UFR Mathématique et informatique - 2019-2020

Projet pluridisciplinaire 7

Réduction de la dimension et Classification régularisée

Réalisé par:
Boungnalith Vanna , Benhaddad Wissam, Messous Nazim , Sidi Yahya Mohamed Ali

Contents

List of Figures

List of Tables

1 Introduction

1.1 Models and objectives

In unsupervised learning, a lot of problems are solved through a common process divided in two steps: we first try to reduce the dimensionality of a dataset in order to have a better visualization of the latter and a better understanding of the correlations between individuals and variables. The dimensionality reduction can be achieved through a numerous number of methods and algorithms such as Principal Component Analysis, Linear Discriminant Analysis, Deep Autoencoders, t-SNE, etc. When the process is done, common clustering algorithms are applied in order to group individuals with common features together such as K-Means, Hierarchical clustering, Density-based Clustering, etc.
In this project, we will mainly focus on methods that achieve these two steps at the same time; i.e, a combination of dimensionality reduction and clustering. The main objectives is to compare the results of those methods to the more "classical" algorithms on the same tasks. In order to do that, we will evaluate the performance of clustering and separability through different scores resulting from minor experiments.

1.2 Dataset

Our work was done on six different datasets:

- **Coil20:** The dataset contains 1440 color images of 20 toys (72 images per object). The objects were placed on a motorized turntable against a black background. The turntable was rotated through 360 degrees to vary object pose. Images of the objects were taken at pose intervals of 5 degrees so it corresponds to 72 poses per object.
- **Coil100:** Same as Coil20 dataset but with 7200 images of 100 objects.
- **ORL:** The dataset contains 400 images of 40 individual's faces (10 images per individuals). The faces are in an upright position in frontal view, with a slight left-right rotation.
- **Yale:** The Yale Face Database contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink.
- **USPS:** The dataset contains 9298 images of handwritten digits, each images are the size of 16×16.
- **MNIST:** The dataset also contains images of handwritten digits but it has 70000 images which is each of a size of 32×32.

2 Methods and algorithms used

2.1 Algorithme 1

In this section we will introduce this new algorithm named Algorithm 1 by [Nad19], the main objective of this algorithm is to create a new representation of the data minimizing a loss function in order to

get a better clustering and a better visualization. The steps of the algorithm can be resumed like this : Initialization :

- First step is to initialize Q, for that purpose, we apply a PCA on the data in order to obtain the first q principal components (XQ), then the matrix Q is the eigenvectors obtained throughout the PCA and we compute XQ.
- Then we compute the laplacien matrix L, to do that we computed first the wight matrix (W) using euclidean distance and the degree matrix D, then we computed L=D-W, Finall we apply a SVD on L in order to obtain the data embedding B.
- After that using the XQ matrix, we use a simple K-means algorithm to compute the centers of the clusters, we obtained g (number of clusters) centers that we affect to a matrix S.
- Finally we create the matrix Z by affecting each elements i to the nearest center.

After completing the initialization we enter the loop and stop when $\epsilon < 0.01$, in each iteration we proceed to the following steps :

- Fisrt we compute S centers of the cluster:

$$S = (Z^T Z)^{-1} XQ \tag{1}$$

- then we compute the partition Z, as Z is a binary matrix, its calculation is deduced by a assignment of an (XQ)i element of the reduced space to the nearest center element of S.

$$z_{ik} = argmin_{k'} \|(XQ)_i - s_{k'}\|^2 - 2\beta(BQ_g)_{ik'} \tag{2}$$

- Now we compute Q, using the svd method, it should be noted that we used the identity matrix instead of the D matrix :

$$Q = UV^T \tag{3}$$

  we compute Qg :

$$Q = U_g V_g^T \tag{4}$$

2.1.1 Q parameter :

It represents the number of principal components we took after applying the PCA on the data, we took q=20 for all the dataset because Cumulative Variable's Explained Variance reached 70%-75% each time, and after that the gain made in cumulative variance is negligible (less than 0.1). We should add that taking q=15 or q=20 give worse results in term of NMI and ARI, and also separability.

