Handbook of Cluster Analysis
Chapman & Hall/CRC Handbooks of Modern Statistical Methods
Hennig Christian, Meila Marina, Murtagh Fionn, Rocci Roberto

Quadratic Error and k-Means

Boris Mirkin

Published online on: 01 Dec 2015

How to cite: Boris Mirkin. 01 Dec 2015, Quadratic Error and k-Means from: Handbook of Cluster Analysis, Chapman & Hall/CRC Handbooks of Modern Statistical Methods CRC Press
Accessed on: 21 Dec 2018
3

Quadratic Error and k-Means

Boris Mirkin

CONTENTS

Abstract .................................................. 33
3.1 Conventional k-Means Clustering ......................... 34
  3.1.1 General ............................................. 34
  3.1.2 Three Frameworks for the Square-Error Criterion .... 34
    3.1.2.1 Naive Framework .............................. 35
    3.1.2.2 Approximation Framework ..................... 38
    3.1.2.3 Probabilistic Modeling Framework .......... 39
3.2 Equivalent Reformulations of k-Means Criterion ....... 41
  3.2.1 Anomalous Clustering Criterion .................... 41
  3.2.2 Inner-Product k-Means Criterion .................. 42
  3.2.3 Kernel-Wise Criterion of Maximization of the Semi-Averaged Internal Similarities .......... 42
  3.2.4 Spectral Rayleigh Quotient Formulation ............ 42
3.3 Challenges for k-Means Criterion and Algorithm ....... 43
  3.3.1 Properties of the Criterion ....................... 43
    3.3.1.1 Convexity of Cluster Domains ............... 43
    3.3.1.2 Monotonicity of the Minimum with Respect to the Number of Clusters K ........... 43
    3.3.1.3 Dependence on the Feature Measurement Scales .......... 43
    3.3.1.4 Propensity to Balance the Sizes of Clusters .......... 44
    3.3.1.5 Square Error Versus Consensus ................ 44
  3.3.2 Different Clustering Structures .................... 45
  3.3.3 Initialization of k-Means ........................ 46
  3.3.4 Getting a Deeper Minimum ........................ 47
  3.3.5 Interpretation Aids ................................ 48
  3.3.6 Feature Weighting, Three-Stage k-Means, and Minkowski Metric ......................... 49
3.4 Notes on Software for k-Means ......................... 50
Acknowledgments ........................................ 52
References .............................................. 52

Abstract

This chapter presents an updated review of k-means clustering, arguably the most popular clustering method. First, the square-error k-means criterion and method are introduced in three frameworks: a naive one, data recovery, and mixture of distributions. Then several
equivalent reformulations are given as leading to different local optimization strategies. A number of challenges and ways for addressing them are discussed as related to both the properties of solutions and ways for optimum finding. A few extensions are mentioned including fuzzy clustering and feature weighting.

3.1 Conventional k-Means Clustering

3.1.1 General

Currently, k-means is the most popular clustering technique. Amazingly, when looking through Google web browser on July 2, 2012, query “k-means” returned 234 million pages, whereas query “clustering” has led to less than 26 million web pages, which has been somewhat alleviated by 208 million pages returned for query “cluster.” Such a popularity probably is because thousands of data analysis practitioners in banks, marketing research, pharmaceuticals, etc. are using “k-means” software without much regard for more general aspects of clustering.

The method became known in late sixties after a first theoretical result was proved in MacQueen (1967). Later predecessors have been found; two histories of the method, with different emphases, are now available (see Bock (2008) and Steinley (2006)). The method is present, in various forms, in major statistics packages such as SPSS (Green and Salkind 2003) and SAS (Der and Everitt 2001) and data mining packages such as Clementine (Clementine 2003) and DBMiner (Han et al. 2011), as well as freeware like Weka (Witten 2011). It is described in all materials on clustering including most classical texts such as in Hartigan (1975) and Jain and Dubes (1988). The algorithm is appealing in many aspects. It is computationally easy, fast, and memory-efficient. Conceptually, this method may be considered a model for the cognitive process of making a data-conditioned typology. Also, it has nice mathematical properties. However, there are some issues too, most urgently with respect to the initial setting and stability of results. Some questions arise with respect to the meaning and relevance of the k-means criterion.

3.1.2 Three Frameworks for the Square-Error Criterion

The square-error clustering criterion can be applied in various settings, for various dataset and cluster structure formats. The generic k-means applies to a most common data format, a set of entities $i \in I$ presented as $m$-dimensional space points $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ (see rows of Table 3.1).

