Noname manuscript No.
(will be inserted by the editor)

Feature relevance in Ward's hierarchical clustering using the L_p norm

Renato Cordeiro de Amorim

Received: 10 January 2013 / Accepted: 20 December 2013

Abstract In this paper we introduce a new hierarchical clustering algorithm called Ward$_p$. Unlike the original Ward, Ward$_p$ generates feature weights, which can be seen as feature rescaling factors thanks to the use of the L_p norm. The feature weights are cluster dependent, allowing a feature to have different degrees of relevance at different clusters.

We validate our method by performing experiments on a total of 75 real-world and synthetic datasets, with and without added features made of uniformly random noise. Our experiments show that: (i) the use of our feature weighting method produces results that are superior to those produced by the original Ward method on datasets containing noise features; (ii) it is indeed possible to estimate a good exponent p under a totally unsupervised framework. The clusterings produced by Ward$_p$ are dependent on p. This makes the estimation of a good value for this exponent a requirement for this algorithm, and indeed for any other also based on the L_p norm.

Keywords Ward method · Hierarchical clustering · Feature weights · Feature relevance · L_p norm · Minkowski metric

This is an accepted manuscript and now published in Journal of Classification 32 (2015), 46-62, doi:10.1007/s00357-015-9167-1.

RC de Amorim
Department of Computer Science and Information Systems, Birkbeck University of London, Malet Street, London WC1E 7HX, UK.
Tel.: +44 020-7631-6700
Fax: +44 020-7631-6727
E-mail: renato@dcs.bbk.ac.uk

1 Introduction

Cluster analysis aims to partition a dataset Y into K clusters $S = \{S_1, S_2, ..., S_K\}$ without the need of labelled samples. Such task has been applied to various problems in fields of research including data mining, computer security, computer vision, taxonomy and many others (Jain, 2010; Mirkin, 2005; Kaufman and Rousseeuw, 1990).

Generally, the algorithms used in cluster analysis can be divided into partitional and hierarchical. The former is composed of algorithms originally producing disjoint clusters in which an entity $y_i \in Y$ can be assigned to a single cluster S_k. Fuzzy set theory (Zadeh, 1965) extends this concept by allowing an entity to belong to all clusters with a degree of membership $u_{ik} \in [0, 1]$ for $k = 1, 2, ..., K$ and $i = 1, 2, ..., N$, N representing the cardinality of Y. There are a number of partitional algorithms, K-Means (Ball and Hall, 1967; MacQueen, 1967) being the most popular (Jain, 2010) together with its variant Fuzzy C-Means (Bezdek, 1981).

Hierarchical clustering algorithms take a different approach. They aim to produce a set of clusters as well as the relationship between them. This tree-like relationship can be demonstrated visually with a dendogram. In these a given entity may belong to more than one cluster, as long as these clusters are related and the belongness occurs at different levels. Hierarchical algorithms can be further divided into divisive or agglomerative depending whether the algorithm takes a top-down or bottom-up approach, in this paper we focus on the latter. There are many agglomerative algorithms depending on the criterion used to decide which clusters to merge, among them and perhaps the most popular, we have the Ward method (Ward, 1963). This method merges the two clusters that have the smallest cost to merge as per the equation below.

$$Ward(S_i, S_j) = \frac{N_{S_i} N_{S_j}}{N_{S_i} + N_{S_j}} d(c_{S_i}, c_{S_j}), \tag{1}$$

where N_{S_i} and c_{S_i} represent the cardinality and centroid of cluster S_i, respectively, while N_{S_j} and c_{S_j} represent the same for cluster S_j, and $d()$ is a function returning the distance between the centroids of each of the two clusters. Each cluster S_k is represented by a single centroid $c_k \in C$, which is the centre of gravity of cluster S_k, equivalent to its average if using the L_2 norm.

Other hierarchical clustering algorithms include single-linkage and complete-linkage. The former merges the two clusters that have the minimum dissimilarity between entities of one and the other cluster, given by $min_{x,y}\{d(x, y)|x \in S_i, y \in S_j\}$. This criterion may lead to a bias towards elongated clusters and sensitivity to noise and outliers. Complete-linkage considers the distance between two clusters to be the maximum dissimilarity between the entities of one and the other cluster, given by $max_{x,y}\{d(x, y)|x \in S_i, y \in S_j\}$. The latter criterion is less susceptible to noise than the former, but tends to break large clusters and generate clusters with a similar diameter. For a review, see the works of Mirkin (2005); Kaufman and Rousseeuw (1990), and Xu and Wunsch (2005).

Hierarchical clustering, and in particular the Ward method, has been used to address a number of different problems, including finding the number of clusters in datasets (Haldar et al., 2008). The Ward method criterion in (1) allows for the merging of the two clusters that will increase the total within-cluster variance by the minimum possible. However, this method is not without weaknesses. It is an iterative method of low scalability with a greedy nature that assumes all features have the same relevance. The latter being of particular interest in this paper.

