Kernel MDL to Determine the Number of Clusters

Ivan O. Kyrgyzov[1], Olexiy O. Kyrgyzov[2], Henri Maître[1], and Marine Campedel[1]

[1] Competence Centre for Information Extraction and Image Understanding for Earth Observation, GET/Télécom Paris - LTCI, UMR 5141, CNRS 46, rue Barrault, 75013, Paris, France
name[1]@enst.fr

[2] Department of Computer Science and Electrical Engineering, OGI School of Science and Engineering, Oregon Health and Science University, 20000 NW Walker Road, Beaverton, OR, USA, 97006 name[2]@csee.ogi.edu

Abstract. In this paper we propose a new criterion, based on Minimum Description Length (MDL), to estimate an optimal number of clusters. This criterion, called Kernel MDL (KMDL), is particularly adapted to the use of kernel K-means clustering algorithm. Its formulation is based on the definition of MDL derived for Gaussian Mixture Model (GMM). We demonstrate the efficiency of our approach on both synthetic data and real data such as SPOT5 satellite images.

1 Introduction

We are interested in knowledge extraction from a SPOT5 satellite image database. One of our tasks is to find categories of images and to classify them without prior knowledge on the type or number of these categories. Considering the amount of available data we are concerned in using simple, fast and efficient clustering algorithms. K-means is one of them but suffers from several drawbacks: i) it cannot adapt to any cluster shape ii) the knowledge of number of clusters is necessary iii) the result strongly depends on the initialization process.

To answer the first problem, a classical solution is to use Kernel K-means algorithm [9] [14]. During the last decade kernel-based algorithms attracted lots of researchers who applied them to various tasks such as machine learning, pattern recognition, computer vision, *etc.* The success of these approaches is related to the fact that using a kernel (see definition and properties of kernel in [13] [14]) is equivalent to defining a feature space transform; the resulting feature space is tuned to simplify the classification process and allows efficient classical algorithms (like K-means) processing. This feature space depends on kernel parameter(s); several approaches are proposed in the literature to determine the optimal parameter(s) [3]: in this work we use one kernel with fixed parameter.

To answer the second and third problems we propose to use a standard approach such as selection of a clustering solution obtained using different number of clusters and initializations. This selection is based on the minimum of

our KMDL criterion. It allows us to stabilize clustering results and to have a smoothed KMDL curve.

Our proposition about using MDL criteria to determine the number of clusters is based on several arguments. Firstly, MDL is able to give access to an optimal code or an optimal data representation for a certain model of data [10], *e.g.* for GMM in our case. Secondly, this criterion works well when lots of data are available [6]. This is our case because we have a huge storage of satellite images. Finally, in the literature we have not found previous works about applying MDL criteria to Kernel K-means to find the optimally associated number of clusters. It gives us the motivation to formulate MDL criteria for Kernel K-means clustering.

We revise the main definition of MDL for GMM and we show a simplification of MDL through the complete log-likelihood of GMM in Sect. 2. The objective function for Kernel K-means is presented in Sect. 3. Then we formulate KMDL in Sect. 4 using the simplified MDL for GMM. Results on synthetic data and real satellite images are presented in Sect. 5 and Sect. 6, respectively. Conclusions are in Sect. 7.

2 MDL for the Gaussian Mixture Model

2.1 Gaussian Mixture Model

The finite mixture model is widely used to represent data in statistical pattern recognition. Let $X = \{X_1, ..., X_I\}$ denote the data set of samples X_i, where each X_i is a vector $X_i = (X_{i1}, ..., X_{iD})$ of feature values X_{id}. The set X is modelled by a finite mixture model consisting of two parts [10]:

1. the prior probability $P(X_i \in j \mid \Theta_j) = \alpha_j$ that every sample X_i is a member of only one mixture component j, $(j = 1, ..., J)$, where $\alpha_j = n_j/I$, (n_j denoting the number of samples belonging to the mixture component j);
2. the conditional probability modelling each component j by the parameterized probability density function (pdf) $P_j(X_i \mid \Theta_j)$, where Θ_j denotes the parameter set.

Let $P_j(X_i \mid \Theta_j)$ denote the class-probability of observing the sample X_i conditional to X_i belonging to the component j. The finite mixture model expresses the probability of observing the sample X_i as a sum of pdf:

$$P(X_i \mid \Theta) = \sum_{j=1}^{J} \alpha_j P_j(X_i \mid \Theta_j)\,. \tag{1}$$

An important sub-class of mixture models is the multivariate Gaussian distribution, based on a Gaussian class-distribution:

$$P_j(X_i \mid \Theta_j) = \mathcal{N}(X_i \mid \mu_j, \Sigma_j) = \frac{e^{-\frac{1}{2}\left((X_i-\mu_j)^T \Sigma_j^{-1}(X_i-\mu_j)\right)}}{(2\pi)^{D/2} \mid \Sigma_j \mid^{1/2}}\,, \tag{2}$$

where μ_j and Σ_j are the mean and the covariance matrix of the j^{th} component, respectively. Estimates of the j^{th} mean and covariance matrix are classically obtained as:

$$\mu_j = \frac{1}{n_j}\sum_{l=1}^{n_j} X_l \,, \tag{3}$$

$$\Sigma_j = \frac{1}{n_j}\sum_{l=1}^{n_j}(X_l - \mu_j)^T(X_l - \mu_j) \,, \tag{4}$$

where $X_l \subseteq j$.