**TABLE 3.1**

<table>
<thead>
<tr>
<th>Points</th>
<th>$v_1$</th>
<th>$v_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p1$</td>
<td>1.0</td>
<td>2.0</td>
</tr>
<tr>
<td>$p2$</td>
<td>1.5</td>
<td>2.2</td>
</tr>
<tr>
<td>$p3$</td>
<td>3.0</td>
<td>1.3</td>
</tr>
<tr>
<td>$p4$</td>
<td>2.5</td>
<td>1.1</td>
</tr>
<tr>
<td>$p5$</td>
<td>0.9</td>
<td>2.1</td>
</tr>
<tr>
<td>$p6$</td>
<td>2.0</td>
<td>1.5</td>
</tr>
<tr>
<td>$p7$</td>
<td>2.5</td>
<td>1.4</td>
</tr>
</tbody>
</table>
A two cluster clustering at k-means ($K = 2$) can be variously defined by specifying (i) two cluster centers, $c_1$ and $c_2$, or (ii) a two cluster partition on $I$, $S = \{S_1, S_2\}$ so that $S_1 \cup S_2 = I$ and $S_1 \cap S_2 = \emptyset$, or (iii) by both. In general, centers can belong to a different space, as for example, when they characterize a parametric regression function over a cluster. Here, they are assumed to belong to the space of points, as usual, for example, $c_1 = (1, 2.5)$ and $c_2 = (3, 2)$. To score the goodness of a clustering $(S, c)$ with $K$ clusters $S_k$ and centers $c_k$ ($k = 1, 2, \ldots, K$), different frameworks can be used of which the following three prevail: (i) a naive one, (ii) data recovery (approximation), and (iii) probabilistic modeling.

### 3.1.2.1 Naive Framework

It assumes that there is a distance or similarity function defined between entity points $x_i$ and centers $c_k$, the most popular choice being the squared Euclidean distance

$$d(x_i, c_k) = \sum_{h=1}^{m} (x_{ih} - c_{kh})^2$$  \hspace{1cm} (3.1)

The k-means square-error criterion is defined as the summary distance between entities and their cluster centers:

$$W(S,c) = \sum_{k=1}^{K} \sum_{i \in S_k} d(x_i, c_k) = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{h=1}^{m} (x_{ih} - c_{kh})^2$$ \hspace{1cm} (3.2)

This criterion is illustrated in Figure 3.1: each of the points $x_i$ relates to the center of its cluster so that the total number of distances in the sum is $n = |I|$ and does not depend on the number of clusters $K$. 

![Figure 3.1](image)
The smaller the value $W(S, c)$, the better the clustering $(S, c)$. Although, finding optimal pair $(S, c)$ is quite a computationally intensive task, finding either optimal $S$ or optimal $c$ is fairly easy, as the following statements say.

### 3.1.2.1.1 Centers Set

Given partition $S = \{S_k\}$, the optimal centers $c_k$, with respect to Equation 3.2, are computed as the mean points in $S_k$:

\[
    c_k = \frac{\sum_{i \in S_k} x_i}{n_k}
\]

(3.3)

where $n_k$ is the number of elements in $S_k$, $k = 1, \ldots, K$.

The means are aggregate representations of clusters and as such they are referred to as standard points or centroids or prototypes or seeds—these are considered synonymous in clustering.

With the optimal centers in Equation 3.3, a controversial issue of simultaneously processing quantitative and categorical features gets a positive resolution with the $k$-means criterion. Indeed, given a category $v$ such that a subset $S_v \subseteq I$ falls in it, let us introduce a dummy variable $vv$ such that $vv(i) = 1$ if $i \in S_v$ and $vv(i) = 0$, otherwise. Obviously, the sum of $vv$ one-zero values within $S_k$ is equal to the size of the intersection of $S_k$ and $S_v$, $n_{kv} = |S_k \cap S_v|$, and the average of $vv$ within $S_k$ is

\[
    c_{vv} = \frac{n_{kv}}{n_k} = p(v | k)
\]

(3.4)

the conditional probability of $v$ at $S_k$. Therefore, given a nominal feature with categories $v$, it is to be enveloped into a set of corresponding dummy variables, so that the part of the center of cluster $S_k$ corresponding to the feature recoded as the set of category-related dummy variables is but a vector of conditional probabilities of the $v$ categories. This is exactly the characteristic of a nominal feature which is used by those who treat the numerical and categorical parts of data separately. Equation 3.4 leads us therefore to recommend, at clustering with $k$-means, to recode the categorical part of the data into the dummy variable format and process the mixed data in such a way as this is just a numeric data.

