Une heuristique pour le paramétrage automatique de l'algorithme de *clustering* spectral

Pierrick Bruneau, Olivier Parisot, Philippe Pinheiro

Centre de Recherche Public - Gabriel Lippmann, 41, rue du Brill, L-4422 Belvaux
(bruneau | parisot | pinheiro)@lippmann.lu

Résumé. Trouver le nombre optimal de groupes dans le contexte d'un algorithme de *clustering* est un problème notoirement difficile. Dans cet article, nous en décrivons et évaluons une solution approchée dans le cas de l'algorithme spectral. Notre méthode présente l'avantage d'être déterministe, et peu coûteuse. Nous montrons qu'elle fonctionne de manière satisfaisante dans beaucoup de cas, même si quelques limites amènent des perspectives à ce travail.

1 Introduction

Le *clustering* d'un ensemble d'objets en un nombre de groupes pré-déterminé est un problème souvent difficile suivant le critère d'optimisation ou le modèle choisi. Le choix optimal du nombre de groupes (identifié de manière univoque par la variable k dans le reste de l'article) l'est probablement davantage. Le principe généralement accepté du rasoir d'Occam, favorisant un nombre minimal de *clusters*, s'oppose à leur exhaustivité, sans qu'un compromis satisfaisant pour tous soit possible *a priori*. En pratique, ce paramètre est donc souvent laissé à la discrétion du praticien par les logiciels d'analyse de données, même récents. Dans le cas d'une approche exploratoire, où k peut être inconnu, une heuristique est souhaitable.

Dans cet article, nous nous limitons à l'algorithme de *clustering* spectral, et proposons une nouvelle manière extrêmement simple, peu coûteuse, et bien fondée, d'estimer k à partir du spectre de laplacien propre à cet algorithme. Le test de Bartlett pour l'égalité des variances est utilisé depuis longtemps pour déterminer le nombre de facteurs à retenir dans le contexte d'une Analyse en Composantes Principales (ACP) (James, 1969). Nous montrons qu'il est possible de l'adapter assez facilement pour estimer k dans le contexte de l'algorithme de *clustering* spectral.

Dans un premier temps, nous rappelons l'état de l'art du *clustering* spectral, ainsi que des méthodes d'estimation automatiques de k existantes. Nous décrivons ensuite notre méthode, *in fine* matérialisée par un algorithme simple. L'efficacité de la méthode est illustrée par des expériences sur des données synthétiques et réelles de la littérature. L'analyse critique de nos résultats nous permet de formuler quelques perspectives, données en conclusion.

2 Fondamentaux du *clustering* spectral

Les bases du *clustering* spectral remontent à la théorie des graphes. Il a été popularisé par (Shi et Malik, 2000) et (Ng et al., 2001). Considérant une collection de N éléments, représentée par une matrice symétrique de similarités [1] entre couples d'éléments $\mathbf{S}$, l'algorithme de *clustering* spectral en k groupes peut être résumé comme suit :

- Calcul de la matrice diagonale $\mathbf{D}$, avec $\mathbf{D}_{nn} = \sum_{n'=1}^{N} \mathbf{S}_{nn'}$;
- Calcul du laplacien $\mathbf{L} = \mathbf{D} - \mathbf{S}$;
- Décomposition en valeurs propres de $\mathbf{L}$;
- Extraction des k vecteurs propres mineurs (i.e. les vecteurs propres associés au k plus petites valeurs propres), formant $\mathbf{Y}$ avec ces vecteurs en tant que colonnes ;
- Exécution de l'algorithme k-means sur les lignes de $\mathbf{Y}$, qui produit les étiquettes des éléments respectifs de $\mathbf{S}$;

