# Kernel Hierarchical Agglomerative Clustering
## *Comparison of Different Gap Statistics to Estimate the Number of Clusters*

Na Li, Nicolas Lefebvre and Régis Lengellé
*Charles Delaunay Institute - UMR CNRS 6281, Université de Technologie de Troyes, 12 Rue Marie Curie, Troyes, France*

Keywords: Hierarchical Agglomerative Clustering, Gap Statistics, Kernel Alignment, Number of Clusters.

Abstract: Clustering algorithms, as unsupervised analysis tools, are useful for exploring data structure and have owned great success in many disciplines. For most of the clustering algorithms like *k*-means, determining the number of the clusters is a crucial step and is one of the most difficult problems. Hierarchical Agglomerative Clustering (HAC) has the advantage of giving a data representation by the dendrogram that allows clustering by cutting the dendrogram at some *optimal* level. In the past years and within the context of HAC, efficient statistics have been proposed to estimate the number of clusters and the Gap Statistic by Tibshirani has shown interesting performances. In this paper, we propose some new Gap Statistics to further improve the determination of the number of clusters. Our works focus on the kernelized version of the widely-used Hierarchical Clustering Algorithm.

## 1 INTRODUCTION

Clustering is the task of grouping objects according to some measured or perceived characteristics of them and it has owned great success in exploring the hidden structure of unlabeled data sets. As a useful tool for unsupervised classification, it has drawn increasing attention in various domains including psychology (Harman, 1960), biology (Sneath et al., 1973) and computer security (Barbará and Jajodia, 2002). Clustering algorithms have a long history. They originated in anthropology by Driver and Kroeber (Driver and Kroeber, 1932). In 1967 one of the most useful and simple clustering algorithms, *k*-means (Mac Queen et al., 1967), has been proposed. Since then a lot of classical algorithms, like fuzzy *c*-means (Bezdek et al., 1984), Hierarchical Agglomerative Clustering (HAC) etc have emerged.

Meanwhile, another clustering method, kernel-based clustering, has arisen and owned great success because of its ability to perform linear tasks in some non linearly transformed spaces. In machine learning, the kernel trick has been firstly introduced by Aizerman (Aizerman et al., 1964). It became famous in Support Vector Machines (SVM) initially proposed by Cortes and Vapnik (Cortes and Vapnik, 1995). SVM has shown better performances in many problems and this success has brought an extensive use of the kernel trick into other algorithms like kernel PCA (Schölkopf et al., 1998), non linear (adaptive) filtering (Príncipe et al., 2011) etc. Kernel methods have been widely used in supervised classification tasks like SVM and then they were extended to unsupervised classification. A lot of kernel-induced clustering algorithms have emerged due to the extensive use of inner products. Most of these algorithms are kernelized versions of the corresponding conventional algorithms. Surveys of kernel-induced methods for clustering have been done in (Filippone et al., 2008; Kim et al., 2005; Muller et al., 2001). The first proposed and the most well-known kernel-induced algorithm is kernel *k*-means by Scholkopf (Schölkopf et al., 1998). A further version has been proposed by Girolami (Girolami, 2002). After that, several kernel-induced algorithms have emerged such as kernel fuzzy *c*-means, kernel Self Organizing Maps, kernel average-linkage etc. Compared with the corresponding conventional algorithms, kernelized criteria have shown better performance especially for non-linearly separable data sets.

According to our literature survey, a few work has been done on kernel based HAC (see, e.g. (Qin et al., 2003), (Kim et al., 2005)). The prominence of HAC consists in the data description provided by a dendrogram which represents a tree of nested partitions of the data. Hierarchical clustering usually depends on distance calculations (to compute between-class and within-class dispersions, minimum linkage, max-

Li N., Lefebvre N. and Lengellé R..
Kernel Hierarchical Agglomerative Clustering - Comparison of Different Gap Statistics to Estimate the Number of Clusters.
DOI: 10.5220/0004828202550262
In *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2014), pages 255-262
ISBN: 978-989-758-018-5

imum linkage etc) which are based on inner products. In order to explore wider classes of classifiers, we can perform HAC after mapping the input data onto a higher dimension space using a nonlinear transform. This is one of the main ideas of kernel methods, where the transformed space is selected as a Reproducing Kernel Hilbert Space (RKHS) in which distance calculations can be easily evaluated with the help of the kernel trick (Aizerman et al., 1964).

