2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

# Careful Seeding based on Independent Component Analysis for k-means Clustering

Takashi Onoda
*System Engineering Lab.*
*Central Research Institute Electric Power Industry*
*Tokyo, JAPAN*
*Email: onoda@criepi.denken.or.jp*

Miho Sakai
*Department of Computational Intelligence and Systems Science*
*Tokyo Institute of Technology*
*Yokohama, JAPAN*
*Email: sakai@ntt.dis.titech.ac.jp*

Seiji Yamada
*Digital Content and Media Sciences Research Division*
*National Institute of Informatics*
*Tokyo, JAPAN*
*Email: seiji@nii.ac.jp*

***Abstract*—The k-means method is a widely used clustering technique because of its simplicity and speed. However, the clustering result depends heavily on the chosen initial value. In this report, we propose a seeding method with independent component analysis for the k-means method. Using a benchmark dataset, we evaluate the performance of our proposed method and compare it with other seeding methods.**

***Keywords*-k-means; k-means++; independent component analysis; seeding;**

## I. Introduction

Clustering is one of the classic problems in machine learning and computational geometry. In the popular k-means formulation, one is given an integer $k$ and a set of $n$ data points in $\mathbf{R}^d$. The goal is to choose $k$ centers so as to minimize the sum of the squared distances between each point and its closest center.

Solving this problem exactly is NP-hard, even with just two clusters [1], but twenty-five years ago, Lloyd[2] proposed a local search solution that is still very widely used today (see for example [3], [4]). Indeed, a recent survey of data mining techniques states that it ”is by far the most popular clustering algorithm used in scientific and industrial applications”[5].

Usually referred to simply as k-means, Lloyd's algorithm begins with $k$ arbitrary centers, typically chosen uniformly at random from the data points. Each point is then assigned to the nearest center, and each center is recomputed as the center of mass of all points assigned to it. These two steps (assignment and center calculation) are repeated until the process stabilizes.

One can check that the sum of the squared distances between each point and its closest center is monotonically decreasing, which ensures that no clustering is repeated during the course of the algorithm. Since there are at most $k^n$ possible clusterings, the process will always terminate. In practice, very few iterations are usually required, which makes the algorithm much faster than most of its competitors.

Unfortunately, the empirical speed and simplicity of the k-means algorithm come at the price of accuracy. There are many natural examples for which the algorithm generates arbitrarily bad clusters. Furthermore, these examples do not rely on an adversarial placement of the starting centers, and the ratio can be unbounded with high probability even with the standard randomized seeding technique.

In this paper, we propose a way of initializing k-means by choosing Independent Component Analysis based starting centers. We also provide preliminary experimental data showing that in practice, our proposed method really does outperform k-means and k-means++ in terms of both accuracy and speed.

## II. Related Works

In this section, we formally define the k-means problem, as well as the k-means and k-means++ algorithms.

For the k-means problem, we are given an integer $k$ and a set of $n$ data points $\chi \subset \mathbf{R}^d$. We wish to choose $k$ centers $\mathcal{C}$ so as to minimize the potential function,

$$\phi = \sum_{\mathbf{x}\in\chi} \min_{\mathbf{c}\in\mathcal{C}} \|\mathbf{x} - \mathbf{c}\|^2.$$

Choosing these centers implicitly defines a clustering for each center, we set one cluster to be the set of data points that are closer to that center than to any other. As noted above, finding an exact solution to the k-means problem is NP-hard. Throughout the paper, we will let $\mathcal{C}_{opt}$ denote the optimal clustering for a given instance of the k-means problem, and we will let $\phi_{opt}$ denote the corresponding potential. Given a clustering $\mathcal{C}$ with potential $\phi$, we also let $\phi(\mathcal{A})$ denote the contribution of $\mathcal{A} \subset \chi$ to the potential (i.e., $\phi(\mathcal{A}) = \sum_{x\in\mathcal{A}} \min_{\mathbf{c}\in\mathcal{C}} \|\mathbf{x} - \mathbf{c}\|^2$).

978-0-7695-4191-4/10 $26.00 © 2010 IEEE
DOI 10.1109/WI-IAT.2010.102

### A. The k-means algorithm

The k-means method is a simple and fast algorithm that attempts to locally improve an arbitrary k-means clustering. It works as follows.

1) Arbitrarily choose $k$ initial centers $\mathcal{C} = \mathbf{c}_1, \ldots, \mathbf{c}_k$.
2) For each $i \in \{1, \ldots, k\}$, set the cluster $\mathbf{c}_i$ to be the set of points in $\chi$ that are closer to $\mathbf{c}_i$ than they are to $\mathbf{c}_j$ for all $j \neq i$.
3) For each $i \in \{1, \ldots, k\}$, set $\mathbf{c}_i$ to be the center of mass of all points in $\mathbf{C}_i$: $\mathbf{c}_i = \frac{1}{|\mathbf{C}_i|}\sum_{x\in\mathbf{C}_i}\mathbf{x}$.
4) Repeat Steps 2) and 3) until $\mathcal{C}$ no longer changes.

