ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

Viral Initialization for Spectral Clustering

Vahan Petrosyan and Alexandre Proutiere

Royal Institute of Technology (KTH) - Department of Automatic Control
Stockholm - Sweden school

Abstract. Spectral Clustering is one of the most widely used clustering algorithms. To find k clusters, it runs the K-means algorithm on the top k eigenvectors of a Laplacian matrix constructed from the data. As a consequence, it inherits the initialization issues of K-means. In this paper, we propose Viral Initialization (VI), a novel initialization procedure implemented in the Spectral Clustering algorithm before K-means is applied. VI is designed so that the resulting clusterings exhibit low normalized cut (Ncuts) values. This design principle is aligned with the recent observation that "good" clusterings have low Ncuts values. We show, through extensive numerical experiments, that the Spectral Clustering algorithm with VI consistently outperforms other state-of-the-art clustering techniques.

1 Introduction

One of the most intriguing questions in cluster analysis is "what is a good clustering?" In other words, how can we distinguish between good and bad clusterings? The problem of making such a distinction is often referred to as the cluster validation problem in the literature. Kannan et al. [4], provides examples where several validation methods fail to find reasonably good clusterings. They further suggest a function of the cluster assignment, which, if minimized, results in clusters that coincide with how human would identify clusters in 2D. This function is closely related to the so-called normalized cuts (Ncuts) measure proposed in [9] and defined as follows.

Consider a dataset $\mathbf{X}^{n\times p} = (\mathbf{x}_i)_{i\in V}$ with $V = \{1,\dots,n\}$. Most popular clustering methods start by constructing a $n \times n$ affinity matrix $\mathbf{A} = (A_{ij})_{i,j\in V}$[1] from the data. Now if C_1 and C_2 are two non-overlapping clusters of the data (i.e., $C_1 \cup C_2 = V$ and $C_1 \cap C_2 = \emptyset$). The Ncuts of this clustering is:

$$\text{Ncuts} = \frac{\text{cut}(C_1, C_2)}{\text{cut}(C_1, V)} + \frac{\text{cut}(C_2, C_1)}{\text{cut}(C_2, V)},$$

where $\text{cut}(C, D) = \sum_{i\in C, j\in D} A_{ij}$ denotes the cut between two subsets C and D of V. A smaller value of the Ncuts means a better clustering. The Ncuts can also be defined for clusterings consisting of k clusters [6], i.e., the Ncuts of clustering $(C_1,\dots,C_k)$ is $\text{Ncuts}_k = \sum_{j=1}^{k} \frac{\text{cut}(C_j, V\setminus C_j)}{\text{cut}(C_j, V)}$. Unfortunately, finding clusterings with minimal Ncuts is a NP-hard problem [9].

Spectral Clustering [7], one of the most popular clustering methods, starts from the affinity matrix, and essentially runs the K-means algorithm on the

[1] E.g., $A_{ij} = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma}\right)$ if $i \neq j$ and 0 otherwise.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

top k eigenvectors of its Laplacian. The spectral method has been shown to output clusterings with low Ncuts [9], but inherits the initialization issues of the K-means algorithm. In this paper, we address this limitation, and propose Viral Spectral Clustering (VSC), an algorithm that proceeds just as the spectral clustering algorithm described above, but includes before running the K-means algorithm an initialization procedure that tends to further direct the output of the algorithm towards clusterings with low Ncuts.

Our initialization procedure, referred to as Viral Initialization (VI), is inspired by the Viral Clustering (VC) algorithm [8], developed to identify the number of clusters present in a dataset. The VC algorithm relies on two antagonist components (Spread Virus and Suppress Virus). The VI procedure ressembles the VC algorithm, but is tailored to run after the spectral decomposition of the affinity matrix. We evaluate the VSC algorithm (based on the VI procedure) on five well-known real-world datasets, and compare its performance to that obtained under spectral algorithms based on other initialization methods. As it turns out, the VSC algorithm outperforms the existing algorithms both in terms of Ncuts and within-cluster sum of squares (WCSS) (remember that the K-means algorithm finds a local minimum of the WCSS).