In this paper we focus on the kernel based HAC framework which is robust in clustering non linearly separable data sets and, at the same time, several criteria are proposed to determinate the number of clusters, being inspired by the Gap Statistic of Tibshirani (Tibshirani et al., 2001). This paper is organized as follows: we start with an introduction of kernel HAC. Then we introduce the principle of Gap Statistics and our criteria to estimate the number of clusters. Subsequently, we provide the results of a simulation study and we compare the results obtained with those of Tibshirani, in standard and kernel HAC. We show that alternate Gap Statistics are possible to estimate the number of clusters and one of them, the Delta Level Gap, is efficient and robust to variations in the clustering procedure. Finally, we propose some perspectives to this work in order to obtain a fully automatic clustering algorithm.

# 2 KERNEL HIERARCHICAL AGGLOMERATIVE CLUSTERING (K-HAC)

## 2.1 Kernel Trick

The key notion in kernel based algorithms is the kernel trick. To introduce the kernel trick, we first recall Mercer's theorem (Mercer, 1909). Let $X$ be the original space. A kernel function $K : X \times X \longrightarrow \mathbb{R}$ is called a *positive definite kernel (or Mercer Kernel)* if and only if:

- $K$ is symmetric: $K(x,y) = K(y,x)\ \forall (x,y) \in X \times X$
- $\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j K(x_i,x_j) \geq 0, \forall i = 1,...,n, \forall n \geq 2$ where $c_i \in \mathbb{R}$

For each Mercer kernel we have:

$$K(x_i,x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle \tag{1}$$

where $\phi : X \rightarrow F$ performs the mapping from the original space onto the (high dimensional) feature space. As shown in equation (1), inner product calculations in the feature space can be computed by a kernel function in the original space, without explicitly specifying the mapping function $\Phi$. The computation of Euclidean distance in feature space $F$ benefits from this idea.

$$\begin{aligned} d^2(\Phi(x_i),\Phi(x_j)) &= \|\Phi(x_i) - \Phi(x_j)\|^2 \\ &= K(x_i,x_i) + K(x_j,x_j) - 2K(x_i,x_j) \end{aligned} \tag{2}$$

Several commonly used Mercer kernels are listed in (Vapnik, 2000). In this paper, we only consider the gaussian kernel defined as follows:

$$K(x_i,x_j) = \exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma^2}\right) \tag{3}$$

## 2.2 Kernel-HAC

The advantage of HAC lies in the obtention of dendrogram (see an example in Figure 3), which gives a description of the data structure. By cutting the dendrogram at a given level, one can perform data clustering. The aim of introducing the kernel trick in HAC is to easily explore wider classes of similarity measures. Figures 1 and 2 show an example of a data set that cannot be clustered by standard HAC (Figure 1) while kernel HAC allows perfect clustering (Figure 2).

The HAC algorithm steps are 1) assign each data point to be a singleton; 2) calculate some similarity/dissimilarity between each pair of clusters; 3) merge the pairwise closest clusters into one; 4) repeat the two previous steps until only one final cluster is obtained.

Different similarity measures have been proposed and surveys have been done in (Murtagh, 1983; Olson, 1995). Examples of linkage criteria are:

- Single linkage
  $d(r,s) = min(d(x_{ri},x_{rj})), \quad x_{ri} \in r, \quad x_{rj} \in s$
- Complete linkage
  $d(r,s) = max(d(x_{ri},x_{rj})), \quad x_{ri} \in r, \quad x_{rj} \in s$
- Average linkage
  $d(r,s) = \frac{1}{n_r n_s}\sum_{i=1}^{n_r}\sum_{j=1}^{n_s} d(x_{ir},x_{js})$
- Ward's linkage
  $d^2(r,s) = n_r n_s \frac{\|\bar{x_r} - \bar{x_s}\|_2^2}{(n_r+n_s)}$

In this paper, we focus on the gaussian kernel based HAC using Ward's linkage criterion. In Ward's linkage, $\bar{x_r}$ denotes the the centroid of cluster $r$, $\bar{x_r} = \frac{1}{n_r}\sum_i^{n_r} x_i$. So in the feature space $F$, we have:

$$\begin{aligned} d^2(r^\Phi,s^\Phi) = \frac{n_r n_s}{(n_r+n_s)}\Bigg(&\frac{1}{n_r^2}\sum_i^{n_r}\sum_j^{n_r} K(x_i,x_j) \\ &+\frac{1}{n_s^2}\sum_i^{n_s}\sum_j^{n_s} K(x_i,x_j) - \frac{2}{n_r n_s}\sum_i^{n_r}\sum_j^{n_s} K(x_i,x_j)\Bigg) \end{aligned} \tag{4}$$

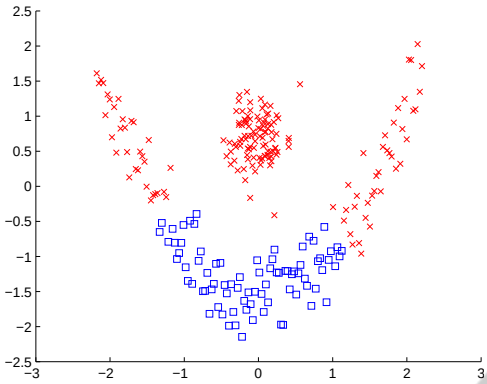

Figure 1: Result using HAC.