Taking the above into account we set the main contribution of this paper to be a new algorithm called Ward$_p$. Our algorithm applies subspace feature weighting to take into consideration the different degrees of relevance of each feature $v \in V$. This allows a given feature v to have K weights, one for each cluster. Our solution uses the p^{th} root of the L_p distance, analogous to the Euclidean squared distance, frequently used in clustering algorithms. We define the former distance for the V-dimensional entities x and y as $d_p(x, y) = \sum_{v=1}^{V} |x_v - y_v|^p$, where x_v and y_v represent the value of feature $v \in V$ in x and y, respectively. This clearly introduces p as new parameter to our algorithm, leading to our second contribution: an unsupervised method to estimate a good value for the exponent p.

In the rest of this paper we will present our background research, as well as a proper introduction for our algorithm. We describe the setting of the experiments validating Ward$_p$ and show the results for 75 datasets including real-world and synthetic datasets.

2 Background

The Ward method iteratively merges two clusters at a time, making sure the merger will increase the total within-cluster variance by the minimum possible. Below, we formalize the algorithm where N is the total number of entities in the dataset Y.

1. Set $K = N$. Each cluster in $S = \{S_1, S_2, ..., S_K\}$ is composed of a single different entity, making it a singleton.
2. Merge the clusters S_i and S_j which are the closest as per (1), creating the new cluster $S_{S_i \cup S_j}$. Remove references to the old clusters S_i and S_j, as well as their centroids c_{S_i} and c_{S_j}.
3. Set the centroid of $S_{S_i \cup S_j}$ to its centre of gravity.
4. Reduce K in 1, if K is still bigger than the desired number of clusters go back to Step 2.

This is a very popular method. It has been extended numerous times before (Murtagh and Legendre, 2013), including an extension applying powers of the Euclidean distance (Szekely and Rizzo, 2005). In its original version, the algorithm stops when all entities $y_i \in Y$ are combined into a single cluster of size N. In our experiments K is known, allowing us to change the stop criterion so that the cluster merges cease when the number of partitions in S

is equal to K. The complexity of the Ward method is of $\mathcal{O}(N^3)$, making it not a particularly scalable algorithm. However, this is not a problem we address in this paper.

There are indeed other hierarchical clustering algorithms that follow a similar approach to the above. Most notably single, average and complete linkage algorithms (Sorensen, 1948; Sokal and Michener, 1958; Florek et al., 1951). Their basic difference to the Ward method lies in the criterion used to decide which clusters to merge in step 2. However, none of these algorithms recognizes that different features may have different degrees of relevance.

The use of feature weights in clustering was introduced by DeSarbo (1984) with SYNCLUS, a partitional clustering algorithm. SYNCLUS first applies K-Means to a given dataset and then estimates an optimal set of weights by optimizing a weighted mean-square, stress-like cost function. These two steps are iterated until convergence, which may be too computationally demanding for large datasets (Green et al., 1990).

In hierarchical clustering, Soete (1986, 1988) introduced a method to solve the feature weighting problem finding optimal feature weights for ultrametric and additive tree fitting. This method was later extended for K-Means clustering (Makarenkov and Legendre, 2001) using the Polak-Ribiere optimization procedure, making the algorithm not particularly fast.

Other related work has been done on feature selection as a pre-processing step for hierarchical clustering algorithms (Talavera, 1999). However, feature selection algorithms simply select a subset of meaningful features from V, by setting degrees of relevance to either zero or one. Under this framework it is not possible to distinguish the degree of relevance of features that have been selected.

More recently, Liu and Yu (2005) introduced a method to integrate feature selection algorithms for classification and clustering. Mitra et al. (2002) take a different approach, by introducing a novel unsupervised algorithm for feature selection. This method is based on measuring the similarity between features whereby redundancy is limited. However, the latter introduces a parameter whose optimum seems difficult to be estimated.

We see no reason why feature weighting should be a preprocessing step, and no reason why one should use a method constrained to weights of either zero or one. Feature weighting can be done at the same time as the clustering itself. Feature weighting has received considerable attention in partitional clustering (Amorim and Mirkin, 2012; Amorim and Fenner, 2012; Chan et al., 2004; Huang et al., 2005, 2008; Makarenkov and Legendre, 2001), but not so in hierarchical clustering. Surely, it is possible to apply a feature selection algorithm to a dataset before using the Ward method. However, this is not incorporated into Ward and it does not take into account that even among relevant features there may be different degrees of relevance.

Our recent work on feature weighting in partitional algorithms (Amorim and Mirkin, 2012; Amorim and Fenner, 2012) has introduced the use of weights under the L_p norm as we show in Equation 2. We have decided to use the L_p norm because this transforms the weights into feature rescaling factors, in

contrast to the work of Chan et al. (2004), and Huang et al. (2005, 2008).

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

where y_i is an entity in the dataset Y. Each weight w_{kv} being calculated by using Equation 3.

$$w_{kv} = \frac{1}{\sum_{u \in V} [D_{kvp}/D_{kup}]^{1/(p-1)}}, \tag{3}$$

where $D_{kvp} = \sum_{i \in S_k} |y_{iv} - c_{kv}|^p$. The weights $w_{k1}, w_{k2}, ..., w_{kV}$ are subject to a sum of one for a given entity y_i, as well as a crisp clustering.

In our previous publications we used a semi-supervised method to estimate a good value for the exponent p. This method involves the clustering of the whole dataset Y various times, each of which with a different value for p. We then select the best p based on the proportion of correctly clustered entities whose labels are known.