### 3.1.2.1.2 Clusters Set

Given a set of centers $c_k$, $k = 1, \ldots, K$, the optimal with respect to Equation 3.2 clusters $S_k$ are determined according to the so-called minimum distance rule: each entity $x_i$ is assigned to its nearest center $c_k$ so that $S_k = \{i : d(x_i, c_k) = \min_{l} d(x_l, c_l)\}$. When two or more of the distances $d(x_i, c_k)$, $k = 1, \ldots, K$, coincide, the assignment is done arbitrarily among the nearest candidates, for example, to that with the minimum number of elements. The cluster set process for an $x_i = p2$ is illustrated in Figure 3.2.

The minimum distance rule is popular in data analysis and can be found in many approaches such as Voronoi diagrams and vector learning quantization.

Consider, for example, pair $(S, c)$ where $S_1 = \{p1, p2, p3, p4\}$ and $S_2 = \{p5, p6, p7\}$ and $c_1 = (1, 1), c_2 = (2, 2)$.

Table 3.2, top, presents the squared Euclidean distances to $c_1$ and $c_2$ from all the seven entities. The criterion $W(S, c)$ value is therefore the summary distance of $p1 - p4$ to $c1$, $1.00 + 1.69 + 4.09 + 2.26 = 9.04$, and $p5 - p7$ to $c2$, $1.22 + 0.25 + 0.61 = 2.08$, so that $W(S, c) = 9.04 + 2.08 = 11.12$. Let us find the optimal centers at the given $S$ according to the centers set procedure, see the means $c_1'$ and $c_2'$ in the bottom part of Table 3.2.
Point $p_2$ is assigned to $c_1$ because it is nearer than $c_2$.

**TABLE 3.2**

Squared Euclidean Distances of the Seven Two-Dimensional Points from Centers

<table>
<thead>
<tr>
<th>Center</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$p_3$</th>
<th>$p_4$</th>
<th>$p_5$</th>
<th>$p_6$</th>
<th>$p_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1 = (1, 1)$</td>
<td>1.00</td>
<td>1.69</td>
<td>4.09</td>
<td>2.26</td>
<td>1.22</td>
<td>1.25</td>
<td>2.41</td>
</tr>
<tr>
<td>$c_2 = (2, 2)$</td>
<td>1.00</td>
<td>0.29</td>
<td>1.49</td>
<td>1.06</td>
<td>1.22</td>
<td>0.25</td>
<td>0.61</td>
</tr>
<tr>
<td>$c_1' = (2.00, 1.65)$</td>
<td>1.12</td>
<td>0.55</td>
<td>1.12</td>
<td>0.55</td>
<td>1.41</td>
<td>0.02</td>
<td>0.31</td>
</tr>
<tr>
<td>$c_2' = (1.80, 1.67)$</td>
<td>0.75</td>
<td>0.37</td>
<td>1.57</td>
<td>0.81</td>
<td>1.00</td>
<td>0.07</td>
<td>0.56</td>
</tr>
</tbody>
</table>

With these centers, the summary distance is $W(S, c') = 1.12 + 0.55 + 1.12 + 0.55 + 1.00 + 0.07 + 0.56 = 4.97$, more than a 50% decrease. Similarly, given $c_1$ and $c_2$, let us set clusters according to the minimum distance rule applied to the distances in the upper part of Table 3.2. Rather unexpectedly, the distances from $c_2$ to each of the seven entities are less than or equal to those from $c_1$ – which happens because $c_1$ is rather far away from the entities. Such an effect can generate a “degenerate clustering case” when one or more clusters get empty because of “wrong” centroids. Yet, in this case, we can define $S'_2$ to consist of all the entities that are nearer to $c_2$ than to $c_1$, $S'_2 = \{p_2, p_3, p_4, p_6, p_7\}$ and $S'_1 = \{p_1, p_5\}$. Therefore, the summary distance in this case will be $W(S', c) = 1.00 + 0.29 + 1.49 + 1.06 + 1.22 + 0.25 + 0.61 = 5.92$ which is not as small as the previous value, yet is much smaller than the original $W(S, c) = 11.12$. The clusterings $(S, c')$ and $(S', c)$ can be further improved with moves to a better partition or centers set. The best possible criterion value will be achieved at the partition with clusters $\{p_1, p_2, p_5\}$ and $\{p_3, p_4, p_6, p_7\}$. It is not difficult to see that, at this partition and the optimal centers $g_1 = (1.13, 2.10)$ and $g_2 = (2.50, 1.32)$, $W(S, c) = 0.82$. No better 2-cluster partition of $I$ ever exists.

