Les Cahiers du GERAD ISSN: 0711–2440

Statistical learning with the determinantal point process

S. Vicente,
A. Murua

G–2020–23–EIW05

April 2020

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2020
– Bibliothèque et Archives Canada, 2020

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2020
– Library and Archives Canada, 2020

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Statistical learning with the determinantal point process

Serge Vicente
Alejandro Murua

Département de mathématiques et de statistique,
Université de Montréal, Montréal (Québec),
Canada, H3T 1J4

vicentes@dms.umontreal.ca
murua@dms.umontreal.ca

April 2020
Les Cahiers du GERAD
G–2020–23–EIW05

Abstract: *The determinantal point process (DPP) provides a promising and attractive alternative to simple random sampling in cluster analysis or classification, for the initial random selection of points required by most algorithms. As a probabilistic model of repulsion, the DPP elects which points are similar and have less probability to appear together, favouring then more diverse subsets of points. After a short introduction to DPP, we show how its use for choosing initial subsets of points in a clustering algorithm run multiple times on large datasets can improve the quality of final results.*

1 Introduction

A classical core procedure in fields such as biology, psychology, medicine, marketing, computer vision and remote sensing is to group elements based on similar features (cluster analysis) [13], to provide a framework for learning. Some clustering techniques, such as the standard k-means algorithm or the partitioning around medoids (PAM) algorithm, are characterized by an initial choice of a subset of random points. We find the same type of initial choice in some classification techniques, such as neural networks or machine learning. However, selecting a simple random subset of points does not take into account the diversity among the selected points. As this type of sampling gives to every point an equal probability of being selected, a subset of points may include many similar points that carry the same type of information and representability. In some domains of research, where the diversity of elements is a major concern and ensures a better coverage of all its facets, single random sampling can lack some of them. The determinantal point process settles which points are similar and therefore have less probability to appear together, in contrast to simple random sampling. It intents to capture negative correlations between n points and has been used in machine learning as a model for subset selection [9]. Kulesza and Taskar [15] emphasize that the negative correlations are measured by a $n \times n$ matrix whose entries represent a measure of similarity between each pair of points. Similar elements have less probability to be co-selected, resulting in subsets that are more *diverse*. Clustering techniques in particular seek to obtain a unique optimal partition of data, by maximizing both intra-cluster similarity and inter-cluster dissimilarity. However, as stressed by [29], if different partitional techniques are applied to the same data, they can produce very different clustering results, due to the lack of an external objective and impartial criterion. The techniques' dependency on the initial choice of points can also explain those differences. To improve the quality and robustness of clustering results, [28] proposed the *cluster ensembles* framework, which main objective is to combine different clustering results into a single consolidated clustering. Monti et al. [21] introduced a cluster ensemble method in genomic studies and gene expression: *the consensus clustering*. Based on resampling and bootstrapping techniques, it seeks to attain a single consolidated clustering configuration from multiple runs of the same clustering algorithm. For sampling the initial points of the algorithm, we used the determinantal point process presented by [10, 15]. The paper is organized as follows: we present the determinantal point process in Section 2; we explain our consensus clustering algorithm in Section 3; we study the case of large datasets in Section 4; we present the quality measure for results in Section 5; we refer algorithms taken as reference in Section 6; we show results on simulated and real data in Section 7.

2 The determinantal point process (DPP)

Origins of DPP date back to [19] in quantum physics, known then as "fermion process", intended to model distributions of fermion systems at thermal equilibrium. The name "Determinantal Point Process" is established, introduced and made accepted as standard in mathematics' community by [3]. It also arises in studies of nonintersecting random paths, random spanning trees, and eigenvalues of random matrices [8, 4, 10].