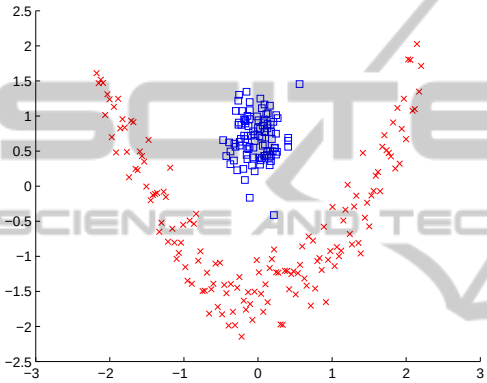

Figure 2: Result using kernel HAC.

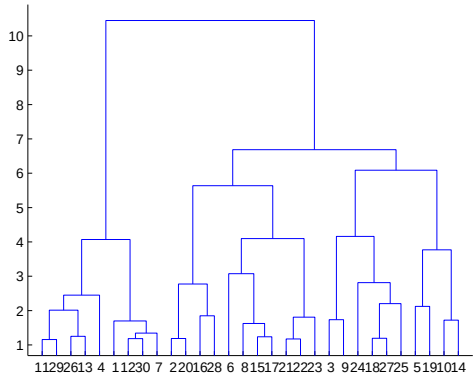


Figure 3: Dendrogram of this data set using kernel HAC.

As can be seen in figures 1 and 2 kernel HAC allows clustering of separable classes with small between-class dispersion in the original space.

# 3 DETERMINING THE NUMBER OF CLUSTERS

Determining the number of clusters is one of the most difficult problems in cluster analysis. For some algorithms like *k*-means, the number of clusters needs to be provided in advance. Over the past years, a lot of methods have emerged and the Gap Statistic proposed by Tibshirani (Tibshirani et al., 2001) is probably one of the most promising approach.

In the earlier times, Milligan and Copper (Milligan and Cooper, 1985) have done a research on the criteria for estimating the number of clusters, reporting the results of a simulation experiment designed to determine the validity of 30 criteria proposed in the literature. After that, several criteria emerged like Hartigan's rule (Hartigan, 1975), Krzanowski and Lai's index (Krzanowski and Lai, 1988), the silhouette statistic suggested by Kaufman and Rousseeuw (Kaufman and Rousseeuw, 2009) and Calinski and Harabasz's index (Caliński and Harabasz, 1974), which has demonstrated better performance under most of the situations considered in Milligan and Copper's study. More recent methods on determining the number of clusters include an approach using approximate Bayes factors proposed by Fraley and Raftery (Fraley and Raftery, 1998) and a jump method by Sugar and James (Sugar and James, 2003).

Unfortunately, most of these methods are somewhat ad hoc or model-based and hence sometimes require parametric assumptions which lead to a lack of generality. However, the principle of Gap Statistics proposed by Tibshirani et al. (Tibshirani et al., 2001) is to compare the within cluster dispersion obtained by the considered clustering algorithm with that we would obtain under a single cluster hypothesis. It is designed to be applicable to virtually any clustering method like the commonly used *k*-means and HAC.

## 3.1 Gap Statistics

Consider a data set $x_{ij}, i = 1,2,...,n, j = 1,2,...,p$, consisting of $p$ features, all of them being measured on $n$ independent observations. $d_{ii'}$ denotes some dissimilarity between observations $i$ and $i'$. The most common used dissimilarity measure is the squared Euclidean distance $(\sum_j (x_{ij} - x_{i'j})^2)$. Suppose that the data set is composed of $k$ clusters and that $C_r$ denotes the indices of observations in cluster $r$ and $n_r = |C_r|$. According to Tibshirani, and using his notations, we define:

$$D_r = \sum_{i,i' \in C_r} d_{ii'} \tag{5}$$

as the sum of all the distances between any two observations in cluster $r$. So, using again the notations of Tibshirani,

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \tag{6}$$

is the within-class dispersion. $W_k$ monotonically decreases as the number of clusters $k$ increases but, according to Tibshirani (Tibshirani et al., 2001), from some $k$ onwards, the rate of decrease is dramatically reduced. It has been shown that the location of such an *elbow* indicates the appropriate number of clusters.