The above method recovered clusters close to the optimal in experiments on various real-world and synthetic datasets containing noise features (Amorim and Mirkin, 2012). However, this semi-supervised method does not apply to scenarios in which there are no labelled entities.

3 Ward$_p$

Taking into consideration the weaknesses identified in the Ward method, together with its popularity, we found reasonable to try to improve it. With this in mind we have decided extend the Ward method by incorporating feature weighting, creating the Ward$_p$ method.

Our new method uses the weighted L_p norm (2). Our choice was justified by its success in other clustering algorithms based on K-Means (Amorim and Mirkin, 2012; Amorim and Komisarczuk, 2012a) and partition around medoids (Amorim and Fenner, 2012). We can see at least two advantages of Ward$_p$ over these feature-weighted partitional clustering algorithms: (i) neither Ward or Ward$_p$ require the number of clusters to be known beforehand; (ii) hierarchical algorithms provide more information regarding the structure of a dataset. As a disadvantage, one should not expect a hierarchical algorithm to run faster than a partitional, particularly in large datasets as we have previously demonstrated while clustering malicious software (Amorim and Komisarczuk, 2012b).

The implementation of feature weights using the L_p norm in the Ward distance (1) is straightforward:

$$Ward_p(S_i, S_j) = \frac{N_{S_i} N_{S_j}}{N_{S_i} + N_{S_j}} \sum_{v \in V} w_{kv}^p |c_{S_i v} - c_{S_j v}|^p. \tag{4}$$

The above requires a change in the way weights are calculated. Equation (2) works perfectly when there is only one cluster k involved, w_{kv} is the weight

of feature v in the cluster represented by the centroid c_k. However, in (4) we clearly have two clusters, S_i and S_j. The fastest solution we could find was to use the average of the weights of each cluster. We have also experimented calculating w_k for $S_i \cup S_j$ for all possible pairs, but this proved to be time consuming and did not produce good results, hence we do not explore this path here.

We formalise Ward$_p$ below:

1. Set $K = N$ and $w_{kv} = 1/V$. Each cluster in $S = \{S_1, S_2, ..., S_K\}$ is composed of a single different entity, a singleton.
2. Merge the clusters S_i and S_j which are the closest as per (4) the weight used is the average of w_{S_iv} and w_{S_jv}, creating the new cluster $S_{S_i \cup S_j}$.
3. Set the centroid of the new cluster to the cluster's centre of gravity. Remove references to the old clusters and their centroids.
4. Update each w_{kv} for $k = \{1, 2, ..., K\}$ and $v = \{1, 2, ..., V\}$ using (3).
5. Reduce K in 1, if K is still bigger than the desired number of clusters go back to Step 2.

Similarly to the original Ward, Ward$_p$ could be run until each entity $y_i \in Y$ belongs to a single cluster of size N. However, in our experiments we stop the cluster merges when the number of clusters in S equals the desired number. This is possible because the desired number of clusters of each of the datasets we experiment with is known.

Clearly, the above algorithm requires the calculation of centroids. This is straightforward when p is equal to one or two, as the centre of gravity is given by the median and mean respectively. Should p be a different value, one can use a steepest descent algorithm (Amorim and Mirkin, 2012). Given the reals $y_{1v}, y_{2v}, ..., y_{Nv}$ where v is a given feature and N the cardinality of Y, the L_p norm centre can be defined as c minimising the summary rule below.

$$d_p(c) = \sum_{i=1}^{N} |y_{iv} - c|^p. \tag{5}$$

Since $d_p(c)$ in (5) is convex for $p > 1$, the steepest descent algorithm uses its first derivative, $'(c) = p(\sum_{i \in I^+}(c - y_{iv})^{p-1} - \sum_{i \in I^-}(y_{iv} - c)^{p-1})$, where I^+ represents the set of indices i at which $c > y_{iv}$, and I^- is the set of indices i at which $c < y_{iv}$ for $i = 1, 2, ..., N$. The algorithm is given below.

1. Sort the values of a given feature v in ascending order so that $y_{1v} \leq y_{2v} \leq ... \leq y_{Nv}$.
2. Set $c_0 = y_{i^*v}$, the minimiser of $d(c)$ on y_{iv}, and a positive learning rate λ of say, 10% of the feature range $y_{Nv} - y_{iv}$.
3. Set c_1 to $c_0 - \lambda d'_p(c_0)$ if it falls within the minimum interval $(y_{iv'}, y_{iv''})$ containing y_{i^*v} and such that $d(y_{iv'}) > d_p(y_{i^*v})$, $d(y_{iv''}) > d_p(y_{i^*v})$. Otherwise decrease λ by say 10%, and repeat the step.
4. If c_1 is equal to c_0 within a pre-specified threshold, output c_1 as the optimal value for c and stop.

5. If $d(c_1) \leq d(c_0)$, set $c_0 = c_1$ and $d(c_0) = d(c_1)$, go to Step 2. Otherwise, decrease λ by 10% and go to Step 3 with an unchanged c_0.

With the above it is possible to use the Ward$_p$ algorithm under any $p \geq 1$. The final clustering generated by Ward$_p$ is subjective to the p used, we show how to find a good value for this parameter in Section 5.2.