Starting with a global overview of DPP, let $\mathcal{S} = \{\boldsymbol{x_1}, \dots, \boldsymbol{x_n}\}$ be a discrete set of n elements, with $2 \leq n < \infty$ and where $\boldsymbol{x_i}$ represents a $p-$dimensional vector, i.e. $\boldsymbol{x_i} \in \mathbb{R}^p$, $i = 1, \dots, n$. A point process on $\mathcal{S}$ is a probability measure on $2^{\mathcal{S}}$, the set of all subsets of $\mathcal{S}$. It is called a DPP if, for a particular random subset $Y \in 2^{\mathcal{S}}$, its probability mass function is given by

$$P\left(\boldsymbol{Y} = Y\right) = \frac{\det(L_Y)}{\det(L + I_n)}, \tag{1}$$

where $\boldsymbol{Y}$ is a random variable representing the subset selected from $2^{\mathcal{S}}$, L is a $n \times n$ real, symmetric and positive semidefinite matrix measuring similarity between pair-wised elements of $\mathcal{S}$, L_Y is the submatrix of L with rows and columns indexed by Y, i.e., $L_Y = \left[L_{ij}\right]_{i,j \in Y}$ and I_n is the $n \times n$ identity matrix.

Determinants have a well-known geometric interpretation. Let B be a $m \times n$ matrix such that $L = B^T B$. B can always be found for $m < n$ due to positive semidefiniteness of L. Denoting the columns of B by B_i, for $i = 1, \dots, n$, we have

$$P\left(\boldsymbol{Y} = Y\right) \propto \det\left(L_Y\right) = \mathrm{Vol}^2\left(\{B_i\}_{i \in Y}\right), \tag{2}$$

where Vol^2 represents the squared volume of the parallelepiped spanned by the columns of B corresponding to elements in Y. The columns of B can be interpreted as feature vectors describing the elements of $\mathcal{S}$ and, therefore, L measures similarity using dot products between feature vectors. By Equation (2), we can see the probability assigned by a DPP to a subset Y is related to the volume spanned by its associated feature vectors: diverse sets have then a higher probability, because their feature vectors are more orthogonal and hence span larger volumes.

3 Consensus clustering algorithm

Consider again the set $\mathcal{S}$ of n elements, and a particular partitional clustering technique run M times over the set $\mathcal{S}$. The agreement among the several runs of the algorithm is based on the consensus matrix C, a $n \times n$ symmetric matrix where the entry C_{ij}, $i, j = 1, \dots, n$ represents the proportion of runs in which two elements $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ of $\mathcal{S}$ belong to the same cluster, i.e.

$$C_{ij} = \frac{\sum_{m=1}^{M} c_{ij}^m}{M}, \tag{3}$$

where c_{ij}^m is an indicator of wether element $\boldsymbol{x_i}$ belongs to the same cluster as $\boldsymbol{x_j}$ in the m-th run. The consensus clustering method was meant to attain a single consolidated clustering from multiple runs of the same clustering algorithm. Any partitional clustering method can be chosen, then. However, rather than using a well-known clustering method like $k-$means or PAM, we constructed our clustering algorithm to obtain a consolidated consensus clustering configuration. At each run, we start the algorithm with a Voronoi diagram on the set $\mathcal{S}$, which partitions the space into several cells or regions, based on a subset of points that are called generator points. These points will be sampled among the elements of $\mathcal{S}$ using the DPP defined by (1) and the sampling algorithm developed by [10, 15]. For the construction of the similarity matrix L, we will use kernel-based methods, which have been widely used in recent research into pattern analysis, like classification, regression and clustering [11]. As kernels are often considered measures of similarity, a higher kernel value represents a higher correlation in a high-dimensional (possibly infinite) Hilbert space. A popular kernel choice is the Gaussian kernel, which we will use to obtain the entries of L:

$$L = \left[\exp\left(-\frac{\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2}{2\sigma^2}\right)\right]_{i,j=1}^{n}, \tag{4}$$

where the scale parameter σ represents the relative spread of the distances $\|\boldsymbol{x_i} - \boldsymbol{x_j}\|$, the Euclidean distance between $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$, a common choice for the Gaussian kernel.