Algorithme 1 : L'algorithme de *clustering* spectral

Les variantes de cet algorithme ne diffèrent le plus souvent que dans le laplacien utilisé. Le laplacien par défaut, non normalisé, induit des difficultés pratiques (e.g. dépendance au domaine et à la distribution des données) (von Luxburg, 2006). Les normalisations suivantes ont donc été introduites dans la littérature :

$$\text{version symétrique (Ng et al., 2001) : } \mathbf{L}_{\text{sym}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}, \tag{1}$$

$$\text{version } \textit{random walk} \text{ (Shi et Malik, 2000) : } \mathbf{L}_{\text{rw}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{S}, \tag{2}$$

avec $\mathbf{I}$ la matrice identité de taille N. Remarquons que la multiplicité de la valeur propre 0 dans la décomposition spectrale de ces laplaciens peut être interprétée comme le nombre de composantes connexes du graphe sous-jacent (von Luxburg, 2006), i.e. le nombre de *clusters* que forment ses noeuds. Une autre variante notable de normalisation est $\mathbf{L}_{\text{alt}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$ (Zelnik-Manor et Perona, 2004). Une implémentation R récente est d'ailleurs basée sur cette dernière (Karatzoglou et al., 2013). En inspectant l'équation (1), remarquons que $\mathbf{L}_{\text{sym}} = \mathbf{I} - \mathbf{L}_{\text{alt}}$. Conséquemment, l'adaptation de l'algorithme 1 utilisant $\mathbf{L}_{\text{alt}}$ considère les vecteurs propres majeurs, et relie k à la multiplicité de la valeur propre 1.

3 État de l'art sur la détermination du nombre de *clusters*

Le lien entre le paramètre k de l'algorithme 1 et la multiplicité de la valeur propre 0 dans le spectre du laplacien normalisé n'est strictement valable que pour des composantes connexes. Les graphes considérés peuvent cependant contenir des composantes faiblement connectées entre elles, sans être totalement disjointes : par exemple, les similarités calculées *via* une *Radial Basis Function* (RBF) n'égalent jamais exactement 0, induisant nécessairement une seule composante connexe. Le but de l'algorithme est alors précisément d'identifier cette structure.

1. Ces similarités peuvent également être interprétées comme des poids d'arêtes, ou des valeurs de fonction *kernel*, sans nuire à la généralité du propos. La diagonale de cette matrice est conventionnellement nulle.

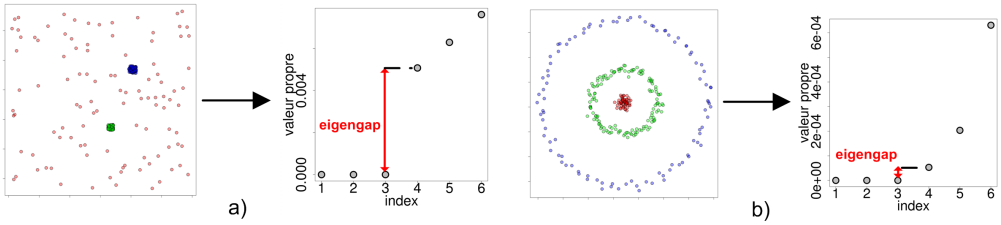


FIG. 1 – *Profil des plus petites valeurs propres pour* synth2 *et* synth1 *(voir la section 5 pour une description).*

Dans le reste du document, pour gérer la variabilité des jeux de données tant en termes de domaine que de distribution, nous utilisons la variante de RBF proposée par (Karatzoglou et al., 2013). Cette dernière adapte le rayon de la fonction à chaque élément selon la médiane de ses K plus proches voisins. Comme préconisé par les auteurs, nous avons retenu $K = 5$ pour nos expériences, ainsi que pour le calcul des spectres présentés dans la figure 1.

La figure 1a montre que le profil des plus petites valeurs propres peut nous renseigner sur la probable valeur optimale de k. Intuitivement, une seule valeur propre égale exactement 0, $k-1$ autres sont *approximativement* égales à 0, et le reste est *significativement* supérieur à 0 : la meilleure valeur de k est ainsi marquée par la différence absolue entre la $k^{\text{ème}}$ et la $(k+1)^{\text{ème}}$ valeur propre, nommée *eigengap* (von Luxburg, 2006).