The optimal settings can be used for iterative improvement of a given partition, or set of centers, or both. By specifying a starting configuration, the optimal settings constitute what is called the batch version of the k-means algorithm. Most formulations of k-means start by initializing with specifying both $K$ and initial centers $c_1, \ldots, c_K$:
Batch k-means

Input: $K$ and $c_1, \ldots, c_K$.

1. **Clusters update**: Given $K$ centers, find clusters using **Clusters set** above.
2. **Centers update**: Given clusters, find cluster means using **Centers set** above.
3. **Stopping test**: If the set of found means do not coincide with the set of centers, set the means as new centers and go back to 1.

Output: Both centroids and clusters.

Because of the optimality of $S_k$ at each step 1 and $c_k$, at each step 2, the criterion may only decrease from step to step, so that no loop in the process may occur. Since the number of partitions on $I$ is finite, this warrants the process to converge starting from any initial centers (seeds). The number of steps usually is rather small, in single or double digits. To avoid the degenerate cluster case of empty clusters, initial centers are set to coincide with $K$ observed data points.

### 3.1.2.2 Approximation Framework

This focuses on the transformation of the dataset that is undergoing as a result of the clustering process. Indeed, the k-means clustering goal is to simplify data by representing them with $K$ centroids rather than the original $n$ entities. Therefore, k-means clustering can be considered as a device to change the original $n \times m$ data matrix $X$ with a simplified matrix at which each row $x_i$ is substituted by row $c_k$ where $c_k$ is the center of cluster $S_k$ containing $i$.

Returning to our illustrative dataset of seven entities in Table 3.1 and clustering $(S, c)$ where $S_1 = \{p_1, p_2, p_3, p_4\}$, $S_2 = \{p_5, p_6, p_7\}$ and $c_1 = (1, 1)$, $c_2 = (2, 2)$, we can counterpose the original and simplified matrices in the following matrix equation:

\[
\begin{bmatrix}
1.0 & 2.0 \\
1.5 & 2.2 \\
3.0 & 1.3 \\
2.5 & 1.1 \\
0.9 & 2.1 \\
2.0 & 1.5 \\
2.5 & 1.4
\end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1.0 \\
0.5 & 1.2 \\
2.0 & 0.3 \\
1.5 & 0.1 \\
-1.1 & 0.1 \\
-0.5 & -0.6 \end{bmatrix} + \begin{bmatrix} 0 & 1.0 \\
0.5 & 1.2 \\
2.0 & 0.3 \\
1.5 & 0.1 \\
-1.1 & 0.1 \\
-0.5 & -0.6 \end{bmatrix}
\]

The matrix on the right is the difference between the original matrix $X$ and the cluster-expressing matrix $Y$, which shows how well the clustering fits the data: the smaller are the differences, the better the fit. The goodness of fit can be measured by the sum of squares of the differences. There should be no wonder that the sum of square differences here is the same as the sum of Euclidean squared distances in Equation 3.2, 11.12.

This can be put formally, for the general case, by introducing $K \times m$ matrix $C = (c_{kh})$ whose rows are cluster centers $c_k$, $k = 1, \ldots, K$, and binary $n \times K$ cluster membership matrix $Z = (z_{ik})$ where $z_{ik} = 1$ if $i \in S_k$ and $z_{ik} = 0$, otherwise. Then the general formula for the matrix equation above is:

\[
X = ZC + E \tag{3.5}
\]

where $E$ is the matrix of differences $e_{ih} = x_{ih} - \sum_{k=1}^{K} z_{ik} c_{kh}$.
The k-means square-error criterion, in this framework, is the sum of squared differences, \( \| E \|^2 = \sum_{i=1}^{l} \sum_{h=1}^{m} e_{ih}^2 \) which is, obviously, equal to \( W(S, c) \). Why then put old wine into new bottles?