Finally, after M runs, we will obtain M Voronoi diagrams, from which we can compute the consensus matrix C with entries defined by (3). Following [2], if $C_{ij} \geq \theta$, with $0 \leq \theta \leq 1$, points $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are defined as "friends" and then included in the same final cluster. As θ is unknown, there are several choices of final clusters, depending on the value of θ. Motivated then by [23] and [22], we used the *least-squares clustering* (LSCLUST) procedure of [7] to choose the optimal final cluster among the several choices: supposing we have B clusters, for each cluster $\boldsymbol{c}$ in $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_B$, a $n \times n$ matrix $\delta(\boldsymbol{c})$ can be built. The (i, j) element of the matrix, $\delta_{i,j}(\boldsymbol{c})$, is an indicator of wether element i of $\mathcal{S}$ belongs to the same cluster than j. Element-wise averaging of these association matrices yields a pairwise probability matrix of clustering, denoted $\widehat{\boldsymbol{\pi}}$. The least-squares clustering $\boldsymbol{c}_{LS}$ is the observed clustering $\boldsymbol{c}$ that solves the following minimization problem:

$$\boldsymbol{c}_{LS} = \underset{\boldsymbol{c} \in \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_B\}}{\arg\min} \sum_{i=1}^{n} \sum_{j=1}^{n} \left[\delta_{i,j}(\boldsymbol{c}) - \widehat{\boldsymbol{\pi}}_{i,j}\right]^2 .$$

4 Case of large datasets

The eigendecomposition of the matrix L of DPP defined by (1) is a central step for obtaining the generator points through the sampling algorithm of [10, 15]. It is well known that the computational complexity of eigendecomposition of a $n \times n$ symmetric matrix is $O\left(n^3\right)$ and, as n grows larger, the computation of the characteristic polynomial itself becomes expensive due to the computational complexity of calculating determinants. Therefore, computing only the largest eigenvalues can substantially reduce the computational burden of obtaining all the eigenvalues. The literature points out many references of well-known algorithms that can extract the t largest (or smallest) eigenvalues, with their associated eigenvectors, of a $n \times n$ Hermitian matrix, where usually, we have $t \ll n$. One of the most classical and used algorithms is the Lanczos algorithm [17] and its variations, such as the implicitly restarted Lanczos method, proposed by [5], which we will use if the dimension of L is very large. The Lanczos algorithm and its implicitly restarted variation were specially developed for large sparse symmetric matrices. Consequently, when L is large, to implement the implicitly restarted Lanczos method, it is necessary to find a good approximation of the dense matrix L by a sparse matrix. Nevertheless, the large size of L can still be a computational burden for the implementation of the algorithm, which motivated us to consider a special approach for dealing with large matrices, inspired by dimension reduction techniques.

Let then L be a large kernel matrix, of size $n \times n$, defined by (4) and let $L_1, L_2, \ldots, L_R$ denote a set of R submatrices of size $r \times r$ each, taken randomly from L, where $r < n$ (ideally, $r \ll n$). We apply the following methodology to the set of submatrices:

1. select randomly an index i_1 from $\{1, 2, \ldots, R\}$ and consider the submatrix L_{i_1};
2. find a sparse approximation of the submatrix L_{i_1} considering the k-nearest neighbours of each point of the submatrix, according to the k-nearest neighbours graph introduced by [25];
3. generate a sample Y_{i_1} from L_{i_1} through DPP, using the usual sampling algorithm of [10, 15] and the Lanczos algorithm for extracting the t largest eigenvalues;
4. build a Voronoi diagram for the n points, using the generated sample Y_{i_1};
5. repeat steps 1 to 4 for indexes $i_2, i_3, \ldots, i_N$, using always $\{1, 2, \ldots, R\}$;
6. apply the consensus clustering summarized in Section 3 to the set of N partitions obtained.