It is standard practice to choose the initial centers uniformly at random from $\chi$. For Step 2), ties may be broken arbitrarily, as long as the method is consistent. Steps 2) and 3) are both guaranteed to decrease $\phi$, so the algorithm makes local improvements to an arbitrary clustering until it is no longer possible to do so. To see that Step 3) does in fact decreases $\phi$, it is helpful to recall a standard result from linear algebra (see [14]).

The k-means algorithm is attractive in practice because it is simple and it is generally fast. Unfortunately, it is guaranteed only to find a local optimum, which can often be quite poor.

### B. The k-means++ algorithm

The k-means algorithm begins with an arbitrary set of cluster centers. We propose a specific way of choosing these centers. At any given time, let $D(\mathbf{x})$ denote the shortest distance from a data point $\mathbf{x}$ to the closest center we have already chosen. Then, we define the following algorithm, which we call k-means++[6].

1a) Choose an initial center $\mathbf{c}_1$ uniformly at random from $\chi$.
1b) Choose the next center $\mathbf{c}_i$, selecting $\mathbf{c}_i = \mathbf{x}' \in \chi$ with probability $\frac{D(\mathbf{x}')^2}{\sum_{\mathbf{x}\in\chi} D(\mathbf{x})^2}$.
1c) Repeat Step 1)b until we have chosen a total of $k$ centers.

Step 2)-4) proceed as with the standard k-means algorithm. We call the weighting used in Step 1b) simply "$D^2$ weighting".

## III. Proposed Method

This section describes a problem for k-means and k-means++ algorithms. Then, we proposes k-means with Independent Component Analysis(ICA) based seeding algorithm.

### A. What is a problem for k-means and k-means++ algorithms

Now, we have 6 points data which consist of $\mathbf{x}_i$, $i = 1, \ldots, 6$ and these points are divided into two clusters. Figure 1 shows these 6 points. And Figure 2 shows the global optimal clustering result for these 6 points data. The first cluster consists of $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5\}$ and the other cluster consists of $\{\mathbf{x}_3, \mathbf{x}_6\}$. We are expecting that clustering algorithms can find the global optimal clusters. But, k-means algorithm generates bad clusters, if $\mathbf{x}_2$ and $\mathbf{x}_5$ are chosen as initial cluster centers $\mathbf{c}_1$ and $\mathbf{c}_2$. Figure 3 shows local optimal clusters, which are bad clusters. The k-means++ algorithm was proposed to avoid this bad clustering. But, the k-means++ algorithm sometimes generates bad clusters, because the algorithm depends on choosing the initial center $\mathbf{c}_1$. The initial center $\mathbf{c}_1$ is chosen uniformaly at random from $\chi$.

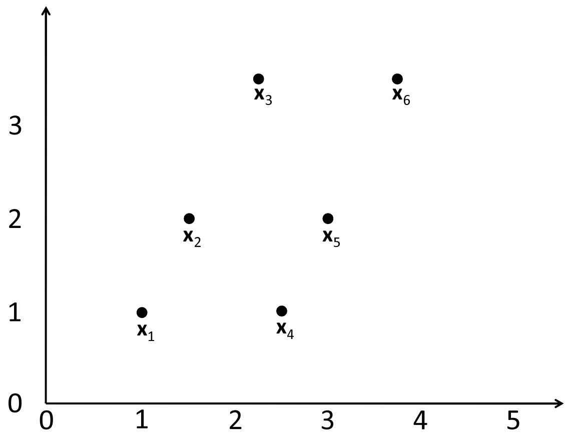

Figure 1. Given Data

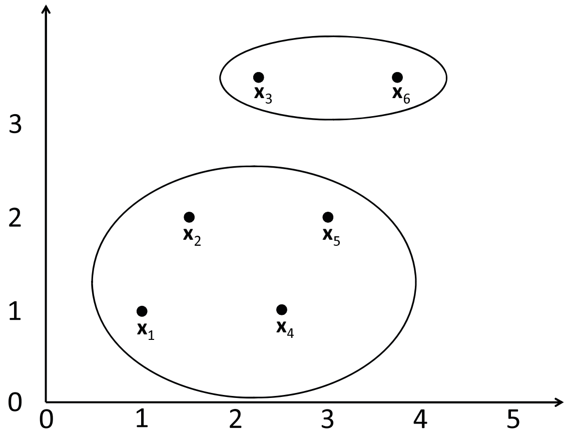

Figure 2. Global Optimal Custering Case

### B. The k-means with ICA based seeding algorithm

The k-means algorithm begins with an arbitrary set of cluster centers. The k-means++ algorithm begins with a little bit arbitrary set of cluster centers. We propose a specific way of choosing these centers. At any given time, we can get independent components(ICs) from given data $\mathbf{x}$. Then, we define the following algorithm.