La plupart des travaux de la littérature détermine l'*eigengap* vraisemblable de manière empirique, soit en comparant les candidats à un seuil arbitraire, soit en analysant le taux de croissance du profil des valeurs propres *via* le *scree test* de Cattell (Cattell, 1966). La figure 1b illustre néanmoins que même dans des cas relativement simples *a priori*, l'application de ce test peut être problématique. Une procédure d'optimisation itérative a également été proposée, mais demeure complexe, tant du point de vue conceptuel que computationnel (Zelnik-Manor et Perona, 2004). Nous proposons une alternative simple et efficace, en adaptant le test de Bartlett pour l'égalité des variances au *clustering* spectral. À l'instar du *scree test*, il était originellement employé à déterminer le nombre de facteurs à extraire dans le contexte d'une ACP (James, 1969).

4 Description de la méthode

Considérant un échantillon de N individus définis sur p variables, l'ACP calcule les q facteurs représentatifs de la matrice de covariance de l'échantillon, en faisant l'hypothèse implicite que des échantillons uni-dimensionnels générés par n'importe lequel des $k = p - q$ facteurs restants doivent avoir une variance identiquement faible. Dans ces conditions, la statistique de test ci-après suit une loi du χ^2 (James, 1969) :

$$-\left(N - 1 - q - \frac{k^2+1}{3k} + \sum_{i=1}^{q} \frac{\bar{\lambda}_k^2}{\left(\lambda_i - \bar{\lambda}_k\right)^2}\right) \ln(V_q) \sim \chi^2_{\frac{(k+2)(k-1)}{2}}, \tag{3}$$

avec λ_i la $i^{\text{ème}}$ valeur propre dans l'ordre décroissant (conventionnel avec l'ACP), $\bar{\lambda}_k$ la moyenne des k valeurs propres mineures, et $V_q = \prod_{i=q+1}^{p}(k\lambda_i/\sum_{j=q+1}^{p}\lambda_j)$. L'algorithme 2 permet ainsi de trouver simplement la plus petite valeur de q acceptable. Cet algorithme est quadratique selon p. Comme la décomposition spectrale est elle-même cubique, le surcoût calculatoire est modeste.

Entrée : Le vecteur des p valeurs propres, un niveau de risque α, e.g. 5%
Résultat : La plus petite valeur de q acceptable
$q \leftarrow 0$;
répéter
  $q \leftarrow q + 1$;
  $s \leftarrow$ statistique de l'équation (3) ;
  /* on contraint $q < p - 1$ car l'équation (3) n'est définie que pour $k > 1$ */
jusqu'à $q = p - 2$ *ou* $P_{\chi^2}(X < s) \leq 1 - \alpha$;
/* on obtient le minimum de q ne menant pas au rejet de l'hypothèse nulle */

Algorithme 2 : Un algorithme simple pour déterminer le nombre de facteurs de l'ACP

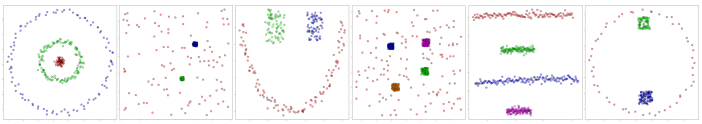
La détermination de k pour l'algorithme de *clustering* spectral est analogue au problème du nombre de facteurs de l'ACP : au lieu de rechercher les q plus grandes valeurs propres d'une matrice de covariance, nous nous intéressons alors aux k plus petites valeurs du spectre d'un laplacien (voir section 2). Il suffit alors d'adapter l'algorithme 2 à la recherche de la plus grande valeur acceptable pour $k = N - q$ (en effet, $N = p$ pour un laplacien). Dans le contexte du *clustering*, $k << N$: il est donc plus efficace de faire démarrer la recherche à $k = 2$, i.e. initialiser q à $p-2$ dans l'algorithme 2, et le décrémenter à chaque itération, avec une condition d'arrêt adaptée.