The number R of submatrices to be sampled must be chosen so that we get benefits from using the submatrices to sample the generator sets through DPP rather using the whole kernel matrix L. We know that the computational complexity of eigendecomposition of the $n \times n$ kernel matrix L is $O\left(n^3\right)$, employing n^3 operations. But, if we have R submatrices of size $r \times r$ each, the eigendecompositions of these submatrices employ Rr^3 operations. To obtain benefits from the sampled submatrices, we must guarantee that

$$\begin{aligned} Rr^3 &< < n^3 \Leftrightarrow \\ \Leftrightarrow R &< \left(\frac{n}{r}\right)^3 \Leftrightarrow \\ \Leftrightarrow R &< \left(\frac{1}{\gamma}\right)^3, \end{aligned}$$

where $\gamma = \frac{r}{n}$ represents the proportions of points considered for the submatrices. As we want to take advantage of dimension reduction and speed, we decided to choose R so that $R \ll \left(\frac{1}{\gamma}\right)^3$, and then, we decided to fix $R = \left\lfloor \frac{\left(\frac{1}{\gamma}\right)^3}{2} \right\rfloor$, where $\lfloor x \rfloor$ represents the floor function.

5 Clustering quality measure

As mentioned by [24], it is a common practice in the clustering literature to measure the goodness-of-fit of the optimal final cluster. Among the many known measures of goodness-of-fit that can be found in the literature, we will use the Adjusted Rand Index (ARI), first introduced by [26] and later adjusted for randomness by [12]. The ARI is a measure of agreement between two clustering configurations. The original Rand Index counts the proportion of elements that are either in the same clusters in both clustering configurations or in different clusters in both configurations. The adjusted version of the Rand Index corrected the calculus of the proportion, so its expected value is zero when the clustering configurations are random. The larger the ARI, the more similar the two configurations are, with the maximum ARI score of 1.0 indicating a perfect match.

6 Reference algorithms for comparison

To validate the performance of the consensus clustering summarized in Section 3 using DPP for choosing an initial set of points, we decided to compare the final results to two traditional clustering algorithms: PAM and k-means algorithms.

The PAM algorithm is a classical partitioning technique of clustering proposed by [14], which chooses data points for centers by simple random sampling. As DPP selects also data points for centers but based on diversity, the goal of comparing it with PAM method is to evaluate how the quality results of clustering behave if we consider diversity as a sampling criterion. The k-means algorithm was proposed by Stuart Lloyd in 1957, and later published in [18]. It starts with an initial set of k means, representing k clusters, assigning then each observation to the cluster with the nearest mean and proceeding with updating steps until convergence to a final optimal cluster configuration. However, as argued by [6], the popular methods for choosing the initial set of k means, such as Forgy, Random Partition and Maximin methods, result often in a final optimal cluster configuration with a low clustering quality. We decided then to use the k-means++ algorithm of [1], a popular choice that avoids the poor quality results of the traditional methods for choosing the initial means. Once more, our goal is to evaluate how the quality results of clustering with DPP behave if we consider diversity as a sampling criterion, when compared to the k-means algorithm that uses k-means++ for choosing initial points.

7 Results

The consensus clustering algorithm presented in Section 3 was applied to the case of datasets with a very large number of observations.

7.1 Simulated data

Using the algorithm of [20], we simulated a dataset constituted by $n = 10000$ vectors of dimension $p = 15$ grouped in 10 predefined clusters. As the dimension of the corresponding matrix L is very large, we decided to use the methodology described in Section 4 for the eigendecomposition required for the DPP sampling, adopting $N = 1000$. Prioritizing also a minimal computational time, we chose $\gamma = 0.1$ (and consequently $R = 500$), $k = 325$ nearest neighbours (which results in sparse matrices with approximately 60% of zeros) and the $t = 25$ largest eigenvalues of the sparse matrices. The consensus clustering algorithm of Section 3 was then applied, performing $M = 1000$ runs, and the quality of the optimal final cluster was assessed by the ARI described in Section 5. To ascertain the ARI variability, we decided to repeat the whole procedure 5 times and obtain one Boxplot for the ARI values. For comparisons, we also included the Boxplot resulting from the application of the clustering algorithm with DPP to the dense matrix L, from which we extracted all the eigenvalues, and the Boxplots resulting from k-means and PAM algorithms. All the comparing methods were also repeated 5 times for the construction of the Boxplots. Figure 1 presents the comparison of the four Boxplots.