2 Related Work

We review here some of the most widely used initialization methods for the K-means algorithm. Forgy [3] suggested to initialize the cluster centers by choosing k points from the data uniformly at random. Arthur and Vassilvitskii [1] proposed the K-means++ algorithm for initialization. This algorithm is presumably the most widely used in industry and is the default initialization technique in Matlab and Python. The algorithm can be summarized as follows. Choose the first cluster center as one the data points uniformly at random. Then, the next center is $\mathbf{x}_i$ with probability $\min(d(\mathbf{x}_i))/\sum_{j=1}^{n}\min(d(\mathbf{x}_j))$, where $\min(d(\mathbf{x}))$ denotes the minimum-distance from a point $\mathbf{x}$ to the previously selected centers.

Su and Dy [10] proposed two deterministic K-means initialization algorithms (referred to as PCA-part and VAR-part). In PCA-part, the algorithm first groups all data points in a single cluster. Then, it iteratively selects the cluster with the greatest sum of squares and divides it into two parts by a hyperplane that is perpendicular to the eigenvector corresponding to the largest eigenvalue of the selected cluster covariance matrix. The cutting hyperplane passes through the selected cluster center. Through extensive numerical experiments, Calebi et al. [2] showed that PCA-part outperforms the previous two (and four other) initialization methods in terms of final WCSS.

3 The Viral Spectral Clustering (VSC) Algorithm

In this section, we describe the VSC algorithm, whose pseudo-code is presented in Algorithm 1. VSC relies on the VI procedure, presented in Lines 6 to 13 in Algorithm 1 and that involves two antagonist steps, the Spread Virus and the

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

Suppress Virus steps whose pseudo-codes are presented in Algorithms 2 and 3, respectively.

Algorithm 1: Viral Spectral Clustering (VSC)

1 **Input:** $\mathbf{A} \in \mathbb{R}^{n \times n}, k \in \{1, 2, \cdots, n\}$;
2 $\mathbf{D}$ is a diagonal matrix s.t $\mathbf{D}_{ii} \leftarrow \sum_{i'=1}^{n} \mathbf{A}_{ii'}$;
3 $\mathbf{L} \leftarrow \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$;
4 $\mathbf{E} \leftarrow [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k] \leftarrow eigs(\mathbf{L}, k) \in \mathbb{R}^{n \times k}$;
5 $\mathbf{E} \leftarrow$ normalize each row of $\mathbf{E}$ to 1;
6 **Initialization:** $j \leftarrow 1$, $\mathbf{y} \leftarrow \{1, 2, \cdots, n\}, \mathbf{z} \leftarrow$chunks$(k)$;
7 **repeat** $\mathbf{y} \leftarrow$ Spread_Virus$(\mathbf{A}, \mathbf{y}, z_j)$ **until** $nb\_clusters(\mathbf{y}) \neq z_j$;
8 **while** $nb\_clusters(\boldsymbol{y}) \neq k$ **do**
9 $\quad j \leftarrow j + 1$;
10 $\quad \mathbf{y} \leftarrow$ Suppress_Virus($\mathbf{E}$, $\mathbf{y}$);
11 $\quad$ **repeat** $\mathbf{y} \leftarrow$ Spread_Virus$(\mathbf{A}, \mathbf{y}, z_j)$ **until** $nb\_clusters(\mathbf{y}) \neq z_j$;
12 **end**
13 **repeat** $\mathbf{y} \leftarrow$ Suppress_Virus($\mathbf{E}$, $\mathbf{y}$) **until** convergence;
14 **Output:** $\mathbf{y}$

The VSC algorithm takes as input the affinity matrix and the number of clusters k. VSC is a spectral algorithm [7], and hence starts by computing and renormalizing the top k eigenvectors of the Laplacian of the affinity matrix (refer to Lines 3 to 5 in Algorithm 1). It then runs the VI procedure (Lines 6 to 12), described in the next subsection. Finally, it iterates K-means steps until convergence (Line 13).

3.1 The VI Procedure

The design of the VI procedure is guided by the same principles as those used in the Viral Clustering (VC) algorithm. VC is an algorithm that determines the number of clusters and perform clustering. The algorithm keeps alternating between its two components, Spread Virus and Suppress Virus, until it finds the unknown number of clusters. We modify the Spread Virus component to speed up the spreading rate of clusters.