With the assumption that the data instances X_i are independently distributed, the joint data probability (probability of observing data set X or likelihood function) is the product of the individual instance probabilities:

$$P(X \mid \Theta) = \prod_{i=1}^{I}\sum_{j=1}^{J} \alpha_j P_j(X_i \mid \Theta_j) \,. \tag{5}$$

The Expectation-Maximization (EM) algorithm [10] can be used to estimate the optimal parameters Θ_j of GMM. Without loss of generality we say that the j^{th} component of GMM models the j^{th} cluster.

The purpose of clustering data is to simplify their representation in the feature space by replacing each sample by a generic class which is likely to express all the properties of the samples. However, when substituting a sample by its model, an error is introduced. The more complex the model, the less the error. The "model complexity" is well expressed by the number of parameters needed to build the model. In the mixture of Gaussians case where every cluster is given by its mean (3) and its covariance matrix (4), the more clusters are used, the more complex the model is, and the less error between data and model. A method to choose the optimal number of clusters consists in selecting the number that most efficiently codes the data, i.e. that provides the shortest description when representing the samples using models and the errors to the model. This method, named Minimum Description Length (MDL), was proposed by Rissanen [2], [11], [12]. MDL is defined as [12]:

$$\min_{\mathbb{k},\Theta} -log(P(X|\Theta)) + \frac{1}{2}\mathbb{k}log(I) \,, \tag{6}$$

where $log(P(X \mid \Theta))$ is the log-likelihood of the mixture model (5) and $\frac{1}{2}\mathbb{k}log(I)$ is a penalty function with $\mathbb{k}$ parameters.

2.2 MDL for the Complete Log-likelihood of GMM

Let see the log-likelihood for the mixture of Gaussian distributions in more details. To complete the likelihood $P(X|\Theta)$ (5) of the finite mixture expressed by (1), we should introduce the hidden variable z which attribute any sample to a class: $z = \{z_1, ..., z_i, ..., z_I\}$ [4] [5]. Label z_i is coded as a binary vector

$z_i = [z_{i1}, ..., z_{ij}, ..., z_{iJ}]$, where $z_{ij} = 1$ if sample i belongs to cluster j, or 0 if not. Using (5), the complete log-likelihood $log(P(X, z|\Theta))$ becomes [4] [5]:

$$\begin{aligned} &log\left(P(X, z \mid \Theta)\right) = log\left(\prod_{i=1}^{I}\sum_{j=1}^{J} z_{ij}\alpha_j P_j(X_i \mid \Theta_j)\right) = \\ &\sum_{i=1}^{I} z_{ij} log(\alpha_j P_j(X_i \mid \Theta_j))\,. \end{aligned} \tag{7}$$

By substituting the multivariate Gaussian distribution $P_j(X_i \mid \Theta_j)$ (2) in the complete log-likelihood (7), we obtain:

$$\begin{aligned} &\sum_{i=1}^{I} z_{ij} log(\alpha_j \mathcal{N}(X_i \mid \mu_j, \Sigma_j)) = \sum_{i=1}^{I} z_{ij} log\left(\alpha_j \frac{e^{-\frac{1}{2}\left((X_i-\mu_j)^T \Sigma_j^{-1}(X_i-\mu_j)\right)}}{(2\pi)^{D/2} \mid \Sigma_j \mid^{1/2}})\right) = \\ &\sum_{i=1}^{I} z_{ij}\left(log\left(\frac{\alpha_j}{\mid \Sigma_j \mid^{1/2}}\right) - \frac{D}{2} log(2\pi) - \frac{1}{2}\left((X_i-\mu_j)^T \Sigma_j^{-1}(X_i-\mu_j)\right)\right) = \\ &\frac{1}{2}\sum_{i=1}^{I} z_{ij} log\left(\frac{\alpha_j^2}{\mid \Sigma_j \mid}\right) - \frac{1}{2}\sum_{i=1}^{I} z_{ij} D \log(2\pi) \\ &-\frac{1}{2}\sum_{i=1}^{I} z_{ij}\left((X_i-\mu_j)^T \Sigma_j^{-1}(X_i-\mu_j)\right)\,. \end{aligned} \tag{8}$$

In this equation, some terms are constant:

$$-\frac{1}{2}\sum_{i=1}^{I} z_{ij} D \log(2\pi) = -\frac{1}{2}\sum_{j=1}^{J} n_j D \log(2\pi) = -\frac{1}{2} ID \log(2\pi) = const_1\,. \tag{9}$$