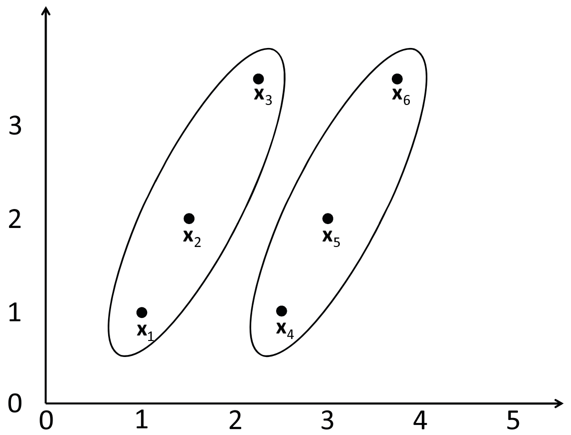

Figure 3. Local Optimal Clustering Case

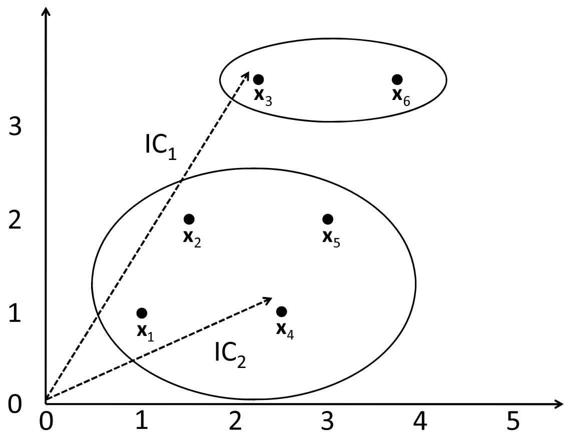

Figure 4. The Concept of our proposed method

1a) Extract $k$ independent components $\mathbf{IC}_m$, $m = 1, \ldots, k$ from given data $\mathbf{x}$.
1b) Choose an initial center $\mathbf{c}_1$, selecting $\mathbf{c}_1 = \mathbf{x}' \in \chi$ with minimum $\frac{\mathbf{IC}_1 \cdot \mathbf{x}'}{|\mathbf{IC}_1||\mathbf{x}'|}$.
1c) Choose the next center $\mathbf{c}_i$, selecting $\mathbf{c}_i = \mathbf{x}' \in \chi$ with minimum $\frac{\mathbf{IC}_i \cdot \mathbf{x}'}{|\mathbf{IC}_i||\mathbf{x}'|}$.
1d) Repeat Step 1c) until we have chosen a total of $k$ centers.

Step 2)-4) proceed as with the standard k-means algorithm. Figure 4 shows the concept of the k-means with ICA based seeding algorithm.

## IV. Experiments

In order to evaluate k-means++ and the proposed method in practice, we have implemented and tested them in matlab. In this section, we discuss the results of these preliminary experiments. We found that the k-means with ICA based seeding makes good performance of both the accuracy and the speed.

### A. Data sets

We evaluated the performance of k-means, k-means++ and the proposed method on three data sets of UCI Machine Learning repository. The first data set, *iris*, is consists of 50 samples from each of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample, they are the length and the width of sepal and petal. Based on the combination of the four features, Fisher developed a linear discriminant model to determine which species from these four measurements. It is used as a typical test for many other classification techniques.

The second dataset, *wine*, is the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

The third dataset, *soybean-small*, is for four soybean disease diagnosis. The dataset is consists of 47 samples and 35 attributes.

### B. Evaluation metrics

We used Normalized Mutual Information as a metric to evaluate the qualities of clustering outputs of different methods. The Normalized Mutual Information measures the consistency of the clustering output compared to the ground truth. It reaches the maximum value of 1 only if the membership $\phi_c$ perfectly matches $\phi_g$ and the minimal zero if the assignments of $\phi_c$ and $\phi_g$ are independent. The membership function $\phi_c(\mathbf{x})$, the mapping of a point $\mathbf{x}$ to one of the $k$ clusters. The membership $\phi_g(\mathbf{x})$ represents the true cluster label for $\mathbf{x}$. Formally, the Normalized Mutual Information is the following equation.