Well, the approximation perspective allows to see a wider picture of clustering, that of fitting a mathematical model of the data. For example, model in Equation 3.5 much resembles the celebrated singular-value decomposition (SVD) of matrix \( X \), except that \( Z \) here must be a binary partition membership matrix, which is related to both principal component analysis and spectral properties of related similarity matrices \( XX^T \) and \( X^T X \). This leads to a number of ideas for tackling the problem of minimization of the k-means criterion in the spectral analysis and similar frameworks. Moreover, for any partition membership matrix \( C \), if \( C \) is filled in with the within-cluster means, the Equation 3.5 leads to a Pythagorean decomposition of the quadratic data scatter,

\[
\| X \|^2 = B(S, c) + \| E \|^2 \tag{3.6}
\]

where

\[
B(S, c) = \sum_{k=1}^{K} \sum_{m} n_k c_{kh}^2 \tag{3.7}
\]

and \( \| X \|^2 = \sum_{i=1}^{l} \sum_{h=1}^{m} x_{ih}^2 \), the so-called data scatter.

Criterion \( B(S, c) \) is the part of the data scatter taken into account by the clustering \( (S, c) \). Therefore, data scatter \( \| X \|^2 \) is decomposed in two parts: the part \( B(S, c) \) explained by the cluster structure \( (S, c) \), and the unexplained part \( \| E \|^2 = W(S, c) \). The larger the explained part, the better the match between clustering \( (S, c) \) and data. The decomposition Equation 3.6 allows to score the proportion of the data scatter taken into account by the clustering \( (S, c) \), \( R^2 = B(S, c)/\| X \|^2 \), that is an index akin to the determinacy coefficient in regression analysis. It can be used for deriving different algorithms for fitting the approximation criterion of k-means.

### 3.1.2.3 Probabilistic Modeling Framework

According to this approach, each of the yet unknown clusters \( k \) is modeled by a density function \( f(x; a_k) \) which represents a family of density functions over \( x \) defined up to a parameter vector \( a_k \). A one-dimensional density function \( f(x) \), for any small \( dx > 0 \), assigns probability \( f(x)dx \) to the interval between \( x \) and \( x + dx \); multidimensional density functions have similar interpretation.

Usually, the density \( f(x; a_k) \) is considered unimodal (the mode corresponding to a cluster center), such as the Gaussian density function defined by its mean vector \( \mu_k \) and covariance matrix \( \Sigma_k \):

\[
f(x; a_k) = \left(2^m \pi^m |\Sigma_k|\right)^{-1/2} \exp\{- (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)/2 \} \tag{3.8}
\]

The shape of Gaussian clusters is ellipsoidal because any surface at which \( f(x; a_k) \) Equation 3.8 is constant satisfies equation \( (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) = const \) defining an ellipsoid. The mean vector \( \mu_k \) specifies the \( k \)-th cluster’s location; the covariance matrix, its spread.

The mixture of distribution approach makes a flexible model-based framework for clustering described at length in Chapter 3.1. Here we consider only an aspect related to the k-means criterion.
According to the mixture of distributions clustering model, the row points \( x_1, \ldots, x_n \) are considered an independent random sample of \( m \)-dimensional observations from a population with density function \( f(x) \) which is a mixture of individual cluster density functions \( f(x; \mu_k, \Sigma_k) (k = 1, \ldots, K) \) so that \( f(x) = \sum_{k=1}^{K} p_k f(x; \mu_k, \Sigma_k) \) where \( p_k > 0 \) are the cluster probabilities, \( \sum_k p_k = 1 \).

To estimate the individual cluster parameters, the maximum likelihood principle applies: those events that have really occurred are those that are most likely. Therefore, the estimates must maximize the logarithm of the likelihood of the observed data under the assumption that the data come from a mixture of distributions. There can be several different formulations of the principle. Let us assume what is referred to as a fixed partition model at which the observations come indeed from individual clusters so that among unknowns is a cluster partition \( S = \{S_1, S_2, \ldots, S_K\} \) for \( i \in S_k \), \( x_i \) comes from the individual distribution \( f(x; a_k) \). Then the likelihood of the sample, up to a power of the constant \( 2\pi \) in Equation 3.8, is the product of individual probabilities (Celeux and Govaert 1992)

\[
L = \prod_{k=1}^{K} |\Sigma_k|^{-n_k/2} \exp \left\{ -\frac{1}{2} \sum_{i \in S_k} (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) \right\}
\]