Moreover, to calculate the matrix Σ_j (4) the only samples from the cluster j are needed, therefore:

$$-\frac{1}{2}\sum_{i=1}^{I} z_{ij}\left((X_i-\mu_j)^T \Sigma_j^{-1}(X_i-\mu_j)\right) = -\frac{1}{2}\sum_{j=1}^{J} n_j DI = -\frac{DI^2}{2} = const_2\,. \tag{10}$$

Then, the complete log-likelihood $log(P(X, z|\Theta))$ (7) may be written as:

$$\frac{1}{2}\sum_{i=1}^{I} z_{ij} log\left(\frac{\alpha_j^2}{\mid \Sigma_j \mid}\right) + const = \frac{1}{2}\sum_{j=1}^{J} n_j log\left(\frac{\alpha_j^2}{\mid \Sigma_j \mid}\right) + const\,. \tag{11}$$

In the right part of the MDL definition (6), $\mathbb{k}$ is the model free parameters number. In case of Gaussian mixture model free parameters are:

- $J-1$ parameters for J weights α_j (since $\sum \alpha_j = 1$);
- D parameters for each mean μ_j;
- $D(D+1)/2$ parameters for each covariance matrix Σ_j.

Therefore, the number of free parameters is:

$$\mathbb{k} = J - 1 + J(D + D(D+1)/2) = J(D^2 + 3D + 2)/2 - 1\,. \tag{12}$$

Using the complete log-likelihood (11) and the free parameter number of (12), the description length (6) of Gaussian mixture model with J clusters is:

$$-\frac{1}{2}\sum_{j=1}^{J} n_j log\left(\frac{\alpha_j^2}{\mid \Sigma_j \mid}\right) + (J(D^2+3D+2)/2 - 1)log(I)/2 + const\,. \tag{13}$$

The *const* term having no influence on MDL for different cluster numbers and as $\alpha_j = n_j/I$, we may minimize:

$$\Lambda = -\sum_{j=1}^{J} n_j log\left(\frac{n_j^2}{\mid \Sigma_j \mid}\right) + J(D^2+3D+2)log(I)/2\,. \tag{14}$$

Equation (14) shows that a quality of clustering only depends on the weighted determinants of the covariance matrices which express the square errors between data and model. Estimating the covariance matrices Σ_j and the populations of each cluster n_j, we can draw the MDL curve Λ as a function of the cluster number J. The minimum on this curve indicates the optimal description of the data set X, i.e. the minimum error with the minimum model complexity.

The MDL criterion (14) may be applied to any clustering method: to EM, which, as said before, provides the best clustering, given a number of clusters, or to simpler algorithms - like K-means which may be seen as a simplified version of EM [10], or Kernel K-means, which is an extension of K-means. Based on this remark, we propose first to define an MDL optimization of Kernel K-means.

3 Kernel K-means Algorithm

In the case where data have a complex structure (e.g. data are non linearly separable), a direct application of K-means is not suit because of its tendency to group data into globe-shaped clusters [10]. To solve this problem, data may be mapped by a transformation into a new feature space where samples are linearly separable [14]. The transformation is defined by a kernel $K(\cdot)$ as the inner product:

$$K(X_k, X_l) = \langle \phi(X_k), \phi(X_l) \rangle\,, \tag{15}$$

where $\phi(\cdot)$ is a mapping of X to an inner product feature space [14] and k, l take values $[1, ..., I]$. The simplest kernel is a linear:

$$K(X_k, X_l) = X_k X_l\,, \tag{16}$$

and one of the frequently used kernels is the Gaussian kernel:

$$K(X_k, X_l) = e^{-\frac{\|X_k - X_l\|^2}{2\sigma^2}}, \tag{17}$$

where σ is a kernel parameter. Kernel K-means minimizes an optimization function on the transformed data space [14]:

$$\min \sum_{j=1}^{J} \sum_{k \subseteq j} \| \phi(X_k) - \bar{\phi}(X_k) \|^2, \tag{18}$$

where $\bar{\phi}(X_k) = \frac{1}{n_j} \sum_{X_k \subseteq j} \phi(X_k)$ is the j^{th} cluster mean. One of the advantages of using the kernel function is that we can solve (18) (*e.g.* for the Gaussian kernel (17) without the explicit representation of function $\phi(\cdot)$. The distance $\| \phi(X_k) - \bar{\phi}(X_k) \|^2$ may be calculated with the inner product $\langle \phi(\cdot)\phi(\cdot) \rangle$. With this objective, the standard steps of K-means algorithm are applied [14]. As can be seen Kernel K-means algorithm is equal to K-means when the linear kernel (16) is used.

4 Kernel MDL

Taking advantage of the formulation of (14), we propose to derive now a more general form for MDL.