2.1.2 Beta parameter :

It plays the role of a regulator, we should take a number beetween 0.1 and 1, otherwise if beta is greater than 1 the loss function give negative values, and less 0.1 the algorithm took too much iteration to converge. B=0.4 is the value we choosed, even if the algorithm worked good with other values in the previous range.

2.2 Clustrd

In this section we will briefly introduce a configurable tandem approach for dimensionality reduction and clustering. In [MDV19], the authors described the algorithms used as a mixture of Reduced Kmeans (RKM) and Factorial KMeans (FKM) in a sense where the objective function to optimize has been tweaked to take account of the two algorithms following the values of the α parameter. the possible values for this parameter are as follows :

- 0.5 The algorithm used is RKM. According to [DC94], the Reduced KMeans tackles the simultaneous dimension reduction and cluster analysis problem in such a way that the cluster allocation and dimension reduction maximizes the between variance of the reduced space. The objective function is defined as :

$$min_{RKM}(B, Z_K, G) = ||X - Z_K G B^T||^2 \tag{5}$$

  where [MDV19]:

  - X the centered and standardized $n \times Q$ data matrix.
  - B is a $Q \times d$ column-wise orthonormal loadings matrix.
  - Z_K is the $n \times K$ binary matrix indicating cluster memberships of the n observations into the K clusters.
  - G denotes the $K \times d$ cluster centroid matrix.

  After a quick transformation, it is shown in [DC94] that the optimization function can be turned into :

$$min_{RKM}(B, ZK) = ||X - PXBB^T||^2 \tag{6}$$

  where $P = Z_k(Z_k^T Z_k)^{-1} Z_k^T$.

  We find ourselves with a function independent from G that we aim to minimize.

  Also in [MDV19] we can see that using the projector matrix P and the trace operator. The following equation is derived :

$$||X - PXBB^T||^2 = Tr(X^T X) - Tr(B^T X^T PXB) \tag{7}$$

  By considering the maximization of $-min_{RKM}$, the right term of the above equation (the between cluster variance) is therefore maximized.

- 0 The algorithm used is Factorial K-means or FKM. In [VK01], the algorithm is described as a men of minimizing the within variance of the clusters in the reduced space. It can be described as simultaneous K-means clustering with PCA. The objective function is described as :

$$min_{FKM}(B, Z_K, G) = ||XB - Z_K G||^2 \tag{8}$$

  By getting rid of G like it was done for RKM in [DC94]. the equation becomes :

$$min_{FKM}(B, Z_K, G) = ||XB - PXB||^2 \tag{9}$$

- 1 Both RKM and FKM are used in tandem (principal component analysis followed by K-means of the factor scores). In [YH14], the authors proposed to reshape the objective function of RKM to look like follow :

$$||X - PXBB^T||^2 = ||X - XBB^T||^2 + ||XB - PXB||^2 \tag{10}$$

such as the the first term of the decomposition is a PCA criterion and the second is the FKM objective function. And by weighting the the two terms by the α coefficient, the resulting objevtive function becomes :

$$min_{tandem}(B, Z_K) = \alpha||X - XBB^T||^2 + (1 - \alpha)||XB - PXB||^2 \tag{11}$$

The solution to this optimization is well detailed in [MDV19]. It involves maximising the trace of a matrix by finding the eigenvectors/eigenvalues of a projection of X.

In [MDV19] the authors accentuated the difference between other values of the α parameter. They've also mentionned that the final model selection can be based on theoretical considerations, for instance by deciding a priori that the desired method is to be a compromise between FKM and RKM ($\alpha = 0.25$), or for instance, right in the middle of PCA and RKM ($\alpha = 0.75$). The most suitable solution for the hyperparameter calibration would be to do a gridsearch on the desired values and selecting the most interesting results in terms of a chosen metric.