Its logarithm is

\[
\log L = \sum_{k=1}^{K} -\frac{n_k}{2} \log |\Sigma_k| - \frac{1}{2} \sum_{k=1}^{K} \sum_{i \in S_k} (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)
\]

Assume that all covariance matrices \( \Sigma_k \) are diagonal and have the same variance value \( \sigma^2 \) on the diagonal, that is, all clusters have the same spherical distribution so that observations in each cluster \( S_k \) form approximately spherical “balls” of the same size. This is the situation of our interest because the maximum likelihood criterion here leads to \( W(S, c) \) criterion of k-means and, moreover, there is a certain homology between the EM and Batch k-means algorithms, although there are differences as well. For example, the variance \( \sigma^2 \) is to be estimated in EM, but this has nothing to do with k-means.

Indeed, under this assumption the log-likelihood function is

\[
I(\{X|\mu_k, \sigma^2, S_k\}) = -nm \log(\sigma) - \sum_{k=1}^{K} \sum_{i \in S_k} (x_i - \mu_k)^T (x_i - \mu_k)/\sigma^2
\]

This function, to be maximized, is a theoretic counterpart to k-means criterion \( W(S, \mu) \) applied to vectors \( x_i \) normalized by the standard deviation. Some may conclude from this that the k-means criterion and algorithm can be applied only in situations at which the clusters to be found are expected to be of a spherical shape and of an equal spread. Yes, indeed—but only in the case when the user believes that the data come from a random sample of a mixture of Gaussians, under the fixed partition model, and is interested in estimating parameters of the function. Then k-means is applicable only for spherical equal-sized Gaussians. Under a different hypothesis of the probabilistic distribution, they should use a different method. However, if the user is not interested in the distribution, and wants just to find a clustering \( (S, c) \) minimizing the criterion \( W(S, c) \) in Equation 3.2, neither the principle of maximum likelihood nor the constraints on the shape and size
of clusters are relevant anymore. Moreover, some of the conclusions can be wrong because
they are derived from a model that might be at odds with data. Take for example the advice
of application of “z-scoring” transformation of features. This comes from the assumption of
equal variances of the variables, which can be satisfied only if the features are normalized
by their standard deviations. However, this normalization gives an unjustified advantage
to single-modal features against those multimodal ones, which is counterintuitive at clus-
tering. The experimental findings in Chiang and Mirkin (2010) show that applying k-means
is adequate at different cluster spreads and elongated, not necessarily spherical, shapes.

The three different frameworks for k-means may be viewed as corresponding to different
levels of domain knowledge by the user and data collectors. When the knowledge is very
poor, the features are superficial, and measurement methods are poorly defined, it is the
naive approach, which is convenient in such a situation. As domain knowledge becomes
stronger, more consistent, measurements are more precise and reproducible, this is a stage
for the approximation approach so that characteristics such as contribution of clustering
or feature to clustering become useful. The probabilistic modeling for k-means should be
used only when there is a perspective of equally sized spherical clusters to be recovered as
a mixture of distributions. Otherwise, more subtle criteria involved in the EM-method for
maximization the likelihood criterion are utilized (see details in Chapter 3.1).

3.2 Equivalent Reformulations of k-Means Criterion

The k-means criterion admits a number of reformulations, which give different per-
spectives leading to different approaches to optimization of the criterion. It is ironic
that, although known for some time already, these have not been picked up by the
research community as a system. Even those that have been explored, like spectral clus-
tering or cosine-based k-means, are developed as purely heuristic, unrelated to k-means,
approaches.

The explained part of the data scatter $B(S, c)$ in Equation 3.7 complements the k-means
criterion to a constant, the data scatter. Therefore, a maximum of $B(S, c)$ corresponds to
a minimum of $W(S, c)$. The following sections present four different reformulations of
$B(S, c)$; when $c_k$ is computed as the mean points in $S_k$, then these criteria are equivalent
to the square-error clustering criterion $W(S, c)$ with the only difference that they are to be
maximized rather than minimized. The fact that all four are different expressions for the
same $B(S, c)$ in Equation 3.7 can be easily proven with little algebraic transformations.

3.2.1 Anomalous Clustering Criterion

Consider

$$B(S, c) = \sum_{k=1}^{K} n_k d(0, c_k)$$  (3.10)

where $n_k = |S_k|$ is the number of entities in $S_k$; 0, all-zero vector, the space origin; and $d$
the squared Euclidean distance.