From (14) it has been said that the simplified MDL is depending on the determinants of the $| \Sigma_j |$ matrices which describe the model to data error. This error may be determined in the original space X, as well as in the transform space after kernel transformation. Therefore, we propose to define a general MDL, similar to (14), as:

$$-\sum_{j=1}^{J} n_j log \left(\frac{n_j^2}{Dist(X_k, X_l | k, l \subseteq j)} \right) + P(J, D, I) \tag{19}$$

where $Dist(X_k, X_l | k, l \subseteq j)$ is the error function for sample X_k being represented by the j^{th} cluster (for instance, the distance between X_k and the mean of cluster j) and $P(J, D, I)$ is a penalty function.

The simplest error function is the Euclidean distance which may be calculated using the kernel K (15). The sum-squares distances from patterns to their corresponding j^{th} cluster centroid was presented in [14] as the optimization function for Kernel K-means:

$$S_j = \frac{1}{n_j D} \sum_{k \subseteq j} \left(K(X_k, X_k) - \frac{1}{n_j} \sum_{l \subseteq j} K(X_k, X_l) \right). \tag{20}$$

In case where K is the linear kernel, S equals the variance in the original space X as expressed by (16). To obtain the complete MDL formulation of (14),

supposing the variances of a cluster equal for each dimension, we may rewrite the determinant of covariance matrix Σ_j as:

$$\mid \Sigma_j \mid = S_j^D \,. \tag{21}$$

As the error S_j (20) may be derived for any kernel, *e.g.* Gaussian (17), we may substitute the determinant (21) in the MDL expression (14) to obtain the kernel MDL:

$$\text{KMDL} = -\sum_{j=1}^{J} n_j log\left(\frac{n_j^2}{S_j^D}\right) + J(D^2+3D+2)log(I)/2\,. \tag{22}$$

For the following experiments the same penalty function as in (14) have been used. The derivation of an alternative penalty is not addressed in this paper. One of the main advantages of this formulation lies in that the explicit mean of a cluster j is not needed. This point is important when this mean has no physical meaning, as it is often the case for non-convex clusters. To calculate MDL criterion for the mixture of Gaussians in the original space X the distance between samples and the nearest cluster centroid must be calculated. Problems may appear in case of data distributed on clusters with holes as in Fig. 1-d.

5 Experiments with synthetic data

We tested our approach on synthetic data before applying it to real data such as satellite images. The simplest and often used example of synthetic data are using Gaussian distributions where each distribution is a cluster. When working on satellite images, we expect to have a large number of clusters because of the great variety of possible scenes. Therefore we demonstrate the potential of the method with a rather large number of clusters, larger than in the usual literature [8]. We make use of 20 Gaussian distributions as presented in Fig. 1-a with 100 samples per cluster. EM algorithm run 20 times for each cluster number, with a different random initialization. Two curves are presented in Fig. 1-b, showing the results of clustering using either MDL (14) or KMDL (22) with Gaussian kernel and parameter $\sigma = 2$. For all curves of KMDL a constant is added to better visualise with MDL. As expected, both curves exhibit a well defined minimum, with an optimal number of clusters equals to 20.

The same experiments were done for another toy example having clusters with a complex structure. Points of this cluster are distributed on a circle. Here again, EM-algorithm and Kernel K-means with Gaussian kernel ($\sigma = 0.5$) have been used. Optimal results are presented Fig. 1-c and Fig. 1-d. From Fig. 1-e, it may be observed that EM with MDL detects more clusters than expected because of the difficulty to linearly separate a cluster with a complex structure (also seen in Fig. 1-c where the circle is split into 4 clusters). On the contrary Kernel K-means with the Gaussian kernel optimally separates the mixture in Fig. 1-d, and KMDL determinates the true number of clusters.

The last experiment concerns two real world data sets Iris and Thyroid taken from the UCI machine learning repository. Iris data contain 3 classes, 50 samples

