towards spherical clusters; (v) it treats all features as having the same degree of importance.

Even with all the above weaknesses, K-Means remains popular to this date. Implementations of K-Means can be found in most data analysis software packages such as MATLAB, R, SPSS and WEKA. The success of K-Means inspired us to tackle two of its major problems: (iii) the use of random initial centroids and (v) the equal treatment to all features. Initially focusing on the later, we decided to investigate the possibility of an algorithm setting the degree of importance of each feature for clustering purposes by using feature weights. We expanded the research on feature weighting of Huang et al. [8, 7, 9] by developing the Minkowski Weighted K-Means (MWK-Means) [10]. This algorithm follows the intuitive idea that features may have different degrees of importance at different clusters and that this should be taken into account by the criterion in Equation (1).

In our original publication [10], we addressed the issue raised by using random centroids by initializing our MWK-Means algorithm with a modified version of intelligent K-Means [2]. The later being a popular algorithm used to find the number of clusters in a dataset as well as its initial centroids [3, 4, 11].

Naturally the MWK-Means algorithm requires a Minkowski exponent p. This exponent can be approximated via semi-supervised learning [10]. In this paper we expand the learning stage of our algorithm to make use of pairwise clustering rules, introducing then the Constrained Minkowski Weighted K-Means (CMWK-Means). We have recently compared a six different initializations for MWK-Means and found intelligent K-Means and a modified version of the Ward method [12] to work the best in our experiments. In this paper we experiment with the later in a total of 12 datasets, derived from four originals to which we added noise features.

II. Background

While developing MWK-Means we had to alter the calculation of distances to apply feature weights to the original K-Means criterion (Equation 1). We decided to introduce a weight w_{kv} dependent on both, cluster k and feature v, allowing a given feature v to have different weights at different clusters k. We also introduced the use of the Minkowski distance to the p power, analogous to the Euclidean squared distance.

$$d_p(y_i, c_k) = \sum_{v=1}^{V} w_{kv}^p |y_{iv} - c_{kv}|^p \tag{2}$$

where V represents the features and p is the Minkowski exponent, a user-defined parameter that can be found with semi-supervised learning. We have further modified the weight variable to take into account the Minkowski exponent p, using then w_{kv}^p. By using this exponent in the distance as well as the feature weight we transform the latter in feature rescaling factors. This means that the feature weights could be used on the data-preprocessing stage regardless of what clustering algorithm one would use.

We have shown that by using the Minkowski distance, rather than Euclidean, our algorithm achieves better accuracy, particularly when dealing with noisy datasets [10, 13]. We have updated the K-Means criterion to:

$$W_p(S, C, w) = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{v=1}^{V} w_{kv}^p |y_{iv} - c_{kv}|^p \tag{3}$$

The calculation of the feature weights follows the intuitive idea introduced by Chan et al. [8], in which a feature v with small variation in relation to others in a particular cluster k has a higher degree of importance for clustering purposes in this particular cluster k than the other features $u \in V$. We calculate the feature weights per cluster using the following equation:

$$w_{kv} = \frac{1}{\sum_{u \in V} [D_{kv}/D_{ku}]^{1/(p-1)}} \tag{4}$$

where the dispersion of a feature v in a cluster k in relation to its cluster centroid c_k is given by

$$D_{kv} = \sum_{i \in S_k} |y_{iv} - c_{kv}|^p \tag{5}$$

Visibly this dispersion followed the Minkowski metric. The MWK-Means criterion can be minimized iteratively by following a similar algorithm to K-Means, with an extra step to calculate the weights.

1) Set the values of K centroids and V weights per centroid (if unknown $v_{ik} = 1/V$).
2) Assign each entity to its closest centroid, calculate the distances with Equation (2).
3) Update each centroid to the Minkowski centre of its cluster. If there are no changes, stop.
4) Recalculate all feature weights for each cluster applying Equation (4); Go back to Step 2.

The Minkowski centre for a given p can be approximated using the steepest descent method we have used in our original publication [10] or even using a genetic algorithm, but the latter is likely to be much slower.

MWK-Means is clearly a non-deterministic algorithm, meaning that by using different initializations one may arrive at a different clustering. We have made a comparison of six different initializations [13] concluding that Minkowski-based versions of the intelligent K-Means [2] and Ward [12] methods are the best performers, the latter particularly in high-dimensional datasets. We formalize the Minkowski Ward initialization below.

1) Set all entities $y \in Y$ as a singletons, having $S = \{S_1, S_2, ..., S_N\}$, where N is the cardinality of Y.
2) Merge clusters S_{w1} and S_{w2}, which are the closest as per the Ward criterion using the Minkowski metric.
3) Replace the references of S_{w1} and S_{w2}, with that representing a merged cluster $S_{w1 \cup w2}$.
4) If $K > K^*$, K^* being the desired number of clusters, return to step 2.