Ward$_p$ applies cluster specific weights, w_{kv} for $k = 1, 2, ..., K$ and $v \in V$ to the clustering process. It would also be possible to apply feature weights w_v for $v \in V$ instead. The former was shown to produce better K-Means based clusterings while using the Euclidean distance (Huang et al., 2005), hence its use in this paper.

4 Setting of the experiments

We have experimented with a total of 75 datasets, 30 real-world-based and 45 synthetic. Regarding the former, we have downloaded most of them from the UCI machine learning repository (Frank and Asuncion, 2010), with the only exception of the Tulugu vowels dataset, first presented by Pal and Majumder (1977). From each of these we have generated other two datasets by adding approximately 50% and 100% extra features containing uniformly distributed noise, as per below:

1. *Iris*. This dataset contained 150 entities over four numeric features, partitioned into three clusters. We have generated two other datasets by adding two and four noise features to it.
2. *Wine*. This dataset contained 178 entities over 13 numerical features, partitioned into three clusters. We have generated two other datasets by adding seven and 13 noise features to it.
3. *Pima*. This dataset contained 768 entities over eight numerical features, partitioned into two clusters. We have generated two other datasets by adding four and eight noise features to it.
4. *Hepatitis*. This dataset contained 155 entities originally over 19 categorical and numerical features, partitioned into two clusters. We have generated two other datasets by adding 10 and 20 noise features to it.
5. *Breast cancer* This dataset contained 699 entities over nine numerical features, partitioned into two clusters. We have generated two other datasets by adding five and nine noise features to it.
6. *Ecoli* This dataset contained 336 entities over seven numerical features, partitioned into eight clusters. We have generated two other datasets by adding four and seven noise features to it.
7. *Glass* This dataset contained 214 entities over ten numerical features, partitioned into six clusters. We have generated two other datasets by adding five and ten noise features to it.
8. *SPECTF heart* This dataset contained 267 entities over 44 numerical features, partitioned into two clusters. We have generated two other datasets by adding 22 and 44 noise features to it.

9. *Tulugu vowels* This dataset, used for the first time by Pal and Majumder (1977), contained 871 entities over three numerical features, partitioned into six clusters. We have generated two other datasets by adding two and three noise features to it.
10. *Vehicle silhouette* This dataset contained 846 entities over 18 numerical features, partitioned into four clusters. We have generated two other datasets by adding 9 and 18 noise features to it.

We generated a total of 45 Gaussian models (GMs) for our experiments with synthetic datasets. We initially generated five models from different mixtures for each of the following configurations: (i) 500 entities over six features, partitioned into five clusters (here denoted 500x6-5), (ii) 500 entities over six features, partitioned into 10 clusters (500x6-10), and (iii) 500 entities over 12 features partitioned into five clusters (500x12-5).

To each of the above configurations we added further 10 GMs, five by adding 50% of noise features and the other five by adding 100% noise features. For instance, from the initial five GMs under 500x6-5 we added 3 noise features to each of the datasets (500x6-5 +3NF) and 6 noise features (500x6-5 +6NF), a subtotal of 15 datasets per configuration.

The clusters of all datasets are spherical with diagonal covariance matrices with the same diagonal value σ^2 generated at each cluster randomly between 0.5 and 1.5. The centroid components were generated independently from a $N(0, 1)$ Gaussian distribution. The cardinalities of each clusters were uniformly random with a minimum of 20 entities.

We have standardized all datasets as per the equation below.

$$y_{iv} = \frac{x_{iv} - \bar{x}_v}{range(x_v)}, \tag{6}$$

where x_i represents an entity in the dataset Y and $\bar{x}_v$ the average of feature v over the whole dataset Y. We have used the range rather than the standard deviation as scaling factor because the latter favours unimodal distributions (Mirkin, 2005). We have also counted with considerable empirical support for the use of the range in clustering (Milligan and Cooper, 1988; Steinley, 2004) as well as our own previous success (Amorim and Mirkin, 2012; Amorim and Fenner, 2012).

The standardisation of categorical features followed a method described by Mirkin (Mirkin, 2005), in which a categorical feature v of range r is transformed into r binary features representing each of the possible categories of v. For a given entity, only one of the new features should be set to one, the feature representing the original category, the rest should be set to zero. We also standardize numerically these features by subtracting its grand mean, the category proportion.

We evaluate Ward$_p$ with two different measurements. First, we demonstrate its ability to recover clusters, particularly in datasets containing noise features, using the adjusted Rand index (Hubert and Arabie, 1985). Second, we show

the contribution of the noise features to the clustering, calculated as per below.

$$V'_{Contribution} = \frac{\sum_{v \in V'} \frac{1}{K} \sum_{k=1}^{K} w_{kv}}{|V'|/|V|}, \tag{7}$$

where V' is a subset of V, containing solely those features composed of uniformly random noise, $|V'|$ and $|V|$ represents their respective cardinalities. This is an easy to interpret measure that is comparable if applied to any of our noise datasets, regardless of the actual number of noise features. For instance, if half of the features in a given dataset are in fact noise features, and these noise features contain half of the feature weights, then our measure would output 1. In the same dataset, an output of 0.5 would mean that the weights of the noise features are 25% of the total. In our tables we multiply Equation (7) by 100 to improve readability.