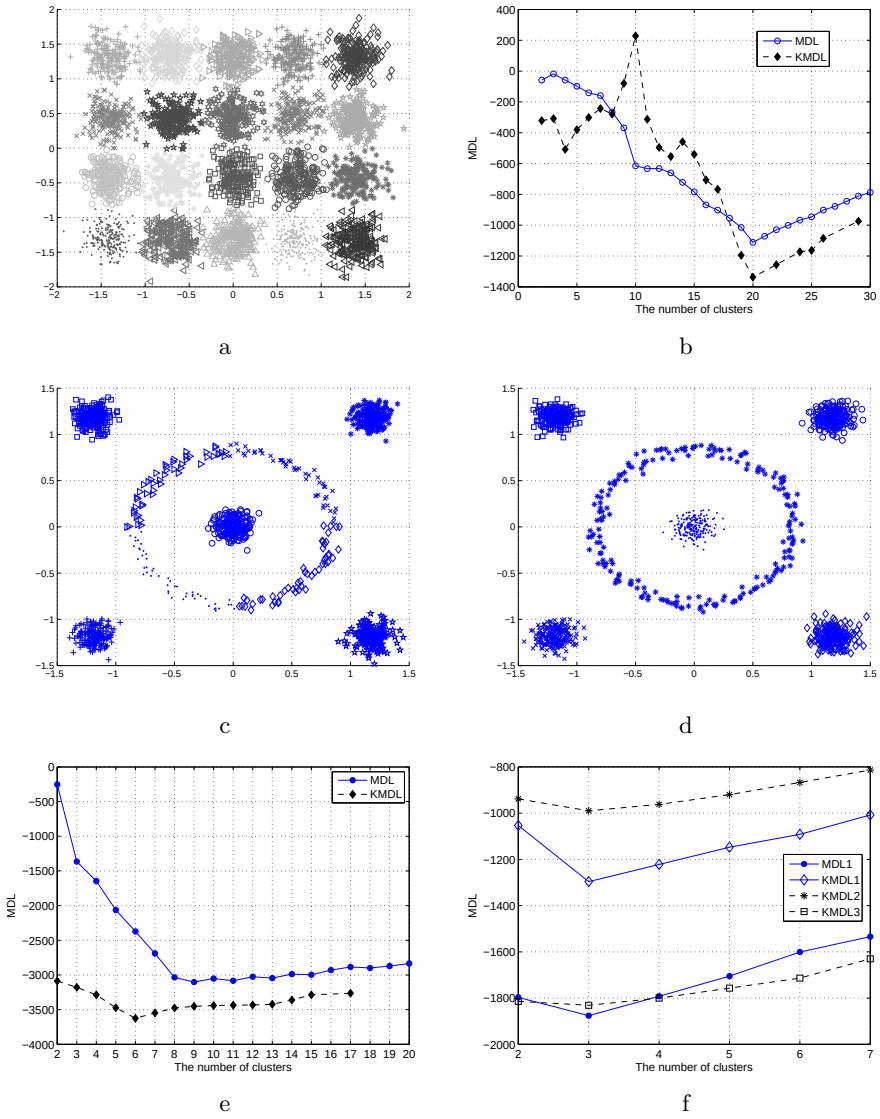


Fig. 1. Synthetic examples. In a: synthetic example 1 with 20 clusters. In b: results on clustering example 1. Detection of the optimal number of clusters by MDL (14) (solid line) and by KMDL (22) (dashed line). In c: example 2 with a circular cluster as clustered by EM. In d: the same as clustered by Kernel K-means. In e: curves drawn for example 2. In f: Optimal number of clusters for Thyroid and Iris data. MDL (14) (solid line with points) and KMDL1 (22) with $\sigma = 5$ (solid line with diamonds) propose 3 as an optimal number of clusters for Thyroid data set. KMDL2 (22) with (16) (dashed line with stars) and KMDL3 (22) with (17) $\sigma = 4$ (dashed line with squares) propose 3 as an optimal number of clusters for Iris data set.

per class and 4 features per sample. The minimum of KMDL (22) with the linear kernel (16) and the Gaussian kernel (17) determines the true number of clusters as three Fig. 1-f. Thyroid data have 3 classes: 150, 35 and 30 samples per class, respectively, and 5 features per sample. Both criteria KMDL (22) with the Gaussian kernel (17) and MDL (14) determine the true number of clusters as three Fig. 1-f.

From this set of experiments, several practical rules have been observed. At first, it seems that it is better to start from high values of cluster number to progressively reduce it in order to have a less chaotic behaviour of the curve. Then we observe that the MDL is often unequivocal, allowing to use speeding search techniques like dichotomy for instance.

6 Experiments with real data: satellite images

6.1 The experiment

In the framework of the CNES-DLR Competence Centre we are interested in information extraction and image understanding for Earth observation with high resolution images [1]. In order to reduce the amount of information carried by an image, we propose to categorize satellite images. To avoid bias and omissions due to human expertise, we investigate unsupervised image category extraction. In this scope we consider each cluster as a category. The optimal number of clusters obtained from a given set of images is therefore an important clue which cannot be arbitrarily fixed. The previous approach (with simplified MDL (14) and KMDL (22)) will be our guideline to determine this number.

We are working with images from the SPOT 5 satellite, they are panchromatic images with a ground resolution of 5m per pixel. Each original image is very large (12000×12000 pixels) and quite complex; therefore we extract smaller images (1024×1024 pixels) with rather homogeneous content on urban areas. These (1024×1024) images will, from now on, be named "the images" since the original large images will no longer be used in the rest of this document. The images represent 6 cities: Copenhagen (Denmark), Istanbul (Turkey), Los Angeles (USA), La Paz (Mexico), Madrid (Spain), Paris (France). We assume that because of geography, culture and history each image has different surface textures. Sub-samples of images are presented in Fig. 2. From these images, we form a database of samples by cutting each image into 400 samples, each of size 64×64 pixels. Samples overlap by 13 pixels. The composed database contained 2400 samples, 6 cities and 400 samples per city. From each sample, 202 features have been extracted: statistics issued from Quadratic Mirror Filters filtering, statistics from Gabor filters, statistics from Haralick co-occurrence matrix descriptors and geometrical features. 15 features were automatically selected from the initial features using unsupervised feature extraction [9].