The VI procedure starts with n clusters, one for each data point, i.e., the initial cluster assignment is $\mathbf{y} = \{1, 2, \cdots, n\}$ (Line 6 in Alg. 1). It then progressively reduces the number of clusters by alternating between the Spread Virus procedure (tending to decrease the number of clusters) and the Suppress Virus procedure (tending to freeze the number of clusters). The Spread and Suppress Virus steps are described later in this section. The pace at which the number of clusters is reduced is dictated by vector $\mathbf{z}$. More precisely the j-th iteration involving the Spread and Suppress steps (Line 10 and 11 in Alg. 1) stops when the number of clusters is reduced to z_j. $\mathbf{z}$ is constructed using the function $chunks(k)$. This function first produces a vector $\mathbf{z}' \in \mathbb{R}^{20}$ such that $z'_1 = 4k$, $z'_{20} = k$, and $z'_i - z'_{i-1} = (21 - i)(z'_{20} - z'_{19})$. Then, $\mathbf{z}$ is obtained by taking the integer part of $\mathbf{z}'$ element-wise. Note that $\mathbf{z}$ vector can have at most 20 elements. The vector $\mathbf{z}$ is constructed that way so that the VI procedure runs the Suppress

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

Virus step more frequently as we get closer to desired number of clusters k. This construction also guarantees that at most 20 iterations of the Suppress Virus step.

Spread Virus. The Spread Virus step proceeds as follows. We randomly sample (without replacement) a data point $\mathbf{x}_l$ from the smallest cluster in the current clustering $\mathbf{y}$. Then, we change its cluster assignment to the cluster of $\mathbf{x}_i$ with probability $p_{l,i} = \frac{A_{li}}{\sum_{j=1}^{n} A_{lj}}$. In other words, the point $\mathbf{x}_l$ has a higher chance to select the cluster of its nearby points. The Spread Virus step stops either when we have sampled all the points in the data, or when the desired number of clusters k is reached. When we repeat the Spread Virus step multiple times, the number of clusters tends to decrease by eliminating the well connected small clusters. Well connected small clusters contribute a lot in increasing the Ncuts value of the clustering. Hence, repeating the Spread Virus step multiple times results in eliminating the clusters with the highest contributions in the Ncuts. In turn, using the Spread Virus steps leads to clusterings with low Ncuts values.

The pseudo-code of Spread Virus is presented in Algorithm 2. There, we denote by $\mathbf{s}$ the set of points that have not been considered yet. $rand\_smallest(\mathbf{s})$ denotes the index of the data point in $\mathbf{s}$ that is randomly selected from the smallest cluster. If $l = rand\_smallest(\mathbf{s})$, then $rand\_change(\mathbf{A}_l)$ samples the index of the data point $\mathbf{x}_i$ whose cluster assignment will be propagated to the data point $\mathbf{x}_l$. This sampling is performed according to the probabilities $p_{l,i}$. $nb\_clusters(\mathbf{y})$ outputs the number of clusters for a given cluster assignment $\mathbf{y}$.

Suppress Virus. This step simply consists in running one iteration of the K-means algorithm starting from the current assignment $\mathbf{y}$. Viral Clustering uses the original data in Suppress Virus steps. Instead here, the Suppress Virus steps are run on the top k eigenvectors $E = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_k]$ of the Laplacian of the affinity matrix. This choice plays a critical role because the new Suppress Virus step will tend to minimize the Ncuts of the resulting clusterings. If there are k' clusters in $\mathbf{y}$, we first compute the k' centers of the eigenvector matrix $\mathbf{E}$, and then assign to each data point the cluster with the closest center. The pseudo-code of this step is presented below in Algorithm 3. The function $centers(\mathbf{E}, \mathbf{y})$ returns k' cluster centers of the eigenvector matrix $\mathbf{E}$.