5 Results

In this section we show the results of our experiments, our aim here is twofold. We first show in Section 5.1 that given a good exponent p, our Ward$_p$ produces results that are generally superior to those obtained with the original Ward method when applied to datasets with noise features. We then show in Section 5.2 that by using the Silhouette index (Rousseeuw, 1987) to analyse the clustering obtained with different values of p we can still select a reasonably good p and obtain results that are generally competitive or superior to those obtained with the Ward method.

5.1 Optimal exponent p

In order to show the behaviour of Ward$_p$ in the best possible scenario, we have run experiments with values of p from 1 to 5 with the progress step of 0.1. We then analysed the accuracy given by the adjusted Rand index of each one of these clusterings and chose as optimal p that with the highest accuracy.

We have experimented with both the real-world and synthetic datasets. Table 1 shows the results of our experiments with real-world datasets with and without noise. The results show a clear superiority of Ward$_p$ over Ward in datasets with noise features, as expected.

Ward$_p$ recovered better clusters than Ward in 18 of the 20 noise datasets we experimented with. The two remaining noise datasets are the noise versions of the same, the Breast Cancer dataset, with five and nine extra noise features. The difference of performance between Ward$_p$ and Ward on these two datasets is of only 1.08 and 3.75, respectively. The results related to the real-world datasets with no noise features are considerably more modest with each algorithm obtaining the best adjusted Rand index in half of the datasets.

Regarding the contribution of the noise features to the clustering, calculated with Equation (7) we can see that Ward$_p$ has failed to reduce this in

only a single dataset out of the 20 noise datasets, the SPECTF Heart with 44 noise features. This dataset has two clusters. The sum of the weights of each noise feature for one of the clusters is zero, however, for the other cluster the only feature with a weight, of one, is among the noise features. The original cardinalities of the clusters are of 212 and 55, but Ward$_p$ partitioned the dataset clusters of 255 and 12 entities. The small number of entities in one of the clusters is probably the reason for the weights obtained.

The three images in Figure 1 show the results for the synthetic datasets configurations 500x6-5, 500x6-5 +3NF and 500x6-5 +6NF, respectively. The first bar represents the adjusted Rand index obtained with Ward over the five datasets under each configuration, and the third with the Ward$_p$ when supplied with the best p. We discuss the results of the second bar, Ward$_p$ with an estimates p, in Section 5.2. Similarly, Figure 2 shows the results for the synthetic datasets configurations 500x6-10, 500x6-10 +3NF, 500x6-10 +6NF, and Figure 3 shows the results for 500x12-5, 500x12-5 +6NF, and 500x12-5 +12NF. Although Ward$_p$ was clearly affected by the increase of noise features, the effect of this increase was considerably worse in the original Ward. There was a particularly high difference between these two algorithms in the maximum adjusted Rand index they obtained under each of the five Gaussian mixtures for each configuration. Again with a clear superiority of Ward$_p$.

5.2 Unsupervised selection of the exponent p

The results we show in Table 1 and Figures 1, 2 and 3 are rather promising, but we acknowledge that it would not be realistic to expect the user to know what exponent p would be the best for each dataset. In partitional clustering there have been solutions for this problem using semi-supervised learning (Amorim and Mirkin, 2012; Amorim and Fenner, 2012; Amorim and Komisarczuk, 2012a), in which a good p is selected by using a small number of labelled data.

Although valid, the use of semi-supervised learning to select p is not well aligned with the use of unsupervised clustering algorithms. Here we show that it is indeed possible to select a reasonably good p without the use of labels. We have run experiments for each dataset with values of p from 1 to 5 with the progress step of 0.1 and have chosen the optimal as the one that produced the clustering with the highest Silhouette index (Rousseeuw, 1987). This index is given by the equation below:

$$Si(y_i) = \frac{b(y_i) - a(y_i)}{max\{a(y_i), b(y_i)\}}, \tag{8}$$

where $a(y_i)$ is the average distance of a given entity $y_i \in S_k$ from $y_j \in S_k$, $i \neq j$, and $b(y_i)$ is the lowest average distance of y_i from $y_j \in S_l$ at $l \neq k$. The Silhouette index for a clustering is given by $\sum_{y_i \in Y} Si(y_i)$.

We show the results for this truly unsupervised method for the real-world datasets in Table 2. Out of the 20 noise datasets we experiment with, Ward$_p$

recovered clusterings with a higher adjusted Rand index than Ward in 16 cases. The remaining four datasets are the two noise versions of the Breast Cancer dataset, Pima +8NF and Vehicle +9NF. The adjusted Rand index difference between Ward and $Ward_p$ for the last two datasets was of only 1.45 and 0.26, respectively.

The Figures 1, 2 and 3 show the results related to the synthetic datasets we experiment with. In all images the middle bar represents the average adjusted Rand index obtained with $Ward_p$ using the estimated p with the Silhouette index. As expected the difference in cluster recovery between $Ward_p$ and Ward is smaller in this unsupervised framework than when using the optimal p in Section 5.1.

The results with the Gaussian models show that $Ward_p$ is superior to Ward in average, in datasets with and without noise features.