The data matrix of size 2400×15 is clustered with two algorithms: EM-algorithm [10] with GMM and Kernel K-means [14] with the Gaussian kernel (17) and parameter $\sigma = 15$. 50 random initializations were performed and the

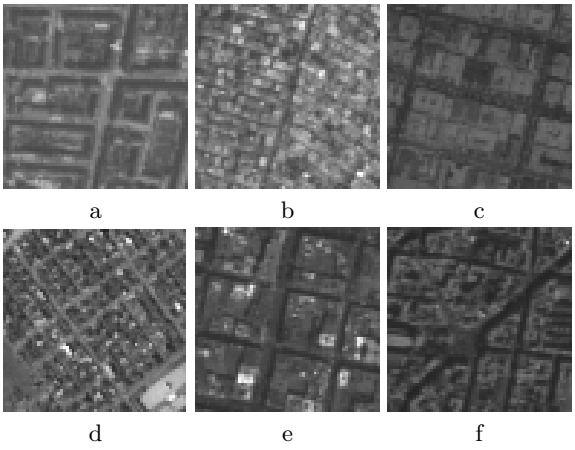


Fig. 2. Samples of SPOT5 images (64 × 64 pixels per sample) : a - Copenhagen (Denmark), b - Istanbul (Turkey), c - Los Angeles (USA), d - La Paz (Mexique), e - Madrid (Spain), f - Paris (France). ©Copyright CNES

best clustering was chosen. In our experiments the data were normalised in a such a way that their mean equals 0 and the standard deviation of each column is 1, so that the weight of each feature be the same.

$$\mu_d = \frac{1}{I}\sum_{i=1}^{I} X_{id}\,, \tag{23}$$

$$\sigma_d = \sqrt{\frac{1}{I}\sum_{i=1}^{I}\left(X_{id}-\mu_d\right)^2}\,, \tag{24}$$

$$\tilde{X}_{id} = \frac{X_{id}-\mu_d}{\sigma_d} \tag{25}$$

Setting in (17) σ as the data dimension ($\sigma = D$), we obtain the curves shown in Fig. 3 for MDL and for KMDL (22). For EM-algorithm the optimal number of clusters is 9 whereas for Kernel K-means it is 11. We may present these optimal clusterings as distribution matrices (as in Tables 1 and 2 respectively), where each column corresponds to a city in the same order as in Fig. 2, and each line represents a cluster.

6.2 Discussion

In the ideal case, where all the cities would be perfectly different, we could consider that the clustering is good if each cluster consists of one city only. From the classification matrices Tables 1 and 2 we can see that the EM-algorithm and

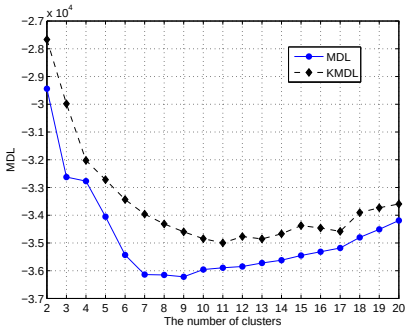


Fig. 3. Detection of the optimal number of clusters by MDL (solid line) and KMDL (dashed line) criteria for SPOT 5 image textures.

Kernel K-means give almost the same clusters. But EM-algorithm finds cluster 4 as a mixture of two cities (Los Angeles and Paris), although these cities exhibit rather different structures Fig. 2. The classification matrix of Kernel K-means (Table 2) shows that these two cities are separated (clusters 3 and 8). Even if we set the number of clusters to 12 for the EM-algorithm the confusion between these cities remains. This confusion disappears when the number of clusters is 15, but it will not be an optimal clustering in terms of MDL. We consider that Kernel K-means better clusters data than EM-algorithm because clusters better correspond to cities. Some texture examples of clustered cities (4 textures per

Table 1. Clustering matrix for 6 cities with EM-algorithm

	Cities						
Clusters	Copenhagen	Istanbul	Los Angeles	La Paz	Madrid	Paris	$\sum$
1	2	3	2	4	155	6	172
2	117	14	0	0	0	0	131
3	86	131	1	0	5	6	229
4	6	3	253	20	24	251	557
5	131	221	0	0	0	0	352
6	0	0	5	256	7	32	300
7	28	11	7	20	32	48	146
8	30	17	132	4	177	56	416
9	0	0	0	96	0	1	97
	400	400	400	400	400	400	

cluster) by Kernel K-means are presented in Tables 3 and 4. The samples closest from the centre of the corresponding clusters have been chosen. Each row of Table 3 has 4 texture examples for clusters from 1 to 6 and Table 4 for clusters from 7 to 11. We analyze visually this examples using classification matrix in