The main idea of Gap Statistics is to compare the graph of $log(W_k)$ with its expectation we could observe under a single cluster hypothesis (the importance of the choice of an appropriate null reference distribution of the data has been studied in (Gordon, 1996)).

According to (Tibshirani et al., 2001), the assumed null model of the data set must be a single cluster model. The most common considered reference distributions are:

- an uniform distribution over the range of the observed data set.
- an uniform distribution over an area which is aligned with the principal components of the data set.

The first method has the advantage of simplicity while the second is more accurate in terms of consistency because it takes into account the shape of the data distribution.

Here again, using the notations of Tibshirani, we define the Gap Statistic as:

$$Gap(k) = E_n\{log(W_k/H_0)\} - log(W_k) \qquad (7)$$

Here $E_n\{log(W_k/H_0)\}$ denotes the expectation of $log(W_k)$ under some null reference distribution $H_0$. The estimated number of clusters $\hat{k}$ falls at the point where $Gap_n(k)$ is maximum. Expectation is estimated by averaging the results obtained from different realizations of the data set under the null reference distribution.

Figures 4 and 5 show an example using kernel HAC. Data are composed of three distinct bidimensional Gaussian clusters centred on $(0,0)$, $(0,1.3)$, $(1.4,-1)$ respectively, with unit covariance matrix $I_2$ and 100 observations per class. The functions $log(W_k)$ and the estimate of $E_n\{log(W_k/H_0)\}$ are shown in Figure 4. The Gap Statistic is shown in Figure 5. In this example, $E_n\{log(W_k/H_0)\}$ was estimated using 150 independent realizations of the null data set. We also estimated the standard deviation $sd(k)$ of $log(W_k/H_0)$. Let $s_k = \sqrt{1+\frac{1}{B}}sd(k)$, which is represented by vertical bars in Figure 5, then, according to Tibshirani, the estimated number of cluster $\hat{k}$ is the smallest $k$ such that:

$$Gap(k) \geq Gap(k+1) - s_{k+1} \qquad (8)$$

Figure 5 shows that, for the considered data set, $\hat{k} = 3$, which is correct.

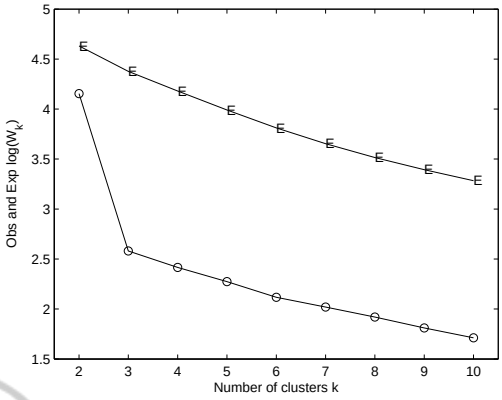


Figure 4: Graphs of $E_n\{log(W_k/H_0)\}$ (upper curve) and $log(W_k)$ (lower curve).

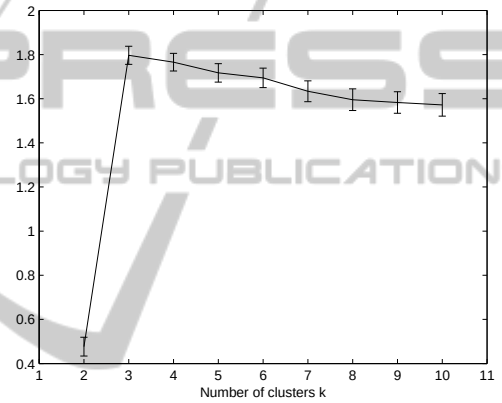


Figure 5: Gap statistic as a function of the number of clusters.

## 3.2 New Criteria to Estimate the Number of Clusters in Kernel HAC

The main idea of Gap Statistics was to compare the within-class dispersion obtained on the data with that of an appropriate reference distribution. Inspired by this idea, we propose to extend it to other criteria which are suitable for HAC to estimate the number of clusters.

These criteria are:

- Modified Gap Statistic
  In the Gap Statistic proposed by Tibshirani, the estimated number of clusters $\hat{k}$ is chosen according to equation 8. In this modified Gap Statistic, we define $\hat{k}$ as the number of clusters where $Gap(k)$ (equation 7) is maximum. As will be shown later in this paper, this allows to potentially improve the estimate, at least for standard HAC.
- Centered Alignment Gap
  A good guess for the nonlinear function $\Phi(x_i)$ should be to directly produce the expected result