Fig. 1 The average adjusted Rand index of $Ward$, $Ward_p$ with an estimated p, and $Ward_p$ with the best p over five synthetic datasets under each of the configurations: 500x6-5, 500x6-5 +3 noise features, and 500x6-5 + 6 noise features. $Ward_p$ est. p uses the silhouette index to select the exponent p.

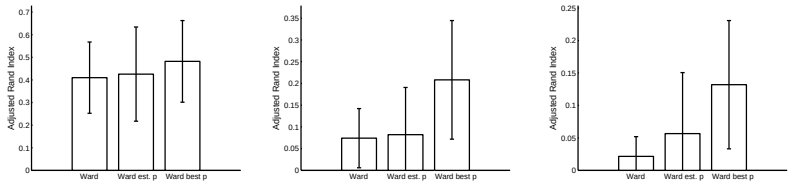
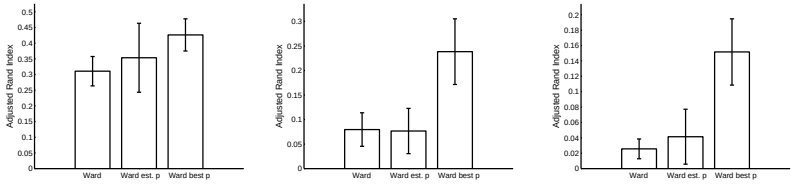


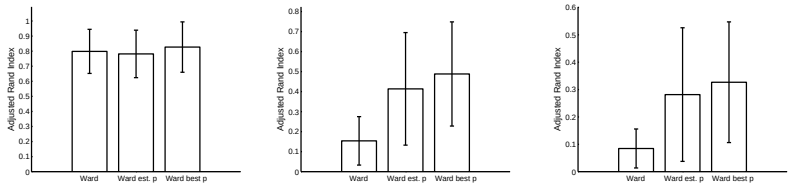
Fig. 2 The average adjusted Rand index of $Ward$, $Ward_p$ with an estimated p, and $Ward_p$ with the best p over five synthetic datasets under each of the configurations: 500x6-10, 500x6-10 +3 noise features, and 500x6-10 + 6 noise features. $Ward_p$ est. p uses the silhouette index to select the exponent p.



6 Conclusion and future work

In this paper we have introduced the use of feature relevance to hierarchical clustering. We have done so by applying subspace feature weighting and the use of the L_p norm to the original Ward method, developing $Ward_p$. Our new

Fig. 3 The average adjusted Rand index of $Ward$, $Ward_p$ with an estimated p, and $Ward_p$ with the best p over five synthetic datasets under each of the configurations: 500x12-5, 500x12-5 +6 noise features, and 500x12-5 +12 noise features. $Ward_p$ est. p uses the silhouette index to select the exponent p.



algorithm allows for a given feature v to have K weights, one for each of the clusters in a dataset Y.

We have empirically shown through numerous experiments with 75 real-world and synthetic datasets, with and without features composed of uniformly random values, that the feature weights produces by Ward$_p$ tend to be higher in the relevant features. Our experiments also show that Ward$_p$ produces results that are competitive or superior to those produced by Ward, particularly in datasets containing noise features.

Our experiments also demonstrated that the final clustering generated by our Ward$_p$ method is subjective to the exponent p used. This exponent is part of the L$_p$ norm and in this paper we worked the convex problem given by $p \geq 1$. We have shown that it is indeed possible to estimate a good value for this exponent without using labelled entities, remaining then totally under an unsupervised learning framework.

We see Ward$_p$ as being an algorithm ready to be used in a number of fields, in particular those in which irrelevant features are common and that require the demonstration of the relation between taxons, such as malware taxonomy and bioinformatics. A particularly interesting application would be use Ward$_p$ for refining phylogenetic inference techniques which are often based on weights and optimization of the L$_2$ norm (Makarenkov and Leclerc, 1999; Felsenstein, 1997).

Clearly there is still room for improvement in Ward$_p$. Both Ward$_p$ and the original Ward require the calculation of centroids, resulting in the former being considerably slower. Obtaining c_{kv} using the Euclidean distance is easily accomplished by $c_{kv} = \frac{1}{|S_k|} \sum_{y_i \in S_k} y_{iv}$. Unfortunately when applying the L$_p$ norm used in Ward$_p$ this is not so straight forward, requiring the algorithm shown in Section 3 to approximate the $p-$center of $y_{iv} \in S_k$. Ward$_p$ would clearly benefit from a faster calculation of such $p-$center. We also see the selection of an exponent p even closer to its optimum as a particularly interesting problem. We intend to address both issues in future research.

References

AMORIM, R.C., and FENNER, T. (2012), "Weighting features for Partition Around Medoids using the Minkowski metric", *Lecture Notes in Computer Science*, 7619, pp. 35-44.

AMORIM, R.C., and KOMISARCZUK, P. (2012a), "On Initializations for the Minkowski Weighted K-Means", *Lecture Notes in Computer Science*, 7619, pp. 45-55.

AMORIM, R. C., and KOMISARCZUK, P. (2012b), "On partitional clustering of malware", *The First International Workshop on Cyberpatterns: Unifying Design Patterns with Security, Attack and Forensic Patterns*, Abingdon, UK, pp. 47-51.