2.3 Mixture of factor analysis (MFA)

Dimensionality reduction using MFA is performed through k factor models with Gaussian factors :

- In maximum likelihood factor analysis (FA), a p -dimensional real-valued data vector x is modeled using a k -dimensional vector of real-valued factors, z , where k is generally much smaller than p [Eve84]. The generative model is given by: $x = \Lambda z + u$
- The distribution of each observation is modelled,with probability π_j (j = 1, . . . , k), according to an ordinary factor analysis model $y = \eta_j + \Lambda_j + e_j$ with $e_j \sim \phi^{(p)}(0, \Psi_j)$, where Ψ_j is a diagonal matrix and $z_j \sim \phi^{(q)}(0, I_q)$
- In the observed space we obtain a finite mixture of multivariate Gaussians with heteroscedastic components:
  $f(y) = \sum_{j=1}^{k} \pi_j \phi^{(p)}(\eta_j, \Lambda_j \Lambda_j^T + \psi_j)$
- The generative model now obeys the following mixture distribution[Zou96] :

$$P(x) = \sum_{j=1}^{m} \int P(x|z, w_j) P(z|w_j) P(w_j) dz. \tag{12}$$

3 Results of clustering and separability

In this section, we will comment on the obtained results following experiments on a set of benchmarks dataset that can be found in [Nad19]. Unfortunately, due to runtime errors and a lack of computational resources, some datasets haven't been evaluated on the **clustrd** methods.

- **Clustrd:** For the methods available in the 'clustrd' package. We've decided to variate the α parameter to take into account the 3 aspects of the 'cluspca' function (i.e, we've used both FKM and RKM and the tandem approaches). As for the *varimax* parameter, it was by default set to 'True' due to a lack of time for running experiments. In addition, the rotation of the factorial axes is in general a good option to take for the majority of data.

- **Mixture of factor analysis:** For the Mixtures of Factor Analyzers method, we set the number of components at 2. It means we will have a new representation of our data on a two dimensional plan. This new representation of the data can be done by choosing between two different output values: **Uclust** and **Umean**. The EM algorithm associate each class to a distribution. Each individuals has a probability to belong to each distribution. In the new representation, the individuals are represented as a combination of those distribution's probabilities. Uclust takes only the highest probability into consideration while Umean is a mean of all the distribution's probabilities. We decided to work with Umean because it seemed more logical to not ignore the influence of other distributions on an individual even though both output values have very similar results.

3.1 Clustering performances

To evaluate each of the algorithms, we've decided to use the two famous metrics :

- Normalized Mutual Information (NMI)

- Adjusted Rand Score (ARI)

the results were as follows:

CLUSTERING	MFA		ALGO1		Cluspca tandem		FKM		RKM	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
COIL20	**0.7684**	**0.5926**	**0.7928**	**0.6212**	**0.6003515**	**0.3771042**	**0.5069892**	0.2850937	**0.6189331**	0.3950172
YALE	0.4833	0.2174	0.6200	0.3863	0.4152568	0.1500758	0.3902275	0.1516608	0.2529657	0.002340013
ORL	0.8034	0.5190	0.7751	0.4400	0.5468571	0.1630179	0.1736628	**0.571202**	0.3419584	0.3969157
COIL100	0.4711	0.1662	0.5755	0.1420	-	-	-	-	-	-
USPS	0.6387	0.5569	0.5873	0.5038	-	-	-	-	-	-
MNIST	0.4528	0.3272			-	-	-	-	-	-

Table 1 – Results for the clustering metrics across all datasets

Note: Each value in bold indicates the best score across all datasets for a single method. A cell in green indicates that this method outperformed the others on a given dataset acording to the NMI score and the blue color indicates a greater performance according to the ARI score.

3.2 Separability: Evaluation and comparison

In this subsection, we will mainly focus on the dimensionality reduction aspect of each algorithm. The separability scores used are the Silhouette score and the Davies Bouldin score proposed in [DB79]. We can summarize the results like below :

SEPARABILITY	MFA		ALGO1		Cluspca		FKM		RKM		ACP	
	Silhouette	DB	Silhouette	DB	Silhouette	DB	Silhouette	DB	Silhouette	DB	Silhouette	DB
COIL20	-0.1885	1.2811e+15	**0.2189**	**1.7428**	**0.002913041**	**6.840305**	**0.0297**	**6.345815**	**0.003326878**	**5.825479**	**0.2111**	2.974
YALE	-0.1764	**12.9540**	0.1012	1.8758	-0.1245349	10.87603	-0.1268055	10.50099	-0.2826233	15.61628	0.0356	4.422
ORL	**-0.0375**	31.1361	0.1500	1.7465	-0.2472073	9.271798	-0.2478981	9.217204	-0.3058923	35.01148	0.0162	2.5739
COIL100	-0.1813	2.038+e12	0.10067	2.2723	-	-	-	-	-	-	0.0587	3.5712
USPS	-0.23	32.557	0.14313	2.0487	-	-	-	-	-	-	0.0728	**2.4322**
MNIST					-	-	-	-	-	-	0.05911	3.9913

Table 2 – Results for the dimensionality reduction metrics across all datasets

We can notice that PCA has relatively good results in separability. Through this work, it helped us as a 'comparison tool' with the other methods.