The Ward distance using the Minkowski metric, d_{wp}, is given by:

$$d_{wp}(S_{w1}, S_{w2}) = \frac{N_{w1} N_{w2}}{N_{w1} + Nw2} \sum_{v=1}^{V} |c_{w1v} - c_{w2v}|^p \tag{6}$$

where c_{w1} and c_{w2} refer to the centroids of clusters S_{w1} and S_{w2}, respectively. N_{w1} and N_{w2} represent the cardinality of each V-dimensional cluster.

The final clustering of MWK-Means very much depends on the Minkowski exponent p used. This exponent can be successfully approximated using a small amount of labelled data. Following this approach, we cluster the whole dataset with different values for p, normally between one and five in steps of 0.1, and choose the optimal p based on the accuracy of a small amount of labelled data.

Visibly, this semi-supervised learning approach opens the doors to the use of constrained clustering. The question we address here is: can this small amount of labelled data be used to something more than simply approximate the Minkowski exponent?.

Constrained clustering in K-Means has been introduced by Wagstaff et al. [14]. This approach makes use of a limited amount of background knowledge by applying pairwise must-link and cannot-link rules to entities. As their name suggest, these rules state what pairs of entities are known to belong to the same cluster and what pairs are known to belong to different clusters.

Since we have a small amount of labelled data when using MWK-Means, we can generate must-link and cannot-link rules from these labels. The basic idea is that this algorithm will change the assignment of the entities present in the rules that would be otherwise incorrectly clustered. This way the centroids will move to locations closer to the real centre of gravity of the clusters and by consequence it may allow a better general clustering. Constrained clustering is of easy application and in a previous work we have successfully adapted it to the intelligent K-Means initialization [15]

III. The Constrained Minkowski Weighted K-Means (CMWK-Means)

MWK-Means requires a small portion of the dataset to be labelled, these labels are used to select a good value for p. Likewise, CMWK-Means requires a limited quantity of labelled data, but unlike MWK-Means it extends its use to find a better clustering by using must-link and cannot-link rules. We have joined these two approaches so the available labelled data is used for estimating p as well as further enhancing the cluster recovery of MWK-Means. We formalize the algorithm below.

1) Run MWK-Means, initialized with the Ward method using the Minkowski metric, 50 times with each p from one to five in steps of 0.1 on the whole dataset.
2) Select p^* as the p with the highest average accuracy over the 50 runs within the labelled data.
3) Using the labelled data, generate the must-link and cannot link rules
4) Run the Minkowski Ward initialization with p^*, finding K Centroids $C = \{c_1, c_2, ...c_K\}$;
5) Set all weights v_{ik} to $1/V$.
6) Assign each entities to its closest centroids making sure the assignment does not break any of the constrained rules, calculate the distances with Equation (2).
7) Update each centroids to the Minkowski centres of its cluster. If there are no changes, stop.
8) Recalculate all feature weights for each cluster applying Equation (4); Go back to Step 6.

We have found the above algorithm to be of easy implementation, the extra overhead processing to generate the must-link and cannot-link rules was minimum. This was not really a surprise as amount of labelled data was small.

IV. Experiments with CMWK-Means

We have performed experiments on 12 datasets. The four original datasets were chosen from UCI [16] based on their popularity. In order to demonstrate the power of feature weighting, we have derived datasets from the four originals, containing features with uniformly random noise. The datasets are:
- Iris
This dataset contains solely numerical data, it has 150 entities over four features, partitioned into three clusters. From this dataset we have derived two other with two and four extra noise features.
- Wine
178 entities over 13 numerical features partitioned into three clusters. From this dataset we have derived two others with 7 and 13 noise features.
- Hepatitis
155 entities over 19 features, mostly categorical, partitioned into two clusters. From this dataset we have derived two others with 10 and 20 noise features.
- Pima Indians diabetes.
768 entities over 8 numerical features, partitioned into two clusters. From this dataset we have derived another two with four and eight extra noise features.

We have standardized all datasets in order to deal with features using different scales. We have standardise numerical features by subtracting their average and dividing the result by the features range. as shown below:

$$z_{iv} = \frac{y_{iv} - \bar{y_v}}{max(y_v) - min(y_v)} \tag{7}$$

where $\bar{y_v}$ represents the average of feature v over all entities in the dataset $y \in Y$. The use of the range rather then standard deviation as scaling factor favours bimodal distributions and it has empirical support [17]. Regarding the standardization of categorical features we decided to follow a method described by Mirkin [2] that allows us to remain faithful to our criterion. In this, each feature with n categories is transformed into n new binary features. These new features have the value of one only in those entities which were under the corresponding category and zero otherwise.

We have used a confusion matrix to map the clusters generated by our algorithm to the labels of the whole datasets. The accuracy of each algorithm was set as the proportion of the correctly labelled entities.