AMORIM, R.C., and MIRKIN, B. (2012), "Minkowski Metric, Feature Weighting and Anomalous Cluster Initializing in K-Means Clustering", *Pattern Recognition*, 45.3, pp. 1061-1075.

BALL, G.H., and HALL D.J. (1967), "A clustering technique for summarizing multivariate data", *Behavioral Science*, 12.2, pp. 153-155.

BEZDEK, J.C. (1981), *Pattern recognition with fuzzy objective function algorithms*, Norwell MA: Kluwer Academic Publishers.

CHAN, E.Y., CHING, W.K., NG, M.K., and HUANG, J.Z. (2004), "An optimization algorithm for clustering using weighted dissimilarity measures", *Pattern recognition*, 37.5, pp. 943-952.

DESARBO, W.S., CARROLL, J.D., CLARK, L. A., and GREEN, P.E. (1984), "Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables". *Psychometrika*, 49.1, pp. 57-78.

FELSENSTEIN, J. (1997), "An alternating least squares approach to inferring phylogenies from pairwise distances", *Systematic biology*, 46.1, pp. 101-111.

FLOREK, K., LUKASZEWICZ, J., PERKAL, J., STEINHAUS, H., and ZUBRZYCKI, S. (1951), "Taksonomia Wroclawska", *Przegl. antrop.*, 17, pp. 93-207.

FRANK, A., and ASUNCION, A. (2010), "UCI Machine Learning Repository", University of California, Irvine, School of Information and Computer Sciences, [accessed on 12 October 2012, http://archive.ics.uci.edu/ml].

GREEN, P.E., CARMONE, F.J., and KIM, J. (1990), "A Preliminary Study of Optimal Variable Weighting in k-Means Clustering", *Journal of Classification*, 7.2, pp. 271-285.

HALDAR, P., PAVORD, I. D., SHAW, D. E., BERRY, M .A., THOMAS, M., BRIGHTLING, C. E., WARDLAW, A. J., and GREEN, R. H. (2008), "Cluster analysis and clinical asthma phenotypes", *American journal of respiratory and critical care medicine*, 178.3, pp. 218-224.

HUANG, J.Z., NG, M.K., RONG, H., and LI, Z. (2005), "Automated variable weighting in k-means type clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27.5, pp. 657-668.

HUANG, J.Z., XU, J., NG, M., and YE, Y. (2008), "Weighting Method for Feature Selection in K-Means", in *Computational Methods of Feature Selec-*

tion, eds. H. LIU, and H. MOTODA, Chapman and Hall/CRC, pp. 193-210.

HUBERT, L., and ARABIE, P. (1985), "Comparing partitions", *Journal of classification*, 2.1, pp. 193–218.

JAIN A.K. (2010), "Data clustering: 50 years beyond K-means", *Pattern Recognition Letters*, 31.8, pp. 651-666.

KAUFMAN, L., and ROUSSEEUW, P.J. (1990), *Finding groups in data: an introduction to cluster analysis*, Hoboken, New Jersey: John Wiley & Sons, Inc.

LIU, H., and YU, L. (2005), "Toward integrating feature selection algorithms for classification and clustering", *IEEE Transactions on Knowledge and Data Engineering*, 17.4, pp. 491-502.

MACQUEEN, J. (1967), "Some methods for classification and analysis of multivariate observations", *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, University of California, Berkeley, pp. 281-297.

MAKARENKOV, V., and LECLERC, B. (1999), "An algorithm for the fitting of a tree metric according to a weighted least-squares criterion", *Journal of classification*, 16.1, pp. 3-26.

MAKARENKOV, V. and LEGENDRE, P. (2001), "Optimal variable weighting for ultrametric and additive trees and K-means partitioning: Methods and software", *Journal of Classification*, 18.2, pp. 245-271.

MILLIGAN, G. W., and COOPER, M. C. (1988), "A study of standardization of variables in cluster analysis", *Journal of Classification*, 5.2, pp. 181-204.

MIRKIN, B. (2005), *Clustering for data mining: a data recovery approach*, Boca Raton FL: Chapman and Hall/CRC.

MITRA, P., MURTHY, C.A., and PAL, S.K.(2002), "Unsupervised feature selection using feature similarity", *IEEE transactions on pattern analysis and machine intelligence*, 24.4, pp. 301-312.

MURTAGH, F., and LEGENDRE, P. (in press, 2013), "Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion", *Journal of Classification.*

PAL, S.K., and MAJUMDER, D.D. (1977), "Fuzzy sets and decision making approaches in vowel and speaker recognition", *Transactions on Systems, Man, and Cybernetics*, 7, pp. 625-629.

ROUSSEEUW, P.J. (1987), "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis", *Journal of computational and applied mathematics*, 20, pp. 53-65.

SOETE, G. (1986), "Optimal Variable Weighting for Ultrametric and Additive Tree Clustering", *Quality and Quantity*, 20.2, pp. 169-180.

SOETE, G. (1988), "OVWTRE: A program for optimal variable weighting for ultrametric and additive tree fitting". *Journal of Classification*, 5.1, pp. 101-104.

SOKAL, R. R., and MICHENER, C. (1958), "A statistical method for evaluating systematic relationships", *Univ Kans Sci Bull*, 38, pp. 1409-1438.