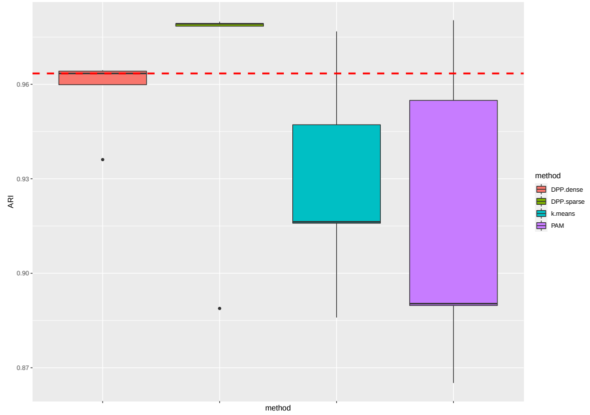

Figure 1: From left to right: Boxplots of the ARI for the DPP with dense matrix, DPP with sparse aproximations, k-means and PAM, with the dashed line indicating the median value obtained with the dense matrix

We also adopted another analysis to evaluate the quality of the sparse approximation of L: we obtain the kernel density estimation of the set of eigenvalues extracted from the sparse approximations of L and check graphically how the estimated density concentrates around all true eigenvalues of the dense kernel matrix L. We choose a Gaussian kernel for the density estimation and [27] rule for the bandwidth of the kernel. Figure 2 shows the estimated density and the values of the true eigenvalues.

We also took advantage of the opportunity to obtain the kernel density estimation of the set of the true eigenvalues extracted from the dense matrix L and measure its divergence from the kernel density estimation of the set of eigenvalues extracted from the sparse approximations of L depicted in Figure 2. The divergence will be measured through the Kullback-Leibler (KL) divergence, introduced by [16]. As the KL divergence does not obey to the symmetry property of a metric, for each pair of compared estimated densities, we will compute the KL divergence in both directions and compute the average of the two divergences. We can find the result in Table 1.

Table 1: KL divergence between the kernel density estimation of the eigenvalues extracted from the sparse approximations of L and the kernel density estimation of the true eigenvalues of L.

KL divergence
0.00005336

Additionally, we will also report and compare the elapsed time in seconds for eigenvalues computation using a sparse approximation of L or the original dense matrix L. The results are shown in Table 2.

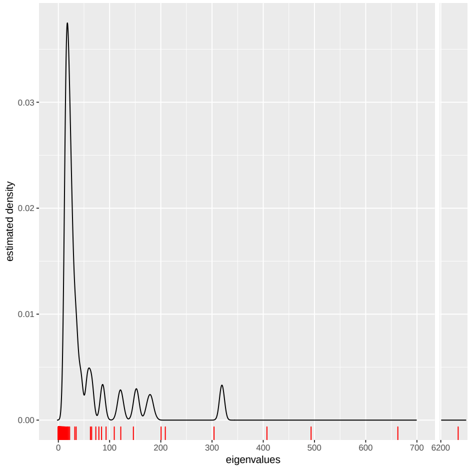

Figure 2: Kernel estimated density of the set of eigenvalues extracted from the sparse approximations of the dense matrix L, with the true eigenvalues marked on the abscissa axis

Table 2: Comparison of elapsed times (in seconds) for eigenvalues calculation.

	Elapsed time
Sparse L	0.079
Dense L	0.216

Finally, to explain the differences between DPP and PAM, we also present in Figure 3 the histograms of the logarithm of the probability mass function given by (1) for $M = 1000$ random subsets, using a DPP sampling or the simple random sampling.

