Notes on k-Means

Paul F. Roysdon, Ph.D.

I. Introduction

k-Means Clustering is an iterative, unsupervised learning algorithm that performs data-partitioning of n observations to exactly one of k cluster centroids. Solving this problem exactly is NP-hard, but Lloyd's algorithm [1] improves the algorithm with a local search solution. The k-means++ [2] algorithm further improves this approach in terms of both accuracy and speed, often by a substantial margin.

Simply stated, k-Means finds the best centroids by alternating between, (1) assigning data points to clusters based on the current centroids, (2) choosing centroids (points which are the center of a cluster) based on the current assignment of data points to clusters.

Given k, the algorithm performs the following:

- Chose k-centroids at random, for cluster initialization.
- Compute distances from each centroid to each n observation.
- Compute the average sum-of-squares point-to-cluster-centroid distances.
- Re-assign individual observations to a different cluster.
- Re-compute the sum-of-squares for each cluster to obtain k new centroid locations.
- Repeat until cluster assignments do not change, or the maximum number of iterations is reached.

The two-phase method described above, is analogous to the Expectation-Maximization (EM) algorithm [3], [4]. EM computes the Expected Value and Maximum Likelihood Estimate (MLE), hence EM, by assigning a probability distribution to the observations. Note: for data with known or assumed distribution, the standard EM algorithm is preferred over k-Means.

II. Algorithm Description

A. Optimization

Define the k-Means cost function as

$$\mathcal{J}(w_{ij}, \mu_j) = \underset{w,\mu}{\mathit{arg\,min}} \sum_{i=1}^{m} \sum_{j=1}^{k} w_{ij} \|x_i - \mu_j\|^2,$$

where μ_j is the centroid of cluster x_i, and $w_{ij} = 1$ if observation x_i belongs to cluster k, otherwise $w_{ij} = 0$.

Because the cost function has two independent variables, the minimization problem is solved in two steps. First minimize $\mathcal{J}$ w.r.t. w_{ij} while μ_j is fixed. Then minimize $\mathcal{J}$ w.r.t. μ_j while w_{ij} is fixed. Recall that optimization requires differentiation w.r.t. to the variable to be minimized.

1) Step-1: Solve and update the cluster assignments, e.g. assign the observation x_i to the closest cluster k based on the sum-squared distance from the cluster centroid.

$$\frac{\partial \mathcal{J}}{\partial w_{ij}} = \sum_{i=1}^{m} \sum_{j=1}^{k} \|x_i - \mu_j\|^2 = 0$$

$$\implies w_{ij} = \begin{cases} 1, & \text{if } k = \underset{j}{\mathit{arg\,min}} \|x_i - \mu_j\|^2 \\ 0, & \text{otherwise} \end{cases}$$

2) Step-2: Recompute the centroid of each cluster to reflect the new assignments.

$$\frac{\partial \mathcal{J}}{\partial \mu_j} = 2 \sum_{i=1}^{m} w_{ij}(x_i - \mu_j) = 0$$

$$\implies \mu_j = \frac{\sum_{i=1}^{m} w_{ij} x_i}{\sum_{i=1}^{m} w_{ij}}$$

Notes:

- In some cases it is recommended to normalize the data, e.g. zero-mean and standard deviation of one, to minimize errors in the distance calculation.
- Random initialization of the iterative k-Means algorithm may lead to convergence to a *local* optimum instead the *global* optimum. Performance can be aided by improved initialization methods [2]. Robust implementations solve the algorithm with several random initializations, called *replicates* [2], and select the result with the lowest sum-squared distance.

B. Distance

The distance from centroid to observation is computed by one of five common methods.