D'autre part, nous avons constaté empiriquement qu'avec $k << N$, l'ensemble de valeurs propres du laplacien normalisé $\{\lambda_i\}_{i \leq q}$ est très proche de 1 en moyenne : cela permet d'approcher $\sum_{i=1}^{q} \bar{\lambda}_k^2/(\lambda_i - \bar{\lambda}_k)^2$ par $q\bar{\lambda}_k^2/(1-\bar{\lambda}_k)^2$ dans l'équation (3), menant à un critère ne dépendant que des k valeurs propres mineures. En entrelaçant les algorithmes 1 et 2, une extraction incrémentale des valeurs propres depuis les plus petites peut alors être arrêtée précocément. Comme $k << N$, nous obtenons ainsi un algorithme de *clustering* spectral quadratique selon N, incluant la détermination automatique de k.

5 Résultats expérimentaux

Nous avons implémenté notre méthode sous la forme d'un package R, *speccalt* [2], i.e. une *alternative* à la fonction *specc* du package R *kernlab*. L'interface, très simple, ne requiert qu'une matrice de similarité ; le paramètre k, optionnel, est estimé automatiquement en cas d'absence. Nous avons utilisé la normalisation de laplacien $\mathbf{L}_{\text{alt}}$, suggérée dans (Zelnik-Manor et Perona, 2004; Karatzoglou et al., 2013), plus stable en pratique pour réaliser le *clustering*. Toutefois, l'algorithme 2 repose toujours sur $\mathbf{L}_{\text{rw}}$ [3] ; deux laplaciens, ainsi que leurs décompositions respectives, doivent donc être calculés. Les complexités annoncées dans la section 4

2. `http://cran.r-project.org/web/packages/speccalt/index.html`.
3. ou indifféremment $\mathbf{L}_{\text{sym}}$, les deux laplaciens ayant le même spectre.



Jeu de données (vérité terrain)	Notre méthode	État de l'art	indice de Rand corrigé
synth1(3)	**3**	$4 \pm 0{,}00$	$0{,}88 \pm 0{,}18$
synth2(3)	**3**	$5 \pm 0{,}00$	$0{,}97 \pm 0{,}12$
synth3(3)	**3**	$\mathbf{3} \pm 0{,}00$	$0{,}90 \pm 0{,}21$
synth4(5)	**5**	$\mathbf{5} \pm 0{,}00$	$0{,}76 \pm 0{,}18$
synth5(4)	**4**	$\mathbf{4} \pm 0{,}00$	$0{,}89 \pm 0{,}21$
synth6(3)	2	$4 \pm 0{,}00$	$0{,}58 \pm 0{,}00$
iris(3)	2	$4 \pm 0{,}00$	$0{,}54 \pm 0{,}00$
isolet(5)	2	$20 \pm 0{,}00$	$0{,}39 \pm 0{,}00$

FIG. 2 – En haut *: Jeux de données synthétiques de (Zelnik-Manor et Perona, 2004). La coloration des glyphes indique la vérité terrain.* En bas *: Synthèse des résultats expérimentaux. Les moyennes et écarts-types de 20 indices de Rand calculés indépendamment sont indiqués. Le même procédé est appliqué à la méthode de (Zelnik-Manor et Perona, 2004).*