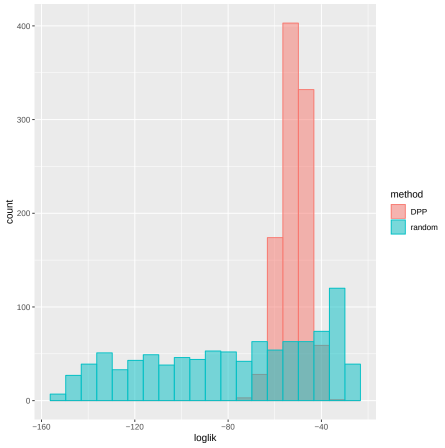

Figure 3: Histograms for the logarithm of the probability mass function (loglik) of $M = 1000$ random subsets using DPP and random sampling.

7.2 Real data

In this subsection, we considered two real datasets:

1. A dataset about human activity recognition and postural transitions using smartphones, collected from 30 subjects who performed six basic postures (downstairs, upstairs, walking, jogging, sitting and standing), including also six transitional postures between static postures (stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie and lie-to-stand), in the same environment and conditions, while carrying a waist-mounted smartphone with embedded inertial sensors. The

dataset consists of 10929 observations, with 561 time and frequency extracted features, which are commonly used in the field of human activity recognition. The dataset has then $n = 10929$ observations, $p = 561$ variables and $K = 12$ classes. The dataset is available on the *UCI Machine Learning Repository*, a well known database in the Machine Learning community for clustering and classification problems.

2. The Modified National Institute of Standards and Technology (MNIST) dataset, one of the most common datasets used for image classification. This dataset contains 60000 training images and 10000 testing images of handwritten digits, obtained from American Census Bureau employees and American high school students. Each observation represents a 28×28 pixel gray-scale image depicting a handwritten version of one of the ten possible digits (0 to 9). Pixels are organized row-wise, so that each row of the dataset represents an image, where the first number of each line is the label, i.e. the digit which is depicted in the image, and the remaining 784 numbers are the pixels of the 28×28 gray-scale image. The scale is available in two versions: the original scale between 0 (background or white) and 255 (foreground or black), or scaled between 0 and 1. For this section, we decided to use the testing set of 10000 images with the scaled pixels between 0 and 1. The dataset has then $n = 10000$ observations, $p = 784$ variables and $K = 10$ classes.

We applied the same strategy of Subsection 7.1 to each dataset, along with a comparison with k-means (with k-means++ for initial points) and PAM algorithms: we decided to sample a proportion $\gamma = 0.1$ of the points of the kernel matrix L (and consequently $R = 500$) and again obtain a sparse approximation of the sampled submatrices with 60% of sparsity, choosing the appropriate number k of nearest neighbours for each dataset. The Lanczos algorithm was then applied to extract the first $t = 25$ eigenvalues of the sparse approximated submatrices. The consensus clustering algorithm of Section 3 was then applied, performing $M = 1000$ runs, and the quality of the optimal final cluster was assessed by the ARI described in Section 5. To ascertain the ARI variability, we decided to repeat the whole procedure 5 times and obtain one Boxplot for the ARI values. For comparisons, we also included the Boxplots resulting from k-means and PAM algorithms. All the comparing methods were also repeated 5 times for the construction of the Boxplots. Figure 4 presents the results for the smartphones dataset and Figure 5 presents the results for the MNIST dataset.