Table 2. The first and sixth rows of Table 3 correspond to 4 textures of La Paz. These clusters show two different surfaces for this city. The second row has samples from every city and corresponds to large places which are likely to be similar almost everywhere around the world. The third column is a typical examples of Paris city blocks and we see from the classification matrix in Table 2 that cluster 3 collects nearly all samples of this city. Cluster 4 has mixed samples from Istanbul and Copenhagen with a domination of Istanbul (see cluster 4 in Table 2). These textures represent both urban and rural areas. Cluster 5 has also similar urban textures from these cities but with a domination of Copenhagen. Cluster 7 in Table 4 has mainly textures from Madrid but also from other cities. Los Angeles is represented by cluster 8 with its typical square streets. Half textures of Madrid are represented by cluster 9. Dense areas of Istanbul correspond to cluster 10. Cluster 11 has textures which contain wide roads. From this early

Table 2. Clustering matrix for 6 cities with Kernel K-means algorithm

	Cities						
Clusters	Copenhagen	Istanbul	Los Angeles	La Paz	Madrid	Paris	$\sum$
1	0	0	0	94	0	1	95
2	28	10	6	22	31	49	146
3	0	0	19	24	9	259	311
4	67	123	1	0	4	6	201
5	112	27	0	0	1	0	140
6	0	0	4	252	5	28	289
7	20	16	72	4	172	34	318
8	13	2	296	0	35	19	365
9	2	2	2	4	142	4	156
10	114	208	0	0	1	0	323
11	44	12	0	0	0	0	56
	400	400	400	400	400	400	

interpretation of classification results, we are quite satisfied by the way the textures have been grouped and the homogeneity of the obtained classes. Results of clusterings in Tables 1 and 2 show that several clusters have redundant information. It means that for different clusterings there are clusters which have the same samples. It will be useful for data mining to combine samples that always belong to common clusters that may reduce redundant information and find some interesting particular clusters in data [7].

7 Conclusions

In this paper we proposed a new criterion called Kernel MDL (KMDL) to estimate the optimal number of clusters for the Kernel K-means algorithm. This criterion is derived from a simplified formulation of the classical MDL for the