restent cependant valides, et ne changent que d'un facteur constant. Pour l'évaluation, nous avons récupéré les 6 échantillons synthétiques introduits dans (Zelnik-Manor et Perona, 2004) (voir la figure 2), et utilisé deux échantillons UCI, *iris* (150 éléments, 4 variables) et les voyelles de l'échantillon *isolet* (1500 éléments, 617 variables). Les échantillons synthétiques sont nommés de *synth1* à *synth6*, suivant leur position de gauche à droite dans la figure 2. Pour ces expériences, nous avons utilisé notre implémentation de l'algorithme de *clustering* spectral avec estimation automatique de k. Cette estimation, ainsi que l'indice de Rand corrigé (Gordon, 1999, section 7.2.4), sont enregistrés pour chaque jeu de données. À titre de comparaison, nous avons également indiqué les estimations respectives de k obtenues par la méthode de (Zelnik-Manor et Perona, 2004)[4]. Comme l'algorithme 1 est sensible aux minima locaux à travers sa dépendance à k-means, l'indice de Rand corrigé est estimé par 20 exécutions indépendantes sur chaque jeu de données. Le même procédé est appliqué pour la méthode de (Zelnik-Manor et Perona, 2004), eu égard à sa nature itérative. En revanche l'estimation de k par l'algorithme 2 est déterministe. Ces résultats sont résumés dans la figure 2.

Nous constatons d'abord que notre heuristique obtient de meilleurs résultats que la méthode de référence. Elle est satisfaisante dans les premiers cas, mais moins pour *isolet*, *synth6*, et *iris* (2 *clusters* découverts, contre respectivement 5, 3 et 3 d'après la vérité terrain). Ceci est d'ailleurs reflété par une nette dégradation des indices de Rand respectifs. La vérité terrain d'*isolet* n'est pas caractérisée par des frontières de décision tranchées, ce qui induit notre méthode à identifier le nombre minimal de *clusters*. Les cas de *synth6* et *iris* sont plus subtils : en suivant exactement l'algorithme 2, nous aurions identifié respectivement 62 et 29 clusters. En

4. Nous avons utilisé l'implémentation Matlab disponible à `http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html`.

effet, notre méthode ne pénalise pas un nombre excessif de *clusters*, ou l'existence de très petits *clusters* : chaque point du cercle peu dense de *synth6* est ainsi identifié comme un *cluster*. La nature quasiment discrète d'*iris* (i.e. toutes ses valeurs ont au plus une décimale) semble également problématique. Pour pallier ceci, notre implémentation de l'algorithme 2 borne explicitement k par 20. Si $1 - \alpha$ n'est atteint pour aucune des valeurs autorisées, ce seuil est abaissé au plus grand quantile mesuré pour $k \in [2, 20]$. Par souci d'équité, des bornes de valeurs de k identiques ont été imposées à la méthode de (Zelnik-Manor et Perona, 2004).

6 Conclusion

Dans cet article, nous avons proposé une méthode simple, peu coûteuse, et performante pour estimer automatiquement k dans le contexte du *clustering* spectral, ainsi que l'attestent nos résultats expérimentaux. Toutefois, nous avons également identifié des limites à l'approche, par sa focalisation exclusive sur la caractérisation de variétés dans les données.

L'algorithme spectral utilise k-means en tant qu'étape intermédiaire : ce dernier, équivalent à un algorithme EM sur un mélange de gaussiennes isotropes, ouvre la voie à une possible combinaison de notre méthode avec une estimation bayésienne de k, par exemple en utilisant notre heuristique comme *a priori*.

Références

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research 1*(2), 245–276.

Gordon, A. D. (1999). *Classification*. Chapman and Hall.

James, A. T. (1969). Test of equality of the latent roots of the covariance matrix. *Multivariate Analysis, Volume 2*.

Karatzoglou, A., A. Smola, et K. Hornik (2013). *kernlab (R package)*.

Ng, A. Y., M. I. Jordan, et Y. Weiss (2001). On spectral clustering : Analysis and an algorithm. *NIPS*.

Shi, J. et J. Malik (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(8), 888–905.

von Luxburg, U. (2006). A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics.

Zelnik-Manor, L. et P. Perona (2004). Self-tuning spectral clustering. *NIPS*.

Summary

Finding the optimal number of groups in the context of a clustering algorithm is a known as a difficult problem. In this article, we describe and evaluate a heuristic thereof for the spectral clustering algorithm. Our method is deterministic, and remarkable by its low computational burden. We show its effectiveness in most cases. Some limits are identified though, and serve to the formulation of perspectives to this work.