$y_i$ for all observations. In this case, the *best* Gram matrix $K$ whose general term is $K_{ij} = K(x_i, x_j)$ becomes $YY'$ where $Y$ is the column vector (or matrix, depending on the selected code book) of all the data labels. Shawe-Taylor et al. ((Shawe-Taylor and Kandola, 2002)) have proposed a function that depends on both the labels and the Gram matrix, called *alignment*, to measure the degree of agreement between a kernel function and the clustering task. The alignment is defined by:

$$Alignment = \frac{< K, YY' >_F}{\sqrt{< K, K >_F < YY', YY' >_F}} \quad (9)$$

where the subscript $F$ denotes Frobenius norm. In this paper, $Y$ is estimated after the clustering process. The Centered Alignment (CA) is defined as:

$$CA = \frac{< K_c, Y_c Y_c' >_F}{\sqrt{< K_c, K_c >_F < Y_c Y_c', Y_c Y_c' >_F}} \quad (10)$$

Here the centered matrix $K_c$ associated to the matrix $K$ is defined by:

$$\begin{aligned} K_c(x,y) &= \langle \Phi(x) - E_x[\Phi], \Phi(x') - E_{x'}[\Phi] \rangle \\ &= K(x,y) - E_x[K(x,y)] - E_y[K(x,y)] \\ &\quad + E_{x,y}[K(x,y)] \end{aligned}$$

where the expectation operator is evaluated by averaging over the data. $Y_c$ is defined by:

$$Y_c = Y - E[Y]$$

So the Centered Alignment Gap is defined by:

$$Gap_{CA}(k) = E_n\{CA/H_0\} - CA \quad (11)$$

The estimated number of clusters $\hat{k}$ is the $k$ such that $Gap_{CA}(k)$ is maximum.

- Delta Level Gap
  In the dendrogram, we consider each level of the similarity measure $h_i, i = 1...n-1$ where $h_1$ is the highest level. Then we define $\Delta h(k) = h_i - h_{i+1}, k = 2...n$. We define the criterion Delta Level Gap:

$$Gap_{\Delta h(k)}(k) = \Delta h(k) - E_n\{\Delta h(k)/H_0\} \quad (12)$$

  The estimated number of clusters $\hat{k}$ is the value of $k$ where $Gap_{\Delta h(k)}(k)$ is maximum.

- Weighted Delta Level Gap
  This criterion is related to the previous one. We define the Weighted Delta Level (denoted by $\Delta h_W(k)$) by: $\Delta h_W(2) = \Delta h(2), k = 2$ and $\Delta h_W(k) = \frac{\Delta h(k)}{\sum_{i=2}^{k-1} \Delta h(i)}, k \geq 2$. We define the criterion Weighted Delta Level Gap as:

$$Gap_{\Delta h_W(k)}(k) = \Delta h_W(k) - E_n\{\Delta h_W(k)/H_0\} \quad (13)$$

  The estimated number of clusters $\hat{k}$ is the value of $k$ where $Gap_{\Delta h_W(k)}(k)$ is maximum.

# 4 SIMULATIONS

We have generated 6 different data sets to compare the proposed criteria with that from Tibshirani:

1. Five clusters in two dimensions (1)
   The clusters consist of gaussian bidimensional distributions $N(0, 1.5^2)$ centered at (0,0), (-3,3), (3,-3), (3,3) and (-3,-3). Each cluster is composed of 50 observations. Clusters strongly overlap. The kernel parameter is $\sigma = 0.90$.
2. Five clusters in two dimensions (2)
   The data are generated in the same way as in the previous case but the variance of the clusters is now $1.25^2$). Clusters slightly overlap. $\sigma = 0.85$.
3. Five clusters in two dimensions (3)
   Same case but with variance equal to $1.0^2$. There is no overlap. $\sigma = 0.80$.
4. Three clusters in two dimensions
   This is a data set used by Tibshirani (Tibshirani et al., 2001). The clusters are unit variance gaussian bidimensional distributions with $25, 25, 50$ observations, centered at $(0,0)$, $(0,5)$ and $(5,-3)$, respectively. $\sigma = 0.80$.
5. Two nested circles and one outside isolated disk in two dimensions
   These three circles are centered at $(0,0)$, $(0,0)$ and $(0,8)$ with $150, 100, 100$ observations. The respective radii are uniformly distributed over $[0,1]$, $[4,5]$ and $[0,1]$. $\sigma = 0.55$.
6. Two elongated clusters in three dimensions
   This is also a data set used by Tibshirani (Tibshirani et al., 2001) to show the interest of performing PCA to define the null distribution. Clusters are aligned with the first diagonal of a cube. We have $x_1 = x_2 = x_3 = x$ with $x$ composed of 100 equally spaced values between $-0.5$ and $0.5$. To each component of $x$ a Gaussian perturbation with standard deviation 0.1 is added. The second cluster is generated similarly, except for a constant value of 10 which is added to each component. The result is two elongated clusters, stretching out along the main diagonal of a three dimensional cube. $\sigma = 1.00$.