Table 3. Texture examples of clusters, Kernel K-means

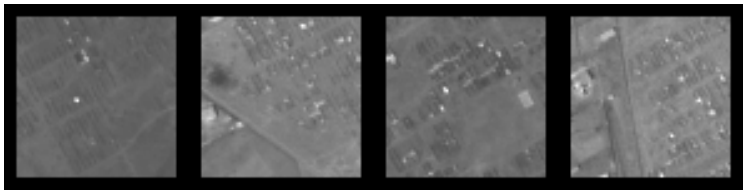
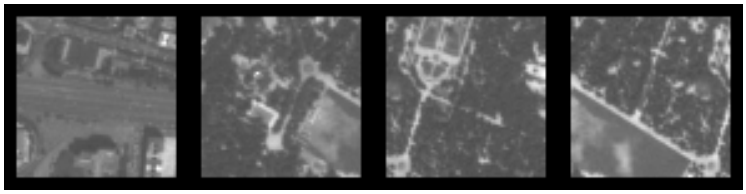
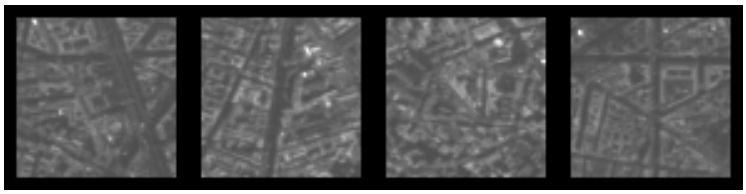
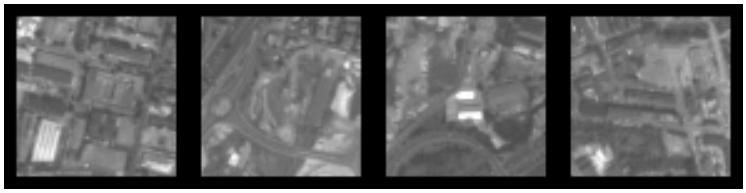
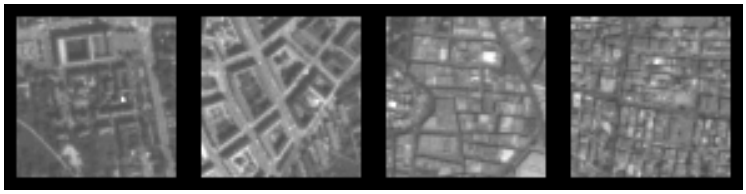
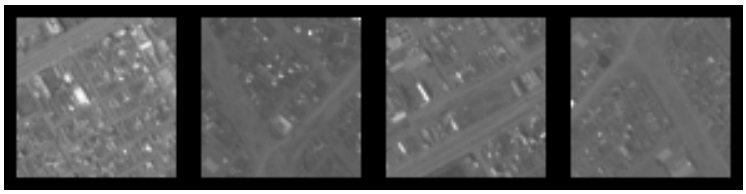
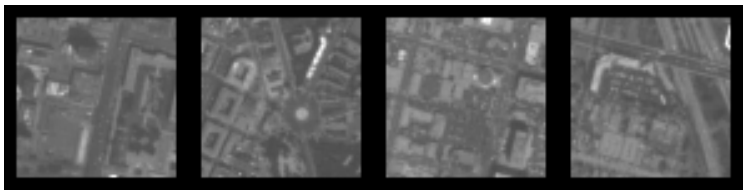
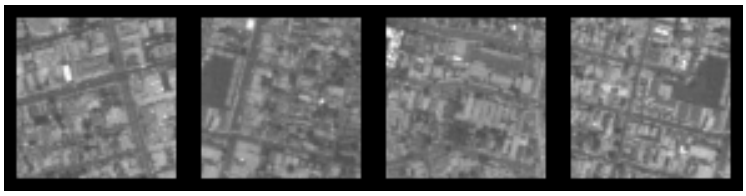
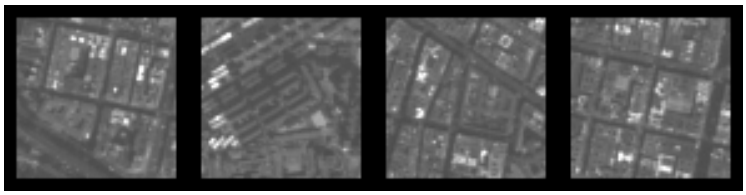
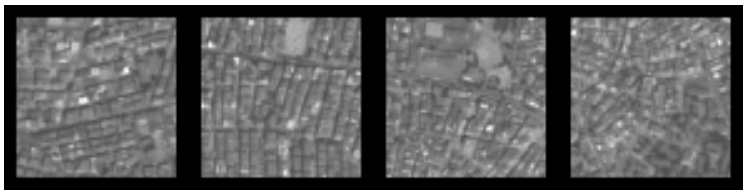
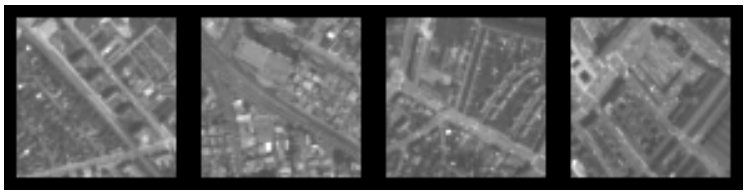
Clusters	Texture examples
1	
2	
3	
4	
5	
6	

Table 4. Texture examples of clusters, Kernel K-means

Clusters	Texture examples
7	
8	
9	
10	
11	

Gaussian Mixture Model. Both KMDL and the simplified MDL allow to determine the optimal number of clusters using simply the error function between the data and the model of clusters. To adapt the criterion to the Kernel K-means algorithm we defined this error function as the corresponding optimized criterion.

The error can be calculated on the kernel function with the Kernel K-means algorithm. The advantage of this approach is that Kernel K-means can linearly separate data which are non linearly separable in the original space. As we can see from experimental results the two criteria MDL and KMDL work well and give optimal numbers of clusters each for its own algorithm. Kernel K-means algorithm with KMDL shows superior results than EM with MDL for synthetic data as well as real data.

Acknowledgements: This study[1] was done with the financial support of Centre National d'Etudes Spatiales (CNES-France). The authors[1] would like to thank M. Datcu and O. Cappé for fruitful discussions.

References

1. http://www.coc.enst.fr/.
2. A. Barron, J. Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *IEEE Trans. Inform. Theory*, 44(6):2743–2760, Oct 1998.
3. Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002.
4. A.K. Figueiredo, M.A.F. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, March 2002.
5. Gérard Govaert. *Analyse des données*. Lavoisier, 2003.
6. P. Heas and M. Datcu. Modelling trajectory of dynamic clusters in image time-series for spatio-temporal reasoning. *IEEE Transactions on Geoscience and Remote Sensing*, 43(7):1635– 1647, nov 2005.
7. H. Maître I. Kyrgyzov and M. Campedel. Combining clustering results for the analysis of textures of spot5 images. In *ESA-EUSC: Image Information Mining*, 2005.
8. A. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, NJ, 1988.
9. H. Maître M. Campedel, E. Moulines and M. Datcu. Feature selection for satellite image indexing. In *ESA-EUSC: Image Information Mining*, 2005.
10. David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
11. J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.
12. J. Rissanen. Universal coding, information, prediction, and estimation. *IEEE Trans. Inform. Theory*, 30(4):629–636, 1984.
13. Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
14. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.