ParKerC: Toolbox for Parallel Kernel Clustering Methods

Sandrine Mouysset[1] and Ronan Guivarch[2]

[1] University of Toulouse, IRIT-UPS, France
[2] University of Toulouse, INP(ENSEEIHT)-IRIT, France

Abstract. A large variety of fields such as biology, information retrieval, image segmentation needs unsupervised methods able to gather data without a priori information on shapes or locality. By investigating a parallel strategy based on overlapping domain decomposition, we propose to adapt two kernel clustering methods respectively based on spectral and density-based properties in order to treat large data sets in fields of pattern recognition.

1 Introduction

Many fields from Social Science to Medicine and Biology generate large amount of data to analyze. Clustering aims at partitioning data sets in clusters in order to group data points with a similarity measure. Kernel clustering methods have become an increasingly popular tool for machine learning [1]. These methods rely on the use of positive definite kernel functions which enable them to operate in a high-dimensional feature space and provide, in particular, interesting spectral properties [2]. In this paper, we investigate a parallel strategy based on domain decomposition to treat large data sets for spectral clustering and mean shift [6], two widely used kernel clustering methods.

2 ParKerC Toolbox

In this section, we first present the principle of the proposed toolbox. Then we introduce the kernel clustering methods, their inherent parameters and the suitability with a domain decomposition strategy.

2.1 Principle

Let consider a data set $S = \{x_i\}_{i=1..n} \in \mathbb{R}^p$. The principle of the parallel toolbox is based on domain decomposition with overlaps. By dividing the data set S in q sub-domains, each processor applies independently the clustering algorithm on the subsets and provides a local partition. For each subdomain and each kernel method, the number of clusters is automatically determined. This heuristic avoids us to fix the targeted number of clusters. The final number of clusters k will be provided after the grouping step.

The grouping step is dedicated to link the local partitions from the subdomains thanks to the overlap and the following transitive relation: $\forall x_{i_1}, x_{i_2}, x_{i_3} \in S$,

$$\text{if } x_{i_1}, x_{i_2} \in C^1 \text{ and } x_{i_2}, x_{i_3} \in C^2 \text{ then } C^1 \cup C^2 = P \text{ and } x_{i_1}, x_{i_2}, x_{i_3} \in P \quad (1)$$

where C^1 and C^2 two distinct clusters and P a larger cluster which includes both C^1 and C^2. By applying this transitive relation (1) on the overlap, the connection between subsets of data is established and provides a global partition.

We can implement this algorithm using a Master-Slave paradigm as summarized in algorithms 1 and 2.

Algorithm 1 Parallel Algorithm: Master

1: Pre-processing step
  1.1 Read the global data and the parameters
  1.2 Split the data into q subsets
  1.3 Compute the affinity parameter δ with the formula given in paper 2; the bandwidth of the overlapping is fixed to $c \times \delta$ with $c \in \mathbb{N}$.
2: Send δ and the data subsets to the slaves
3: Perform the Clustering Algorithm on its subset
4: Receive the local partitions and the number of clusters from each slave
5: Grouping Step
  5.1 Gather the local partitions in a global partition with the transitive relation (1)
  5.2 Output a partition of the whole data set S and the final number of clusters k

Algorithm 2 Parallel Algorithm: Slave

1: Receive δ and its data subset from the Master
2: Perform the Clustering Algorithm on its subset
3: Send its local partition and its number of clusters to the Master

Overlapping bandwidth The idea is to consider an uniform distribution where data points are separated by the same distance each other. To define this distance, we consider both the dimension of the problem as well as the density of points in the given p-th dimensional data set. In fact, the data set S is included in a p-dimensional box bounded by ρ_d the largest distance between pairs of points in each dimension d of S: $\rho_d = \max_{1 \leq i,j \leq n} |x_{id} - x_{jd}|, \ \forall d \in \{1, .., p\}$.
So the uniform distance noted δ could be defined as follows:

$$\delta = \left(\frac{\prod_{i=1}^{p} \rho_i}{n} \right)^{\frac{1}{p}}. \quad (2)$$

From this distance, the ovelapping bandwidth is set as a multiple of δ.

In the following, we present two kernel clustering methods, the spectral clustering and the mean shift, and how to tune inherent parameters of both methods. The first method based on eigen-decomposition of kernel affinity matrix is used in pattern recognition or image segmentation to cluster non-convex domains without a priori on the shapes. The second method relies on a non-parametric estimator of density gradient for locating the maxima of the density function called mode.

2.2 Spectral clustering

Spectral clustering uses eigenvectors of a matrix, called Gaussian affinity matrix, in order to define a low-dimensional space in which data points can be clustered.