Simulation results with kernel HAC are shown in Table 1. 50 realizations were generated for each case and we used principal component analysis to define the distributions used as the null reference samples. A special scenario using a uniform distribution over the initial area covered by the data (without PCA) is provided in case 7. For every simulation, expectation appearing in equation 7 was estimated over 100 independent realizations of the null distribution.

Table 1: Simulation results using Kernel HAC. Each number represents the number of times each criterion gives the number of clusters indicated in the corresponding column, out of the 50 realizations. The column corresponding to the right number of clusters is indicated in boldface. Numbers between parentheses indicate the results obtained using standard HAC, when of some interest. NF stands for Not Found.

| Number of clusters | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | **NF** |
|---|---|---|---|---|---|---|---|---|---|---|
| 1. Five clusters in two dimensions[(1)] | | | | | | | | | | |
| Gap Statistic | 0 (48) | 0 (2) | 4 (0) | **28** (**0**) | 15 (0) | 3 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) |
| Modified Gap Statistic | 0 (2) | 0 (0) | 0 (10) | **9** (**36**) | 9 (1) | 5 (0) | 5 (0) | 3 (0) | 19 (0) | 0 (0) |
| Delta Level Gap | 0 (0) | 0 (0) | 18 (9) | **30** (**38**) | 1 (1) | 1 (1) | 0 (0) | 0 (0) | 0 (1) | 0 (0) |
| Weighted Delta Level Gap | 0 (47) | 7 (0) | 20 (1) | **22** (**1**) | 1 (1) | 0 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) |
| Centered Alignment Gap | 6 (1) | 4 (0) | 11 (1) | **23** (**6**) | 6 (10) | 0 (9) | 0 (8) | 0 (9) | 0 (6) | 0 (0) |
| 2. Five clusters in two dimensions [(2)] | | | | | | | | | | |
| Gap Statistic | 0 (48) | 0 (0) | 0 (0) | **43** (**2**) | 6 (0) | 1 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) |
| Modified Gap Statistic | 0 (0) | 0 (0) | 0 (1) | **19** (**45**) | 10 (3) | 6 (1) | 5 (0) | 2 (0) | 8 (0) | 0 (0) |
| Delta Level Gap | 0 (0) | 0 (0) | 2 (1) | **48** (**48**) | 0 (1) | 0 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) |
| Weighted Delta Level Gap | 0 (45) | 3 (0) | 12 (0) | **35** (**5**) | 0 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) |
| Centered Alignment Gap | 0 (0) | 1 (0) | 2 (0) | **40** (**10**) | 5 (15) | 2 (9) | 0 (7) | 0 (4) | 0 (5) | 0 (0) |
| 3. Five clusters in two dimensions [(3)] | | | | | | | | | | |
| Gap Statistic | 0 (39) | 0 (0) | 0 (0) | **48** (**11**) | 2 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) |
| Modified Gap Statistic | 0 (0) | 0 (0) | 0 (0) | **26** (**47**) | 12 (3) | 2 (0) | 0 (0) | 1 (0) | 9 (0) | 0 (0) |
| Delta Level Gap | 0 (0) | 0 (0) | 0 (0) | **50** (**50**) | 0 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) |
| Weighted Delta Level Gap | 0 (30) | 1 (0) | 3 (0) | **46** (**20**) | 0 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (0) |
| Centered Alignment Gap | 0 (0) | 0 (0) | 0 (0) | **50** (**10**) | 0 (18) | 0 (12) | 0 (6) | 0 (3) | 0 (1) | 0 (0) |
| 4. Three clusters in two dimensions | | | | | | | | | | |
| Gap Statistic | 0 | **38** | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Modified Gap Statistic | 0 | **14** | 18 | 3 | 0 | 5 | 4 | 3 | 3 | 0 |
| Delta Level Gap | 5 | **45** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Weighted Delta Level Gap | 0 | **50** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Centered Alignment Gap | 35 | **15** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5. Two nested circles and one outside isolated disk in two dimensions | | | | | | | | | | |
| Gap Statistic | 0 (0) | **0** (**0**) | 0 (38) | 0 (6) | 0 (1) | 0 (3) | 0 (1) | 1 (0) | 0 (0) | 49 (1) |
| Modified Gap Statistic | 0 (0) | **0** (**0**) | 0 (0) | 0 (0) | 0 (0) | 0 (0) | 0 (1) | 0 (0) | 50 (49) | 0 (0) |
| Delta Level Gap | 0 (0) | **50** (**3**) | 0 (42) | 0 (2) | 0 (0) | 0 (1) | 0 (2) | 0 (0) | 0 (0) | 0 (0) |
| Weighted Delta Level Gap | 0 (48) | **7** (**0**) | 0 (2) | 0 (0) | 2 (0) | 8 (0) | 6 (0) | 13 (0) | 14 (0) | 0 (0) |
| Centered Alignment Gap | 50 (12) | **0** (**23**) | 0 (0) | 0 (0) | 0 (1) | 0 (1) | 0 (2) | 0 (3) | 0 (8) | 0 (0) |
| 6. Two elongated clusters in three dimensions | | | | | | | | | | |
| Gap Statistic | **50** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Modified Gap Statistic | **45** | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Delta Level Gap | **50** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Weighted Delta Level Gap | **0** | 0 | 0 | 0 | 0 | 0 | 4 | 10 | 36 | 0 |
| Centered Alignment Gap | **50** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7. Two elongated clusters in three dimensions (without PCA to define the null reference) | | | | | | | | | | |
| Gap Statistic | **0** | 0 | 1 | 0 | 18 | 13 | 15 | 3 | 0 | 0 |
| Modified Gap Statistic | **0** | 0 | 0 | 0 | 1 | 2 | 6 | 7 | 34 | 0 |
| Delta Level Gap | **50** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Weighted Delta Level Gap | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 47 | 0 |
| Centered Alignment Gap | **50** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The kernel parameter $\sigma$ has been selected as the value which maximizes the centered kernel alignment defined by equation 10.