In the experiments for both MWK-Means and CMWK-Means we have used 20% of labelled data in the semi-supervised step. To find p we run experiments with the whole datasets with ps between 1 and 5 in steps of 0.1 and choose the p with the highest average accuracy in 50 runs.

We have calculated the accuracy of CMWK-Means as well as its expected accuracy. The later relates to the fact that if we feed the algorithm with, say, 20% of labels in the form of must-link and cannot-link we should already expect an increase related to this 20% of the data. The real point to be analysed is the impact of such rules in the accuracy in the remaining 80% of the dataset. We calculate this expected accuracy as per the Equation below:

$$E_\mu = MWK_\mu + \frac{N_L - (MWK_\mu * N_L)}{N} \tag{8}$$

where N is the size of the dataset, N_L is the size of the labelled data and MWK_μ is the average accuracy of MWK-Means on the same dataset. E_{Max} uses a very similar formula with the maximum accuracy of MWK-Means, MWK_{Max} instead of MWK_μ.

Tables I, II, III, IV show the accuracy results for the iris, wine, hepatitis and Pima Indians diabetes datasets. In all tables we show the results for both MWK-Means and CMWK-Means being initialized with the Minkowski Ward method, as well as WK-Means [8, 7, 9] and K-Means.

TABLE I

RESULTS FOR THE EXPERIMENTS WITH THE IRIS DATASET. THE ROWS SHOW THE RESULTS FOR THE ORIGINAL IRIS, WITH TWO AND FOUR EXTRA NOISE FEATURES, RESPECTIVELY

	Accuracy			Expected		
	μ	σ	Max	μ	Max	p
CMWK-Means	95.87	2.60	98.67	95.11	96.7	1.2
	97.02	0.69	98.00	95.40	95.3	1.1
	95.56	5.22	98.67	96.06	96.7	1.1
MWK-Means	94.53	3.2	96.7	-	-	1.2
	94.87	0.3	95.3	-	-	1.3
	95.33	1.7	96.7	-	-	1.1
WK-Means	86.80	2.89	92.71	-	-	1.68
	82.1	3.3	88.8	-	-	1.23
	85.4	2.5	89.4	-	-	1.21
K-Means	84.0	12.3	89.3	-	-	-
	67.1	6.4	76.7	-	-	-
	66.7	7.0	80.0	-	-	-

TABLE II

RESULTS FOR THE EXPERIMENTS WITH THE WINE DATASET. THE ROWS SHOW THE RESULTS FOR THE ORIGINAL WINE, WITH SEVEN AND 13 EXTRA NOISE FEATURES, RESPECTIVELY

	Accuracy			Expected		
	μ	σ	Max	μ	Max	p
CMWK-Means	94.06	3.64	96.07	94.72	95.5	1.71
	94.96	1.21	97.19	94.74	95.5	1.49
	96.29	1.81	97.75	93.18	93.8	1.28
MWK-Means	94.07	1.7	95.5	-	-	1.6
	94.35	0.9	95.5	-	-	1.6
	92.58	1.4	93.8	-	-	1.5
WK-Means	92.34	1.15	93.53	-	-	3.41
	90.7	1.3	93.2	-	-	3.59
	85.4	1.8	89.1	-	-	3.74
K-Means	95.3	0.4	96.6	-	-	-
	93.0	6.5	96.6	-	-	-
	87.5	11.0	93.3	-	-	-

TABLE III

RESULTS FOR THE EXPERIMENTS WITH THE HEPATITIS DATASET. THE ROWS SHOW THE RESULTS FOR THE ORIGINAL HEPATITIS, WITH 10 AND 20 EXTRA NOISE FEATURES, RESPECTIVELY

	Accuracy			Expected		
	μ	σ	Max	μ	Max	p
CMWK-Means	80.34	9.68	89.68	85.12	86.49	1.33
	79.20	8.53	89.68	84.98	85.2	1.49
	79.93	10.71	89.68	85.11	87.04	2.45
MWK-Means	82.52	2.5	85.2	-	-	1.4
	82.10	3.0	85.2	-	-	1.8
	82.64	3.2	86.4	-	-	2.5
WK-Means	76.16	3.44	78.76	-	-	1.92
	77.3	1.9	78.8	-	-	2.10
	77.6	1.3	80.1	-	-	2.65
K-Means	71.51	1.36	72.26	-	-	-
	71.26	1.88	72.90	-	-	-
	70.10	1.78	70.97	-	-	-

TABLE IV

RESULTS FOR THE EXPERIMENTS WITH THE PIMA INDIANS DIABETES DATASET. THE ROWS SHOW THE RESULTS FOR THE ORIGINAL DATASET AND THE VERSION WITH FOUR EXTRA NOISE FEATURES, RESPECTIVELY