Algorithm Assume that the number k of targeted clusters is known (we will see how to automatically determine it). First, the spectral clustering consists in constructing the affinity matrix based on the Gaussian affinity measure between points of the dataset S. After a normalization step, the k largest eigenvectors are extracted. So every data point x_i is plotted in a spectral embedding space of $\mathbb{R}^k$ and the clustering is made in this space by applying $K-means$ method. Finally, thanks to an equivalence relation, the final partition of data set is defined from the clustering in the embedded space. Algorithm 3 presents the different steps of spectral clustering.

Algorithm 3 Spectral Clustering Algorithm

Input: data set S, number of clusters k

1. Form the affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by:

$$A_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{(\sigma/2)^2}\right) & \text{if } i \neq j, \\ 0 \text{ otherwise,} \end{cases} \tag{3}$$

2. Construct the normalized matrix: $L = D^{-1/2} A D^{-1/2}$ with $D_{i,i} = \sum_{j=1}^{n} A_{ij}$,
3. Assemble the matrix $X = [X_1 X_2 .. X_k] \in \mathbb{R}^{n \times k}$ by stacking the eigenvectors associated with the k largest eigenvalues of L,
4. Form the matrix Y by normalizing each row in the $n \times k$ matrix X,
5. Treat each row of Y as a point in $\mathbb{R}^k$, and group them in k clusters via the *K-means* method,
6. Assign the original point x_i to cluster j when row i of matrix Y belongs to cluster j.

Justification From the definitions of both the Gaussian affinity A_{ij} between two data points x_i and x_j and the Heat kernel $K_t(x) = (4\pi t)^{-\frac{p}{2}} \exp\left(-\|x\|^2/4t\right)$ in free space $\mathbb{R}_+^* \times \mathbb{R}^p$, we can interpret the gaussian affinity matrix as discretization of heat kernel by the following equation:

$$A_{ij} = (2\pi\sigma^2)^{\frac{p}{2}} K_t\left(\sigma^2/2, x_i - x_j\right). \tag{4}$$

So, we can prove that eigenfunctions for bounded and free space Heat equation are asymptotically close [3,5]. With Finite Elements theory, we can also prove that the difference between eigenvectors of A and discretized eigenfunctions of K_t is of an order of the distance between points include inside the same cluster. This means that applying spectral clustering into subdomains resumes in restricting the support of these L^2 eigenfunctions which have a geometrical property: their supports are included in only one connected component.

Thus, the overlapping domain decomposition does not alter the global partition because the eigenvectors carry the geometrical property and so, the clustering property.

Tuning parameters Spectral clustering depends on two parameters: the Gaussian affinity parameter σ and the number of clusters k. The *Gaussian affinity matrix* (3) is widely used and depends on a free parameter σ. It is known that this parameter affects the results in spectral clustering and spectral embedding. From the definition of δ defined by (2) in which we consider the case of an uniform distribution in the sense that all pair of points are separated by the same distance δ in the box of edge size D_{max}, we can state that clusters may exist if there are points that are at a distance no more than a fraction of δ.
The *number of clusters* k is defined from a measure based on the ratio of the Frobenius norms of the affinity measure between distincts clusters and within clusters [4]. The value of k that minimizes this ratio becomes the number of clusters.

2.3 Mean shift

Introduced by Fukunaga and Hostetler [6], mean shift method considers the points in the feature space as a probability density function. Dense regions in feature space corresponds to local maxima (or mode). The clusters are then associated with the modes.

Algorithm Mean shift associates each data point in $\mathbb{R}^p$ with the nearby peak of the dataset's probability density function. For each data point, mean shift defines a window around it and computes the mean of the data points which belong to this window. Then it shifts the center of the window to the mean and repeats the algorithm till it converges. In other words, the window shifts to a more denser region of the dataset.

Algorithm 4 Mean shift Algorithm

Input: data set S, bandwidth h
For each data point $x_i \in S$,

1. Compute mean shift vector $m(x_i)^t$
2. Move the density estimation window to $m(x_i)^t$
3. repeat till convergence i.e $\|m(x_i)^{(t+1)} - m(x_i)^t\| \leq threshold$

Justification Mean shift relies on kernel density estimation. Kernel density estimation [8] (also called the Parzen window technique [9]) is the most popular non parametric density estimation method. Given a kernel K, a bandwith parameter h, kernel density estimator for a given set of n p-dimensional points is:

$$f(x) = \frac{1}{nh^p} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{5}$$