Algorithm 2: Spread Virus

Input:
$\mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{y} \in \{1, 2, \cdots, n\}^n, k$;
Initialization: $\mathbf{s} = \{1, 2, \cdots, n\}$;
while $s \neq \emptyset$ **do**
  $l \leftarrow$ rand_smallest(**s,** **y**);
  $\mathbf{s} \leftarrow \mathbf{s} \backslash \{l\}$;
  $l' \leftarrow$ rand_change($\mathbf{A}_l$) ;
  $\mathbf{y}_l \leftarrow \mathbf{y}_{l'}$;
  if nb_clusters($\mathbf{y}$) $= k$ **break**
end
Output: $\mathbf{y}$;

Algorithm 3: Suppress Virus

Inputs: $\mathbf{E} \in \mathbb{R}^{k \times n}, \mathbf{y} \in \{1, 2, \cdots, n\}^n$;
$k \leftarrow$ nb_clusters($\mathbf{y}$) ;
$(\mu_1, \ldots, \mu_k) \leftarrow$ centers($\mathbf{E}, \mathbf{y}$) ;
for $l = 1 : n$ **do**
  $y_l \leftarrow \arg\min_{j \in \{1, \ldots, k\}} \|\mathbf{e}_l - \mu_j\|$;
end
Output: $\mathbf{y}$;

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

Complexity. Assume that at the j-th iteration (Lines 10 and 11), there are q clusters. Since the Suppress Virus step corresponds to one iteration of K-means algorithm, its complexity is $\mathcal{O}(npq)$. In Spread Virus, we visit each data point once and change its cluster to the cluster of one of its nearest neighbors. In order to sample from the smallest cluster, we need to sort the clusters (w.r.t. their sizes). Hence, the complexity of Spread Virus is $\mathcal{O}(n + qlog(q))$. In VI, we perform at most 20 Suppress Virus steps, and a limited number of Spread Virus steps (up to 100 steps in our experiments). Exceeding this limited number of steps would be a strong indication that there are more than k well separated clusters in the dataset.

4 Numerical Experiments

We compare the proposed algorithm with three other initialization methods discussed in Section 2. For our experiments we use 5 datasets which can be found in UCI Machine Learning Repository. These datasets include: Letter (n = 20000, p = 16, k = 26), Pen Digit (n = 10992, p = 16, k = 10), MNIST test set (n = 10000, p = 784, k = 10), ISOLET (n = 7797, p = 617, k = 26), Parkinsons (n = 5875, p = 21, k = 42). We used exponential kernel [7], k-nearest neighbor kernel [5] and Self-tuned kernel [11] to construct the affinity matrix. For each kernel and dataset, we run 50 experiments and record the minimum, median and maximum values of Ncuts and WCSS. The results are summarized in Table 1.