To evaluate the centered kernel alignment, labels must be known. They are obtained as the result of kernel CAH clustering. Coding of the $Y$ is performed in such a way that the centered kernel alignment is invariant to the (arbitrary) cluster number. In this paper, the vector code book is, for an observation $x_i$ belonging to cluster $m, m = 1, \ldots, k$:

$$\begin{cases} y_{ij} = 1, \text{ if } j = m, \\ y_{ij} = -1, \text{ if } j \neq m. \end{cases} \quad (14)$$

Results in Table 1 clearly indicate that one of the proposed criteria Delta Level Gap outperforms other criteria (including the Gap Statistic by Tibshirani) in almost all cases.

Tibshirani (Tibshirani et al., 2001) mainly focused on well-separated clusters. Our simulations also show that the Gap Statistic estimation is not good at identifying the number of clusters when they highly overlap. See data sets 1, 2 and 3 in Table 1: the lower the overlap, the better the performances. Looking at the example of data set 1 in Table 1, all criteria do not give the expected results. The number of clusters estimated by Delta Level Gap mostly give 4 and 5 clus-

ters, which is better. However, all the criteria suffer from overlap.

Another important conclusion is the importance of the choice of an appropriate null reference distribution. Seeing scenarios 6 and 7 in Table 1, results vary a lot between the uniform distribution and the uniform distributions aligned with principal components. This gives some room to improve our framework by choosing a more appropriate null reference distribution.

We have also done simulations using standard HAC on some of these data sets. Comparing the results obtained show that kernel HAC generally improves the performances. We can observe that Centered Alignment Gap, Gap Statistic and Weighted Delta Level Gap do not perform well on examples $1, 2$ and 3 for standard HAC. Evolution of these statistics as functions of the number of clusters, that can be seen in Figures 6, 7 and 8, explain these results. Figure 6 shows the evolution of the Gap Statistic as a function of the number of clusters $k$ for a realization of the second data set. As can be seen, the criterion initially proposed gives $k = 2$. Figure 7 represents the evolution of Centered Alignment Gap. The behavior is rather erratic and explains the variations in the estimated number of clusters. This can also be observed for Weighted Delta Level Gap shown in Figure 8.

Furthermore, Delta Level Gap shows excellent performances on data set 5, which cannot be classified using non kernel HAC while easily separated by kernel HAC, as can be seen from Table 1.

According to all experiments we have done (not all of them are presented here), we have observed that the initial Gap Statistic of Tibshirani is not always efficient in estimating the right number of clusters. Looking for the maximum value instead of selecting the value proposed as in equation 8, appears to be a possible alternative. However, the Delta Level Gap seems to be the most reliable estimate among those we have studied and potentially one of the less sensitive to the null distribution of the data.

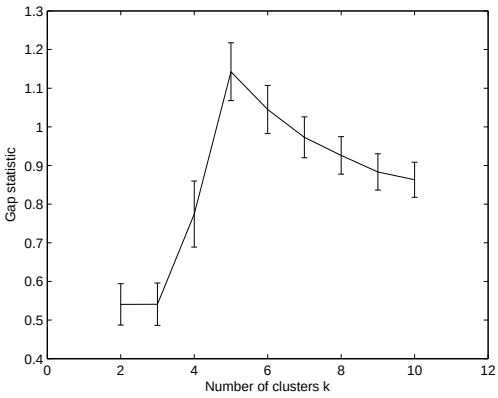


Figure 6: Number of clusters using Gap Statistic.