Mean shift is based on Gradient ascent on the density contour [7]. So, for each data point, we perform gradient ascent on the local estimated density until convergence. So:

$$\nabla f(x) = \frac{1}{nh^p} \sum_{i=1}^{n} K'\left(\frac{x - x_i}{h}\right) \tag{6}$$

where $K'(x)$ is the derivative of $K(x)$. The stationary points obtained via gradient ascent represent the modes of the density function. All points associated with the same stationary point belong to the same cluster. By assuming that $g(x) = -K'(x)$, the following quantity $m(x)$, called mean shift, is computed as follows:

$$m(x) = \frac{\sum_{i=1}^{n} g\left(\frac{x-x_i}{h}\right) x_i}{\sum_{i=1}^{n} g\left(\frac{x-x_i}{h}\right)} - x \tag{7}$$

With this strategy of searching the maximum of local density, this method does not require to fix the number of clusters.

This implies that mean shift can be runned in subsets of S and if a cluster relies on several subdomains then the transitive relation (1) will merge the subclusters.

Tuning parameters As said in the previous section, the number of clusters is automatically defined. But mean shift is sensitive to the selection of bandwidth h. A small h can slow down the convergence whereas a larger one can speed up the convergence and merge two modes. We can define it automatically by considering the bandwidth h as a multiple of the uniform distance δ defined by (2).

3 Implementation and results

The code of the ParKerC toolbox is written in FORTRAN 90 using MPI library to handle the communication between processors.

3.1 Application to clustering datasets

We present in this Fig.1 some results on four different datasets from a clustering benchmark to validate our parallel approach:

	unbalance	spiral	Compound3	jain
Size	6500	312	219	373
Nb Clusters	8	3	3	2

Each problem is solved by using spectral clustering or mean shift methods, in sequential (1×1) and in parallel (2×2 square-partitioning). We tuned the parameters as explained in the theoretical sections: δ is computed; we used it to control the spectral clustering, to define the bandwidth for the mean shift ($C \times \delta$) and the overlapping bandwidth ($2 \times \delta$) for domain partitioning.

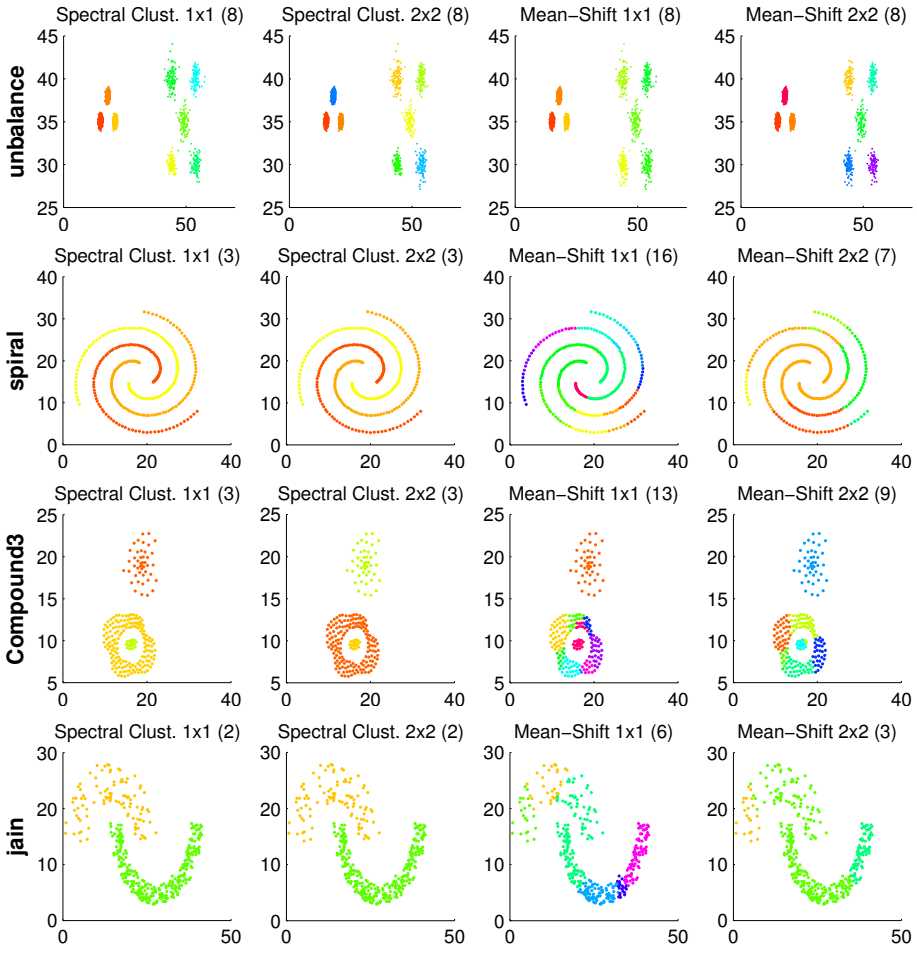

Fig. 1. Examples of dataset segmentation

We can notice that spectral clustering gives expected results for all problems both in sequential and in parallel (the number of exhibited clusters are between parenthesis). In contrary, mean shift have good results when the clusters are convex but poor ones with non-convex clusters; we tried different values of C

and present the less worse results. With no post-treatment (for instance, merging using transitivity), its results are not exploitable.