1) Euclidean distance: The Euclidean distance algorithm is most common in k-Means implementations. Squared Euclidean distance, i.e. L_2-norm, defines each centroid as the mean of the points in that cluster.

$$d(\mathbf{x}, \mathbf{c}) = (\mathbf{x} - \mathbf{c})(\mathbf{x} - \mathbf{c})^{\mathsf{T}}$$

2) Absolute differences: Sum of absolute differences, i.e. L_1-norm, defines each centroid as the component-wise median of the points in that cluster.

$$d(\mathbf{x}, \mathbf{c}) = \sum_{j=1}^{p} |x_j - c_j|$$

3) Cosine: One minus the cosine of the included angle between points (treated as vectors), such that each centroid is the mean of the points in that cluster, after normalizing those points to unit Euclidean length.

$$d(\mathbf{x},\mathbf{c}) = 1 - \frac{\mathbf{x}\mathbf{c}^{\mathsf{T}}}{\sqrt{(\mathbf{x}\mathbf{x}^{\mathsf{T}})(\mathbf{c}\mathbf{c}^{\mathsf{T}})}}$$

4) Correlation: One minus the sample correlation between points (treated as sequences of values), such that each centroid is the component-wise mean of the points in that cluster, after centering and normalizing those points to zero mean and unit standard deviation.

$$d(\mathbf{x},\mathbf{c}) = 1 - \frac{(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{c}-\hat{\mathbf{c}})^{\mathsf{T}}}{\sqrt{((\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^{\mathsf{T}})}\sqrt{((\mathbf{c}-\hat{\mathbf{c}})(\mathbf{c}-\hat{\mathbf{c}})^{\mathsf{T}})}}$$

where

$$\hat{\mathbf{x}} = \frac{1}{p}\left(\sum_{j=1}^{p} x_j\right)\boldsymbol{\gamma}_p$$

$$\hat{\mathbf{c}} = \frac{1}{p}\left(\sum_{j=1}^{p} c_j\right)\boldsymbol{\gamma}_p$$

$$\boldsymbol{\gamma}_p = \text{ ones vector of length p.}$$

5) Hamming distance: Note: this metric is only suitable for binary data. Hamming distance is the proportion of bits that differ, where each centroid is the component-wise median of points in that cluster.

$$d(\mathbf{x},\mathbf{c}) = \frac{1}{p}\sum_{j=1}^{p}\mathcal{I}(x_j \neq y_j)$$

where $\mathcal{I}$ is the indicator function.

III. Example and Applications

Consider the generic example shown in Fig. 1 (a). An obvious clustering of two events are present, with unknown mean and distribution. Applying k-means, we can determine the centroid (mean) of the two clusters, for prediction of future events. Iteration 0 (Initialization), Fig. 1 (b), the cluster centroids k are randomly initialized. Iteration 1a (Step-1), Fig. 1 (c), the distances are computed for the selection of observations n to a cluster k. Iteration 1b (Step-2), Fig. 1 (d), the centroids are re-computed. Iteration 2a (Step-1), Fig. 1 (e), the distances are again computed. Iteration 2b (Step-2) and the final iteration, Fig. 1 (d), the centroids are re-computed.

This algorithm is very popular and used in a variety of applications such as market segmentation, document clustering, image segmentation and image compression, etc. Figs. 2-5 demonstrate the application of k-Means to a color photo with image segmentation (compression) using $2, 4, 8$ clusters (colors).

IV. Advantages & Limitations

Advantages

- Fast, robust, simple to understand and implement.
- Relatively efficient: $O(tknd)$, where n is number objects, k is number clusters, d is number dimension of each object, and t is number iterations. Normally, k, t, $d << n$.
- Gives best result when data are distinct or well separated from each other.

Limitations

- The learning algorithm requires *a-priori* specification of the number of cluster centroids.
- The use of *Exclusive Assignment*, e.g. if there are two highly overlapping data, then k-means will not be able to resolve two clusters, see Fig. 6.
- The learning algorithm is not invariant to non-linear transformations, i.e. with different representation of data we get different results. Data represented in different reference frames, e.g. Cartesian co-ordinates and polar co-ordinates, will give different results.
- Not applicable to categorical data.
- Unable to handle noisy data and outliers.
- Algorithm fails for non-clustered data, e.g. two crescents facing each other in close proximity, see Fig. 6.

References

[1] S. P. Lloyd, "Least Squares Quantization in PCM." *IEEE Transactions on Information Theory*, vol. 28, no. 1, pp. 129–137, 1982.

[2] D. Arthur and S. Vassilvitskii, "K-means++: The Advantages of Careful Seeding." *SODA 07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035, 2007.

[3] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol. I - Estimation Theory*. Prentice Hall PTR, 2013.

[4] ——, *Fundamentals of Statistical Signal Processing, Vol. II - Detection Theory*. Prentice Hall PTR, 1998.