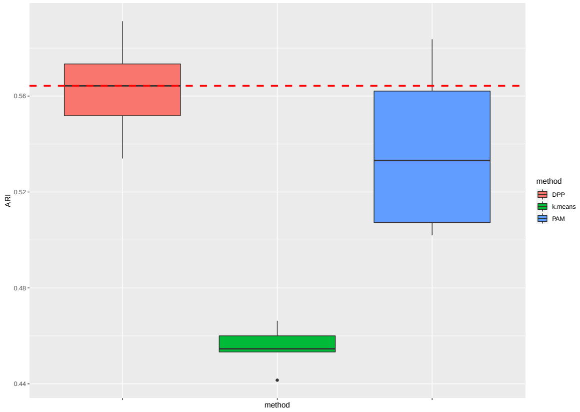

Figure 4: From left to right: Boxplots of the ARI for the DPP, k-means and PAM consensus clustering, with the dashed line indicating the median value obtained with DPP, for the smartphones dataset

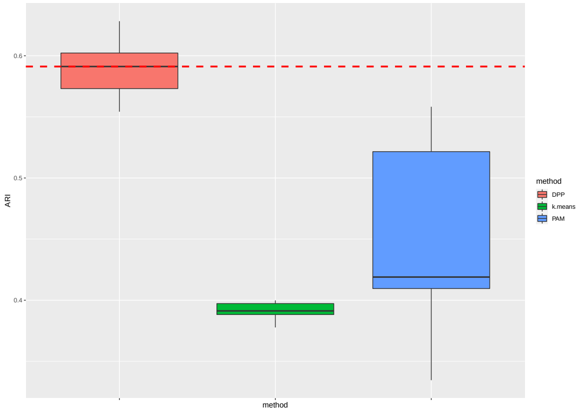

Figure 5: From left to right: Boxplots of the ARI for the DPP, k-means and PAM consensus clustering, with the dashed line indicating the median value obtained with DPP, for the MNIST dataset

8 Conclusion

The use of the sparse approximations is totally justified and has clear benefits, even with a low proportion γ of points sampled. The k-nearest neighbour graph approach provides then a good alternative to the use of the complete dense matrix L. Observing the results section, we present the following conclusions:

1. Simulated dataset: observing the Boxplots, we can see a lower quality of k-means and PAM results when compared to DPP. We also note that the approaches with DPP achieve a higher stability of the ARI, while the approaches with k-means and PAM give more heterogeneous results. Focusing on the kernel estimated density of the eigenvalues extracted from the sparse approximations of L, we can see a reasonably good concentration and fit around the true eigenvalues of L, supported by a low value of KL divergence. In terms of the elapsed time for eigenvalues extraction, we can see a clear time reduction, which becomes particularly important as the consensus clustering implies a repetition of the clustering algorithm a large number of times. Finally, the histograms of the logarithm of the probability mass function clearly show that DPP selects random subsets with higher and less dispersed probability mass values than simple random sampling, explaining a higher stability of the ARI.
2. Real datasets: observing the Boxplots, we can see a lower quality of k-means and PAM results when compared to DPP. We also note that the approaches with DPP achieve a higher stability of the ARI, while the approach with PAM give more heterogeneous results. Even if the k-means algorithm ensures a higher ARI stability when compared to DPP, it provides lower quality results.

The higher likelihood of the random subsets sampled by DPP confirms the higher diversity of those subsets, while the subsets sampled by random sampling can be highly or poorly diverse, with a very high dispersion in terms of diversity. DPP tends then to select points that maintain a high level of diversity at each sampling, proving then to be more consistent and stable than simple random sampling in terms of ensuring the heterogeneity of elements forming the subset. Moreover, taking into account the diversity of elements with DPP as a sampling method rather than simple random sampling, does not harm the quality of results, since the level attained by simple random sampling is more or less maintained, or even improved.

References

[1] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics.

[2] Marcelo Blatt, Shai Wiseman, and Eytan Domany. Superparamagnetic clustering of data. Phys. Rev. Lett., 76:3251–3254, Apr 1996.

[3] A. Borodin and G. Olshanski. Distributions on Partitions, Point Processes, and the Hypergeometric Kernel. Communications in Mathematical Physics, 211:335–358, 2000.

[4] Alexei Borodin and Alexander Soshnikov. Janossy densities i. determinantal ensembles. Journal of Statistical Physics, 113, 01 2003.