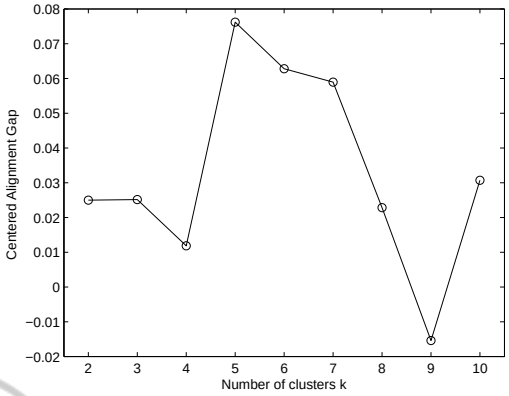


Figure 7: Number of clusters using Centered Alignment Gap.

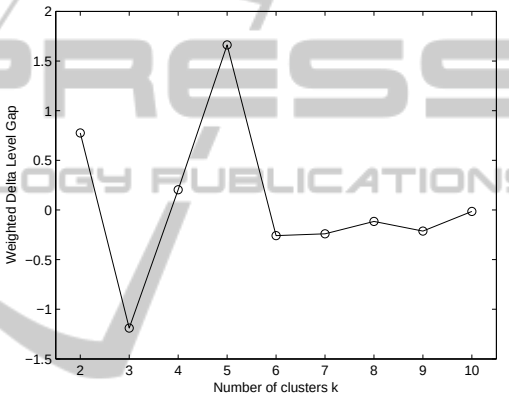


Figure 8: Number of clusters using Weighted Delta Level Gap.

## 5 CONCLUSIONS

In this paper, we have proposed Kernel HAC as a clustering tool. It is robust for clustering separable classes which have a small between-class dispersion in the input space. Using an adequate Gap Statistic, it also allows determination of the number of clusters in the data. Out of them, simulation results have shown that Delta Level Gap, one of our criteria, outperforms the conventional Gap Statistic in many cases. Our future work will focus on new methods for determining the optimal kernel parameter. A few work has been done which has already shown good prospects. Then, kernel engineering (adaptation of the kernel function to the data) will be considered.

## REFERENCES

Aizerman, A., Braverman, E. M., and Rozoner, L. (1964). Theoretical foundations of the potential function

method in pattern recognition learning. *Automation and remote control*, 25:821–837.

Barbará, D. and Jajodia, S. (2002). *Applications of data mining in computer security*, volume 6. Springer.

Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203.

Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Driver, H. E. and Kroeber, A. L. (1932). *Quantitative expression of cultural relationships*. University of California Press.

Filippone, M., Camastra, F., Masulli, F., and Rovetta, S. (2008). A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190.

Fraley, C. and Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, 41(8):578–588.

Girolami, M. (2002). Mercer kernel-based clustering in feature space. *Neural Networks, IEEE Transactions on*, 13(3):780–784.

Gordon, A. D. (1996). Null models in cluster validation. In *From data to knowledge*, pages 32–44. Springer.

Harman, H. H. (1960). Modern factor analysis.

Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. Wiley-Interscience.

Kim, D.-W., Lee, K. Y., Lee, D., and Lee, K. H. (2005). Evaluation of the performance of clustering algorithms in kernel-induced feature space. *Pattern Recognition*, 38(4):607–611.

Krzanowski, W. J. and Lai, Y. (1988). A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*, pages 23–34.

Mac Queen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209:415–446.

Milligan, G. W. and Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179.

Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *Neural Networks, IEEE Transactions on*, 12(2):181–201.

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359.

Olson, C. F. (1995). Parallel algorithms for hierarchical clustering. *Parallel computing*, 21(8):1313–1325.

Príncipe, J. C., Liu, W., and Haykin, S. (2011). *Kernel Adaptive Filtering: A Comprehensive Introduction*, volume 57. John Wiley & Sons.

Qin, J., Lewis, D. P., and Noble, W. S. (2003). Kernel hierarchical gene clustering from microarray expression data. *Bioinformatics*, 19(16):2097–2104.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319.

Shawe-Taylor, N. and Kandola, A. (2002). On kernel target alignment. *Advances in neural information processing systems*, 14:367.

Sneath, P. H., Sokal, R. R., et al. (1973). *Numerical taxonomy. The principles and practice of numerical classification.*

Sugar, C. A. and James, G. M. (2003). Finding the number of clusters in a dataset. *Journal of the American Statistical Association*, 98(463).

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

Vapnik, V. (2000). *The nature of statistical learning theory*. springer.