Note: Each value in bold indicates the best score across all datasets for a single method. A cell in green indicates that this method outperformed the others on a given dataset according to the Silouhette metric. The blue color of a cell indicates that the method outperformed the others across all datasets as well in terms of the DB score.

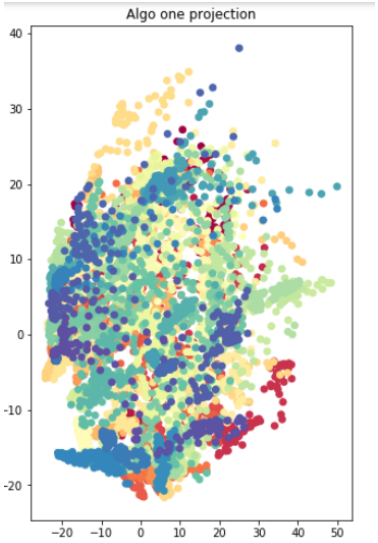

Figure 1 – Projection of COIL100 using the proposed algorithm

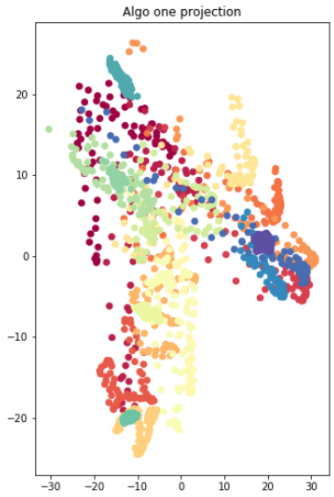

Figure 2 – Projection of COIL20 using the proposed algorithm

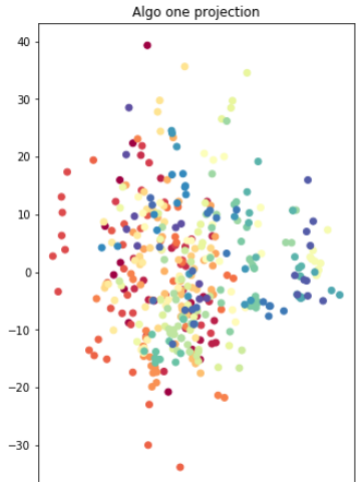

Figure 3 – Projection of ORL using the proposed algorithm

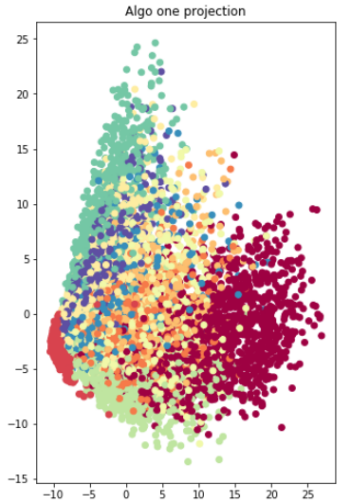

Figure 4 – Projection of USPS using the proposed algorithm

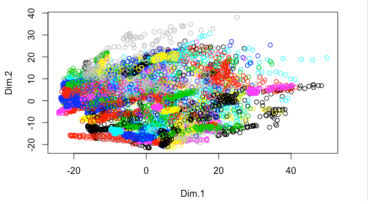

Figure 5 – Projection of COIL100 using PCA

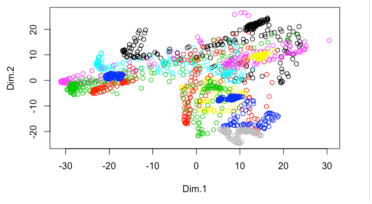

Figure 6 – Projection of COIL20 using PCA

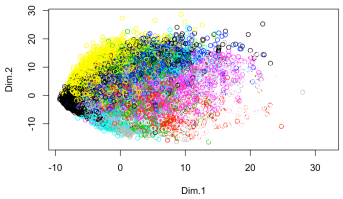

Figure 7 – Projection of mnist using PCA

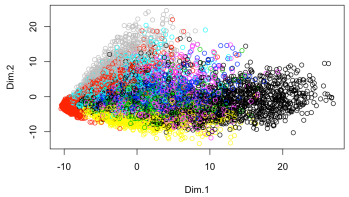

Figure 8 – Projection of USPS using PCA

3.3 Conclusion

Mixture of Factor Analysis We can notice that in the clustering results, the MFA gave us relatively good results, some dataset even had the best scores such as ORL,COIL100 and USPS. However the separability results are terrible compared to the other solutions. MFA is a good method for clustering but a very bad one for data representation and cluster separability.

Cluspca If we take alpha equal to 0.5, the method will execute a FKM that will outperform both the Cluspca tandem and the RKM. We can conclude that a combination of FKM and RKM is not always the best solution and that FKM can be enough to have good results. Overall, results with Cluspca are average for the clustering and quite bad for the separability.

Algorithm1 About the quality of the clustering gave by the nmi and ari value, we see clearly that this algorithm outperform very largely the others the difference range is 0.1 to 0.3, except for MFA method, in this case algorithm one do a little better with COIL20 and Yale dataset but for others MFA is better. When it comes to separability algorithm one gave without contest the best results, outperforming all others