[5] Daniela Calvetti, L Reichel, and And D C Sorensen. An implicitly restarted lanczos method for large symmetric eigenvalue problems. Electronic Trans. Numer. Anal., 2:1–21, 04 1994.

[6] M. Emre Celebi, Hassan A. Kingravi, and Patricio A. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications, 40(1):200–210, 2013.

[7] David B. Dahl. Model-Based Clustering for Expression Data via a Dirichlet Process Mixture Model, pages 201–218. Cambridge University Press, 2006.

[8] D. J. Daley and D. Vere-Jones. An introduction to the theory of point processes. Vol. I. Probability and its Applications (New York). Springer-Verlag, New York, second edition, 2003. Elementary theory and methods.

[9] R. Hafiz Affandi, E. B. Fox, and B. Taskar. Approximate Inference in Continuous Determinantal Point Processes. ArXiv e-prints, November 2013.

[10] J. Ben Hough, Manjunath Krishnapur, Yuval Peres, and Bálint Virág. Determinantal processes and independence. Probability Surveys, 3(0):206–229, 2006.

[11] Tom Howley and Michael G. Madden. An evolutionary approach to automatic kernel construction. In Stefanos Kollias, Andreas Stafylopatis, Włodzisław Duch, and Erkki Oja, editors, Artificial Neural Networks – ICANN 2006, pages 417–426, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[12] Lawrence Hubert and Phipps Arabie. Comparing partitions. Journal of Classification, 2(1):193–218, Dec 1985.

[13] Anil Jain and Richard Dubes. Algorithms for Clustering Data. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[14] Leonard Kaufmann and Peter Rousseeuw. Clustering by means of medoids. Data Analysis based on the L1-Norm and Related Methods, pages 405–416, 01 1987.

[15] A. Kulesza and B. Taskar. Determinantal point processes for machine learning. ArXiv e-prints, July 2012.

[16] S. Kullback and R. A. Leibler. On information and sufficiency. Ann. Math. Statist., 22(1):79–86, 03 1951.

[17] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. Journal of Research of the National Bureau of Standards, 1950.

[18] Stuart P. Lloyd. Least squares quantization in pcm. IEEE Trans. Inf. Theory, 28:129–136, 1982.

[19] Odile Macchi. The Coincidence Approach to Stochastic Point Processes. Advances in Applied Probability, 7(1):83–122, 1975.

[20] Volodymyr Melnykov, Wei-Chen Chen, and Ranjan Maitra. Mixsim: An r package for simulating data to study performance of clustering algorithms. Journal of Statistical Software, Articles, 51(12):1–25, 2012.

[21] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning, 52(1):91–118, 2003.

[22] Johanna Muñoz and Alejandro Murua. Building cancer prognosis systems with survival function clusters. Statistical Analysis and Data Mining: The ASA Data Science Journal, 11(3):98–110, 2018.

[23] Alejandro Murua and Fernando A. Quintana. Semiparametric bayesian regression via potts model. Journal of Computational and Graphical Statistics, 26(2):265–274, 2017.

[24] Alejandro Murua and Nicolas Wicker. The Conditional-Potts Clustering Model. Journal of Computational and Graphical Statistics, 23(3):717–739, 2014.

[25] Rodrigo Paredes and Edgar Chávez. Using the k-nearest neighbor graph for proximity searching in metric spaces. In Proceedings of the 12th International Conference on String Processing and Information Retrieval, SPIRE'05, pages 127–138, Berlin, Heidelberg, 2005. Springer-Verlag.

[26] William M. Rand. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 66(336):846–850, 1971.

[27] B. W. Silverman. Density estimation for statistics and data analysis, volume 26 of Monographs on Statistics & Applied Probability. Chapman and Hall, 1986.

[28] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res., 3:583–617, 2002.

[29] Sandro Vega-Pons and José Ruiz-Shulcloper. A survey of clustering ensemble algorithms. International Journal of Pattern Recognition and Artificial Intelligence, 25(03):337–372, 2011.