	Accuracy			Expected		
	μ	σ	Max	μ	Max	p
CMWK-Means	69.93	9.29	78.25	75.07	74.99	4.02
	72.41	4.96	78.25	73.09	73.78	1.83
	-	-	-	-	-	-
MWK-Means	69.13	1.7	70.3	-	-	3.8
	66.91	2.2	68.7	-	-	1.8
	68.78	1.92	69.79	-	-	2.06
WK-Means	62.73	0.96	64.09	-	-	3.26
	63.2	1.6	65.3	-	-	1.55
	65.1	1.48	66.6	-	-	1.74
K-Means	66.67	0.55	66.80	-	-	-
	52.25	1.11	53.38	-	-	-
	51.58	1.23	53.64	-	-	-

We find the results in our tables quite promising. They show that by using the constrained rules the average accuracy of the CMWK-Means is competitive or higher and the maximum accuracy is always higher when compared with not using the rules.

V. CONCLUSION

In this paper we have presented the Constrained Minkowski Weighted K-Means. This is a further modification to Minkowski Weighted K-Means, an algorithm shown to be more accurate than a variety of other algorithms [10, 13]. Because of its non-deterministic nature we have chosen to initialize it with the Ward method, also utilizing the Minkowski metric and have selected an appropriate Minkowski exponent by using semi-supervised learning.

This new algorithm makes full use of a limited amount of labelled data by using the must-link and cannot-link clustering rules introduced by Wagstaff et al. [14].

In general, our experiments ratify the superiority of using feature-weighting in K-Means, particularly when applying the Minkowski metric. We have also found that the use of constrained rules does not seem to increase the average accuracy in more than what one should expect. The maximum accuracy in the other hand seems to have a considerable increase. This suggests that perhaps we should investigate other initializations for the CMWK-Means, which we intend to do in the future.

References

[1] A.K. Jain. “Data clustering: 50 years beyond K-means”. In: *Pattern Recognition Letters* 31.8 (2010), pp. 651–666.

[2] B.G. Mirkin. *Clustering for data mining: a data recovery approach*. Vol. 3. CRC Press, 2005.

[3] R.C. de Amorim, B. Mirkin, and J.Q. Gan. “Anomalous pattern based clustering of mental tasks with subject independent learning - some preliminary results”. In: *Artificial Intelligence Research* 1.1 (2012), pp. 46–54.

[4] R.C. de Amorim and P. Komisarcsuk. “On partitional clustering of malware”. In: *The First International Workshop on Cyber Patterns: Unifying Design Patterns with Security, Attack and Forensic Patterns (CyberPatterns)*. Abingdon, Oxfordshire, UK. 2012, pp. 47–51.

[5] J. MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 281-297. California, USA. 1967, p. 14.

[6] G.H. Ball and D.J. Hall. “A clustering technique for summarizing multivariate data”. In: *Behavioral Science* 12.2 (1967), pp. 153–155.

[7] J.Z. Huang et al. “Automated variable weighting in k-means type clustering”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.5 (2005), pp. 657–668.

[8] E.Y. Chan et al. “An optimization algorithm for clustering using weighted dissimilarity measures”. In: *Pattern recognition* 37.5 (2004), pp. 943–952.

[9] J.Z. Huang et al. “Weighting Method for Feature Selection in K-Means”. In: *Computational Methods of Feature Selection* (2008), pp. 193–210.

[10] R.C. de Amorim and B. Mirkin. “Minkowski Metric, Feature Weighting and Anomalous Cluster Initializing in K-Means Clustering”. In: *Pattern Recognition* 45.3 (2012), pp. 1061–1075.

[11] M.M.T. Chiang and B. Mirkin. “Intelligent choice of the number of clusters in K-Means clustering: an experimental study with different cluster spreads”. In: *Journal of classification* 27.1 (2010), pp. 3–40.

[12] J.H. Ward Jr. “Hierarchical grouping to optimize an objective function”. In: *Journal of the American statistical association* (1963), pp. 236–244.

[13] R.C. de Amorim and P. Komisarczuk. “On Initializations for the Minkowski Weighted K-Means”. In: *Lecture Notes in Computer Science* 7619 (2012), pp. 45–55.

[14] K. Wagstaff et al. “Constrained k-means clustering with background knowledge”. In: *Machine Learning International Workshop then Conference*. 2001, pp. 577–584.

[15] R.C. de Amorim. “Constrained Intelligent K-Means: Improving Results with Limited Previous Knowledge.” In: *ADVCOMP’08*. IEEE. 2008, pp. 176–180.

[16] A. Frank and A. Asuncion. *UCI Machine Learning Repository*. 2010. URL: http://archive.ics.uci.edu/ml.

[17] G.W. Milligan and P.D. Isaac. “The validation of four ultrametric clustering algorithms”. In: *Pattern Recognition* 12.2 (1980), pp. 41–50.