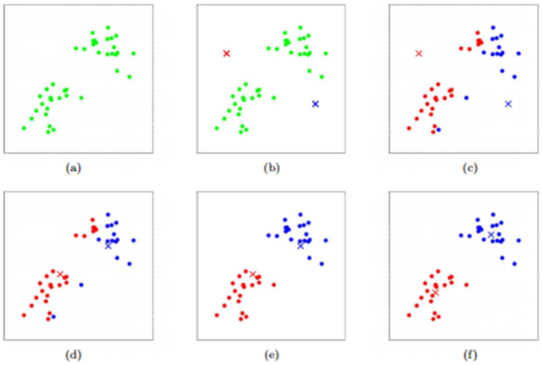

Fig. 1. k-Means steps a-f, with color-coded clusters at each step.

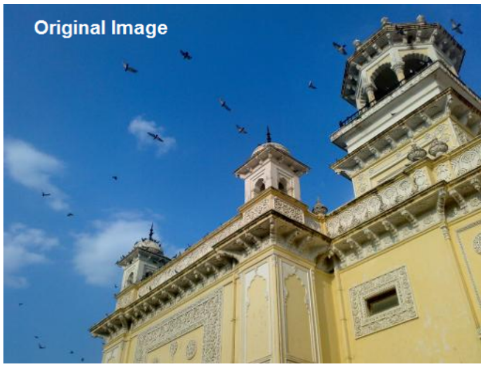

Fig. 2. Original full-color photo.

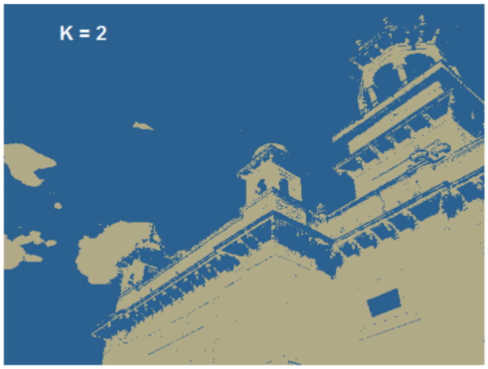

Fig. 3. k-Means with k=2.

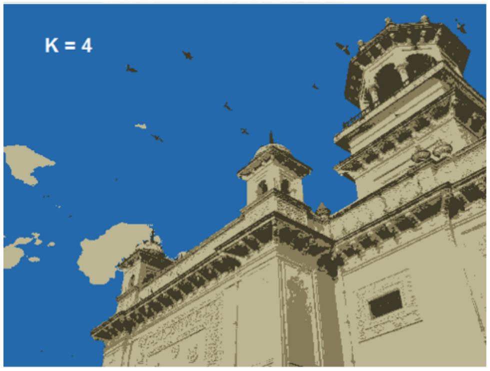

Fig. 4. k-Means with k=4.

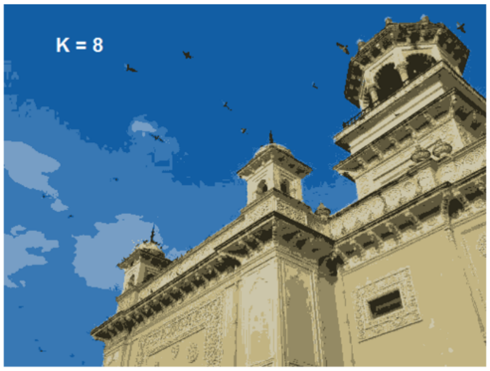

Fig. 5. k-Means with k=8.

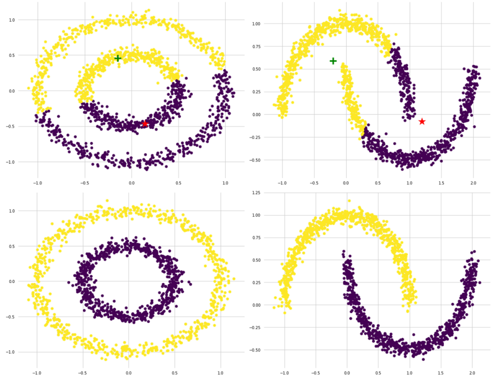

Fig. 6. Incorrect and correct classification of circles and crescents.