SØRENSEN, T. (1948), "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to

analyses of the vegetation on Danish commons", *Biol. skr.*, 5, pp. 1-34.

STEINLEY, D. (2004), "Standardizing variables in K-means", in *clustering, Classification, clustering, and data mining applications*, eds. D. BANKS, F.R. MCMORRIS, P. ARABIE, and W. GAUL, Heidelberg: Springer, pp. 53-60.

SZÉKELY, G.J., and RIZZO, M.L. (2005), "Hierarchical Clustering via Joint Between-Within Distances: Extending Ward's minimum Variance Method". *Journal of Classification*, 22.2, pp. 151-183.

TALAVERA, L. (1999), "Feature selection as a preprocessing step for hierarchical clustering", *Proceedings of the Sixteenth International Conference on machine learning*, Slovenia, pp. 389-397.

WARD JR, J.H. (1963), "Hierarchical grouping to optimize an objective function", *Journal of the American statistical association*, pp. 236-244.

XU, R. and WUNSCH, D. II (2005), "Survey of clustering algorithms", *IEEE Transactions on Neural Networks*, 16.3, pp. 645-678.

ZADEH, L.A. (1965), "Fuzzy sets", *Information and control*, 8.3, pp. 338-353.

Table 1 A comparison on real-world datasets between the Ward method and Ward$_p$, the latter using the best possible exponent p we could find between 1 and 5 with the progress step of 0.1.

	Ward$_p$			Ward
	Adj Rand Index	p	N.F. Contribution	Adj Rand Index
Iris	92.22	2.9	-	71.96
Iris +2	88.58	1.5	4.61	47.64
Iris +4	74.20	2.0	21.18	44.30
Wine	84.83	2.1	-	93.10
Wine +7	86.13	3.5	48.57	71.08
Wine +13	72.87	2.0	31.72	47.19
Pima	2.38	4.7	-	7.27
Pima +4	3.35	3.0	34.51	-0.19
Pima +8	4.75	3.8	37.26	2.46
Hepatitis	32.00	1.4	-	35.49
Hepatitis +10	32.00	1.4	0.00	19.14
Hepatitis +20	35.37	2.6	0.35	8.79
Breast Cancer	86.06	4.4	-	86.64
Breast Cancer +5	84.98	4.9	66.52	86.06
Breast Cancer +9	80.69	1.3	20.51	84.44
Ecoli	51.80	4.9	-	39.93
Ecoli +4	5.29	5.0	77.90	0.45
Ecoli +7	5.29	2.8	61.87	1.37
Glass	29.06	4.8	-	43.47
Glass +5	25.81	2.9	23.52	10.72
Glass +10	23.75	1.2	33.33	2.02
SPECTF Heart	0.43	1.0	-	-10.63
SPECTF Heart +22	-0.55	3.0	45.71	-8.93
SPECTF Heart +44	1.14	1.1	100.0	-10.53
Tulugu Vowels	50.38	3.6	-	38.72
Tulugu Vowels +2	33.14	4.3	72.66	18.48
Tulugu Vowels +3	26.49	4.1	80.0	4.25
Vehicle	17.22	1.9	-	9.77
Vehicle +9	13.75	1.2	0.0	6.69
Vehicle +18	14.86	1.8	3.46	4.37

Table 2 A comparison on real-world datasets between the $Ward$ method and $Ward_p$, the latter using the silhouette index to select the exponent p.

	Ward$_p$			Ward
	Adj Rand Index	p	N.F. Contribution	Adj Rand Index
Iris	86.85	4.6	-	71.96
Iris + 2	50.60	3.9	64.38	47.64
Iris + 4	52.45	3.0	82.04	44.30
Wine	80.41	2.2	-	93.10
Wine + 7	80.60	3.6	49.95	71.08
Wine + 13	72.87	2.0	31.72	47.19
Pima	0.55	3.7	-	7.27
Pima + 4	0.78	4.9	65.71	-0.19
Pima + 8	1.01	5.0	53.39	2.46
Hepatitis	21.68	1.1	-	35.49
Hepatitis + 10	21.68	1.1	0.0	19.14
Hepatitis + 20	28.12	2.7	0.60	8.79
Breast Cancer	85.51	4.6	-	86.64
Breast Cancer +5	84.98	4.9	66.52	86.06
Breast Cancer +9	77.53	1.7	38.30	84.44
Ecoli	50.51	5.0	-	39.93
Ecoli +4	3.23	4.9	78.97	0.45
Ecoli +7	3.64	2.4	75.04	1.37
Glass	29.06	4.8	-	43.47
Glass +5	20.25	3.6	71.32	10.72
Glass +10	21.05	4.2	64.38	2.02
SPECTF Heart	-0.55	4.4	-	-10.63
SPECTF Heart +22	-2.09	2.6	31.65	-8.93
SPECTF Heart +44	-1.08	1.7	60.24	-10.53
Tulugu Vowels	42.57	5.0	-	38.72
Tulugu Vowels +2	27.40	4.8	66.86	18.48
Tulugu Vowels +3	12.08	4.5	72.92	4.25
Vehicle	8.12	4.3	-	9.77
Vehicle +9	6.43	4.6	43.85	6.69
Vehicle +18	7.19	4.9	49.33	4.37