Table 1: Summary of experiments

	Kernel →	Exponential kernel ↓						KNN kernel ↓						Self-tuning kernel ↓					
	Objective →	NCuts ↓			WCSS ↓			NCuts ↓			WCSS ↓			NCuts ↓			WCSS ↓		
	Metric →	min	med	max	min	med	max	min	med	max	min	med	max	min	med	max	min	med	max
Data	Algorithms ↓	↓			↓			↓			↓			↓			↓		
Letter	Viral Init	1.88	2.04	2.33	2882	3141	4007	2.05	2.17	2.44	2989	3215	3743	0.81	0.99	1.45	1656	1834	2416
	Random	2.64	3.53	4.50	3019	3519	4024	2.88	3.54	4.25	3119	3484	3892	1.57	2.85	4.20	1955	2330	3301
	K-means++	2.32	3.42	4.30	2900	3425	4227	2.41	3.74	4.43	2976	3498	4100	2.13	2.93	3.62	1979	2325	2930
	PCA-Part	2.78	2.78	2.78	3061	3061	3061	2.87	2.87	2.87	3144	3144	3144	2.07	2.07	2.07	2006	2006	2006
Pendig	Viral Init	0.65	0.65	0.69	402	402	771	0.74	0.74	0.75	422	422	787	0.59	0.60	0.63	395	395	765
	Random	0.65	0.96	2.31	402	816	1410	0.74	0.87	2.35	422	789	1342	0.58	0.75	1.66	395	716	1408
	K-means++	0.65	0.79	1.63	402	785	1410	0.74	0.92	2.38	422	789	1382	0.60	0.78	1.77	395	778	1384
	PCA-Part	0.65	0.65	0.65	402	402	402	0.74	0.74	0.74	422	422	422	0.60	0.60	0.60	395	395	395
mnist	Viral Init	1.53	1.55	1.63	1787	1870	2051	1.66	1.67	1.79	1918	1918	2404	1.40	1.40	1.44	1671	1671	1925
	Random	1.53	1.55	1.76	1787	1915	2358	1.66	1.74	2.02	1918	2242	2853	1.40	1.41	1.87	1671	1735	2537
	K-means++	1.53	1.55	1.91	1787	1960	2611	1.66	1.75	1.95	1918	2268	2525	1.40	1.41	1.88	1671	1735	2546
	PCA-Part	1.59	1.59	1.59	2034	2034	2034	1.80	1.80	1.80	2462	2462	2462	1.59	1.59	1.59	1984	1984	1984
isolet	Viral Init	5.30	5.43	5.88	849	906	1108	5.48	5.68	5.95	868	931	1078	4.97	5.17	5.50	818	888	1023
	Random	5.67	6.97	9.14	939	1280	1812	6.00	7.16	8.86	957	1341	1784	5.53	6.82	8.51	993	1292	1739
	K-means++	6.27	7.01	8.99	1067	1325	1841	6.15	7.19	8.65	1028	1321	1685	5.53	6.82	8.35	883	1335	1752
	PCA-Part	6.30	6.30	6.30	1058	1058	1058	6.01	6.01	6.01	923	923	923	6.25	6.25	6.25	1112	1112	1112
Parkinson	Viral Init	13.35	13.41	13.61	1516	1545	1643	13.36	13.44	13.65	1510	1542	1630	10.98	11.16	11.52	1430	1508	1639
	Random	13.68	14.46	15.95	1584	1767	2032	13.55	14.35	15.56	1604	1777	1975	11.40	12.35	13.30	1521	1703	1891
	K-means++	13.41	14.54	16.28	1574	1777	2183	13.45	14.44	15.80	1567	1778	2015	11.60	12.20	13.31	1535	1723	1907
	PCA-Part	13.80	13.80	13.80	1629	1629	1629	13.56	13.56	13.56	1605	1605	1605	12.37	12.37	12.37	1667	1667	1667

Experiments show that VI achieves the best minimum NCuts and WCSS values for all datasets. The algorithm exhibits high robustness as the median values of VI are comparable or better than the minimum values of its competitors. In some cases, even the worse run (out of 50 experiments) of VI is still better than the best run of the other 3 algorithms.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

VI takes about one second to initialize a dataset with $n = 20000$ observations on Intel(R) Core i7-4600U CPU @ 2.10GHz. This is comparable to the runtime of K-means++ and PCA-part. After VI initialization, K-means algorithm takes fewer iterations to converge on average than the other three methods. Compared to other initialization methods VI exploits the information from the affinity matrix which is necessary to perform spectral clustering. As a result, it consistently achieves lower NCuts and WCSS.

5 Conclusion

In this paper, we proposed a novel procedure to initialize the K-means algorithm in Spectral Clustering algorithms. This procedure inspired by the recently proposed Viral Clustering algorithm is designed so that the resulting clusterings exhibit low Ncuts values, which constitutes a more desirable objective than targeting low WCSS. Indeed, our algorithm outperforms existing initialization methods, and is particularly robust when the number of clusters and dimensions present in the dataset grows large, i.e., when the initialization becomes harder.

References

[1] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, 2007.

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

[3] E. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 22, 1965.

[4] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51(3):497–515, 2004.

[5] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, Dec. 2007.

[6] M. Meila and L. Xu. Multiway cuts and spectral clustering. Technical report, Univercity of Washington, 2003.

[7] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems (NIPS)*, pages 849–856, 2001.

[8] V. Petrosyan and A. Proutiere. Viral clustering: A robust method to extract structures in heterogeneous datasets. *AAAI*, 2016.

[9] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug. 2000.

[10] T. Su and J. G. Dy. In search of deterministic methods for initializing k-means and gaussian mixture clustering. *Intelligent Data Analysis*, 11(4):319 – 338, 2007.

[11] L. Zenlik-Manor and P. Perona. Self-tuning spectral clustering. *Advances in Neural Information Processing Systems (NIPS)*, pages 1601–1608, 2004.