methods, in term of silhouette score, we see that there is a slight overlap among clusters silhouette score being near the 0, and DB score gave also good results not too far from 0 (values from 1.7 to 2.2). To illustrate what we said, we can compare the plot of the dataset COIL20 with PCA and algorithm1, PCA can't separate clusters properly (big overlapping), in other hand we can more easily distinguish among clusters with the plot obtained by algorithm one. We didn't do the work on mnist for hardware problem, not enough space in RAM.

References

[DB79] D. L. Davies and D. W. Bouldin. "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (Apr. 1979), pp. 224–227. ISSN: 1939-3539. DOI: 10.1109/TPAMI.1979.4766909.

[Eve84] B. S Everitt. "An Introduction to Latent Variable Models . Chapman and Hall". In: (1984).

[DC94] Geert De Soete and J. Douglas Carroll. "K-means clustering in a low-dimensional Euclidean space". In: *New Approaches in Classification and Data Analysis*. Ed. by Edwin Diday et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 212–219. ISBN: 978-3-642-51175-2.

[Zou96] Geoffrey E. Hinton Zoubin Ghahramani. "The EM Algorithm for Mixtures of Factor Analyzers". In: (1996).

[VK01] Maurizio Vichi and Henk A.L. Kiers. "Factorial k-means analysis for two-way data". In: *Computational Statistics Data Analysis* 37.1 (2001), pp. 49–64. ISSN: 0167-9473. DOI: https://doi.org/10.1016/S0167-9473(00)00064-5. URL: http://www.sciencedirect.com/science/article/pii/S0167947300000645.

[YH14] Michio Yamamoto and Heungsun Hwang. "A General Formulation of Cluster Analysis with Dimension Reduction and Subspace Separation". In: *Behaviormetrika* 41.1 (Jan. 2014), pp. 115–129. DOI: 10.2333/bhmk.41.115. URL: https://doi.org/10.2333/bhmk.41.115.

[MDV19] Angelos Markos, Alfonso D'Enza, and Michel van de Velden. "Beyond Tandem Analysis: Joint Dimension Reduction and Clustering in R". In: *Journal of Statistical Software, Articles* 91.10 (2019), pp. 1–24. ISSN: 1548-7660. DOI: 10.18637/jss.v091.i10. URL: https://www.jstatsoft.org/v091/i10.

[Nad19] Mohamed Nadif. *Reduction de la dimension et Classification regularisee*. available at: https://drive.google.com/open?id=15EZjOPsp7fFlfQY1MrcxOTmzZvKZzBHM (Dec. 2019). 2019.