$$NMI(\phi_g, \phi_c) = \frac{MI(\phi_g, \phi_c)}{\max(H(\phi_g), H(\phi_c))},$$

where $MI(\phi_g, \phi_c)$ denotes

$$MI(\phi_g, \phi_c) = \sum_{i=1}^{k} \sum_{j=1}^{k} p_{g,c}(i,j) \log \frac{p_{g,c}(i,j)}{p_g(i)p_c(j)},$$

$H(\phi_g)$ is

$$H(\phi_g) = \sum_{i=1}^{k} p_g(i) \log \frac{1}{p_g(i)},$$

and $H(\phi_c)$ denotes

$$H(\phi_c) = \sum_{j=1}^{k} p_c(j) \log \frac{1}{p_c(j)}.$$

The $p_g(i)$ is the percentage of points in cluster $i$ according to the ground truth, i.e. $p_g(i) = \frac{\sum_{l=1}^{n} 1(\phi_g(\mathbf{x}_l) - i)}{n}$. Similarly,

Table I
EXPERIMENTAL RESULTS FOR *iris* DATASET

| | avg $NMI$ | max $NMI$ | min $NMI$ | $NMI$ with min variance |
|---|---|---|---|---|
| k-means | 0.70325 | 0.751485 | 0.532224 | 0.751485 |
| k-means++ | 0.749295 | 0.751485 | 0.532471 | 0.751485 |
| ICA | 0.751485 | - | - | - |

Table II
EXPERIMENTAL RESULTS FOR *wine* DATASET

| | avg $NMI$ | max $NMI$ | min $NMI$ | $NMI$ with min variance |
|---|---|---|---|---|
| k-means | 0.417794 | 0.428701 | 0.3873 | 0.428701 |
| k-means++ | 0.418351 | 0.428701 | 0.3873 | 0.428701 |
| ICA | 0.428701 | - | - | - |

Table III
EXPERIMENTAL RESULTS FOR *soybean-small* DATASET

| | avg $NMI$ | max $NMI$ | min $NMI$ | $NMI$ with min variance |
|---|---|---|---|---|
| k-means | 0.714445 | 1 | 0.518038 | 0.710813 |
| k-means++ | 0.806213 | 1 | 0.710813 | 0.710813 |
| ICA | 0.710813 | - | - | - |

$p_c(j) = \frac{\sum_{l=1}^{n} 1(\phi_c(\mathbf{x}_l)-j)}{n}$ and $p_{g,c}(i,j)$ is the percentage of points that belong to cluster $i$ in $\phi_g$ and also cluster $j$ in $\phi_c$, i.e. $p_{g,c}(i,j) = \frac{\sum_{l=1}^{n} 1(\phi_g(\mathbf{x}_l)-i)1(\phi_c(\mathbf{x}_l)-j)}{n}$.

The above defined metrics were used to evaluate the accuracy of the clustering algorithms.

### C. Experimental results

Each of the k-means and k-means++ methods was run 100 times with different initializations over all the datasets. And the proposed method was run just one time, because this method does not need an initial value. Table IV shows the experimental results of *iris* dataset, Table V shows the experimental results of *wine* dataset, and Table VI shows the experimental results of *soybean-small* dataset. These tables have the averaged $NMI$, the maximum $NMI$, the minimum $NMI$, and the $NMI$ when the clusters achieved the minimum variance.

Table IV, V show that the proposed algorithm outperforms both k-means and k-means++ algorithms. In Table IV, V, the $NMI$ of our proposed algorithm is as same as the $NMI$ of the maximum performance of the other algorithms and is achieved by just one calculation.

Table VI shows that the $NMI$ of our proposed algorithm is as same as the $NMI$ when the clusters achieved the minimum variance. This situation shows that the performance of our proposed algorithm is as same as the performance of the other algorithms for *soybean-small* data set.

## V. CONCLUSION

This paper proposed k-means with Independent Component Analysis(ICA) based seeding algorithm. In our experimental results, our proposed algorithm shows the better performance than the other algorithms or the performance of our proposed algorithm is as same as the performance of the others. In our future work, we will improve our proposed algorithm to improve the accuracy.

## REFERENCES

[1] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Mach. Learn.*, vol. 56, no. (1-3), pp. 9–33, 2004.

[2] S. P. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–136, 1982.

[3] P. K. Agarwal and N. H. Mustafa, "k-means projective clustering," in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems,*. New York, NY, USA: ACM Press, 2005, pp. 155–165.

[4] R. Herwig, A. Poustka, C. Müller, C. Bull, H. Lehrach, and J. O'Brien, "Large-scale clustering of cdna-fingerprinting data," *Genome Research*, vol. 9, pp. 1093–1105, 1999.

[5] P. Berkhin, "Survey of clustering data mining techniques," Accrue Software, San Jose, CA, Tech. Rep., 2002.

[6] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms,*. New Orleans, Louisiana, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.