However for the examples where mean shift has a good behavior, it is much faster than spectral clustering.

3.2 Application to image segmentation

For image segmentation, the domain decomposition is applied geometrically on the image and can be also applied on the brightness distribution (or color levels). Thus, the kernel function is applied on geometrical and brightness/color data. The kernel function K at a pixel x will be decomposed according to the spatial and color vectors as:

$$K_{h_r,h_s}(x) = K\left(\frac{x^s}{h_s}\right) K\left(\frac{x^r}{h_r}\right) \tag{8}$$

where $x^s \in \mathbb{R}^2$ is the spatial vector of the pixel x and $x^r \in \mathbb{R}^3$ is the 3D color level vector of x and h_s and h_r are respectively the spatial and color parameters. We apply in parallel both spectral clustering and mean shift on a geometrical picture as shown in Fig.2 and Fig.3 (painting from Swiss artist, Sophie Taeuber-Arp).

(a) original image (b) mean shift (5×4) {14 clusters} (c) spectral clust. (5×4) {91 clusters}

Fig. 2. Results of parallel clustering methods on an image (275×194)

4 Conclusion and perspectives

We have validated the two parallel methods, both with small datasets and an image. We will continue our tests on larger problems to fully validate our FORTRAN code. These kernel methods offer different clustering analysis. Spectral clustering, based on connected components, can partition clusters with uniform distribution and circular shapes. Mean shift, relied on a density-based approach, can separate clusters even when they are connected by few points.

(a) original painting (b) mean shift (5 × 4) {1605 clusters} (c) spectral clust. (5 × 4) {1400 clusters}

Fig. 3. Results of parallel clustering methods on an painting (500 × 200)

Our spectral clustering method on a sub-domain uses sequential version of algebra linear kernels (LAPACK or ARPACK). Chen et al [10] proposed a parallel version of the spectral clustering method that can be used on one subdomain. In the same way, we can modify the mean shift on each subdomain by the fully parallel mean shift of Varga et al [11]. Thus, for both spectral clustering and mean shift, we can imagine a two-stage parallel implementation: sub-domain decomposition and then a parallel solution on each sub-domain.

References

1. Hofmann, T. and Schölkopf, B. and Smola, A. J, Kernel methods in machine learning, *The annals of statistics*, 1171–1220, 2008.
2. Ng, A. Y. and Jordan, M. I. and Weiss, Y., On spectral clustering: analysis and an algorithm, *Proc.Adv.Neural Info.Processing Systems*, 2002.
3. Mouysset, S. and Noailles, J. and Ruiz, D., On an interpretation of Spectral Clustering via Heat equation and Finite Elements theory, *International Conference on Data Mining and Knowledge Engineering*, 2010.
4. Mouysset, S. and Noailles, J., Ruiz, D. and Guivarch, R., On a strategy for Spectral Clustering with parallel computation, *High Performance Computing for Computational Science: 9th International Conference*, 2010.
5. Mouysset, S. and Noailles, J. and Ruiz, D. and Tauber, C., Spectral Clustering: interpretation and Gaussian parameter, *Data Analysis, Machine Learning and Knowledge Discovery*, 153–162, 2014.
6. Fukunaga K. and Hostetler L.D., The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition, *IEEE Transactions on Information Theory*, vol 21, 32–40, 1975.
7. Comaniciu, D. and Meer, P., Mean shift analysis and applications, *Proceedings of the 7th IEEE International Conference on Computer Vision*, 1197–1203, 1999.
8. Rosenblatt, M., Remarks on some nonparametric estimates of a density function, *The Annals of Mathematical Statistics*, 27, 3, 832–837, 1956.
9. Parzen, E., On estimation of a probability density function and mode, *The annals of mathematical statistics*, 1065–1076, 1962.
10. Chen, W-Y. and Yangqiu, S. and Bai H. and Lin C-J. and Chang E. Y., Parallel Spectral Clustering in Distributed Systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.
11. Varga, B. and Karacs, K., High-resolution image segmentation using fully parallel mean shift, *EURASIP Journal on Advances in Signal Processing*, 1–17, 2011.