Indeed, to maximize Equation 3.10, the clusters should be as far away from 0 as possi-
ble. Therefore, when the origin is shifted into a reference point such as the grand mean,
the criterion means that clusters should be anomalous indeed—as far away from the “norm,” expressed in 0, as possible. An algorithm for finding anomalous clusters one-by-one according to Equation 3.10 was developed in Mirkin (1996), see also Chiang and Mirkin (2010), Mirkin (2012); no algorithm for simultaneously finding anomalous clusters has been developed so far.

3.2.2 Inner-Product k-Means Criterion

Another reformulation:

\[ B(S, c) = \sum_{k=1}^{K} \sum_{i \in S_k} \langle x_i, c_k \rangle \]  

(3.11)

Equation 3.11 is similar to that of \( W(S, c) \) itself, except that the relation between \( x_i \) and \( c_k \) is scored with the inner product here, not the distance. Therefore, Equation 3.11 is to be maximized rather than minimized. Note, however, that the distance-based criterion does not depend on the choice of the space origin, whereas the inner product-based criterion seemingly depends on that. Yet it is easy to derive that the value of Equation 3.11 changes just by a constant under any shift of the origin. There is a claim in the literature that the inner product is beneficial at higher dimensions of the feature set if the data is prenormalized in such a way that the rows, corresponding to entities, are normed so that each has its Euclidean norm equal to unity (note the change of the emphasis in normalization here, from features to entities), for example, France et al. (2012); the distances are more or less useless in the higher dimensions.

3.2.3 Kernel-Wise Criterion of Maximization of the Semi-Averaged Internal Similarities

Centers vanish from the following equation:

\[ B(S, c) = \sum_{k=1}^{K} \sum_{i,j \in S_k} \langle x_i, x_j \rangle / n_k \]  

(3.12)

Equation 3.12 expresses the criterion in terms of similarities, the inner products \( \langle x_i, x_j \rangle \). As now is well recognized, a criterion in which the data are present only through the inner products can be subject to the so-called “kernel trick.” The trick is to change the inner products \( \langle x_i, x_j \rangle \) for values of a so-called positive semidefinite kernel function \( K(x_i, x_j) \) under the assumption that the function expresses the inner product in a higher dimension space where the original features have been mapped, possibly, in a nonlinear way. Among the popular kernel functions is the so-called Gaussian kernel \( K(x_i, x_j) = \exp(-d(x_i, x_j)/s) \), where \( d \) is the squared Euclidean distance and \( s \) a parameter. The criterion Equation 3.12 is frequently used for clustering similarity data, with no relation to k-means. It should be mentioned that the kernel trick can be applied to a modified form of the criterion \( W(S, c) \) itself (Girolami 2002).

3.2.4 Spectral Rayleigh Quotient Formulation

The formulation here is:

\[ B(S, c) = \sum_{k=1}^{K} \frac{z_k^T A z_k}{z_k^T z_k} \]  

(3.13)
where \( A = XX^T \) and \( z_k = (z_{ik}) \) is the binary membership vector so that \( z_{ik} = 1 \) if \( i \in S_k \) and \( z_{ik} = 0 \), otherwise.

Equation 3.13 is but a matrix form of Equation 3.12, yet it has a special meaning because of its relation to the theory of eigenvalues and eigenvectors of a square matrix. Indeed, the ratio \( r(z) = z^T A z / z^T z \) is well known as the so-called Rayleigh quotient. Given a symmetric matrix \( A \), the Rayleigh quotient’s maximum with respect to arbitrary vectors \( z \) is equal to the maximum eigenvalue of \( A \) and it is reached at the corresponding eigenvector. Moreover, the maximum of criterion Equation 3.13 with respect to arbitrary \( z_1, z_2, \ldots, z_K \) is reached at the eigenvectors of \( A \) corresponding to its maximum \( K \) eigenvalues. Recall that here \( z_k \) must be binary cluster membership vectors. Therefore, the spectral approach is applicable. According to this approach, the binarity constraints are relaxed, so that the eigenvectors form a solution of the relaxed problem. The clusters are to be found as fragments of the natural orderings of the entity set according to the descending ordering of eigenvectors. The spectral form of the clustering criterion Equation 3.13 did not attract much interest before recently. We can indicate only paper (Zha et al. 2001) giving a node to this. An effective heuristic formulation involving eigenvectors of a Laplacian transformation of a square kernel similarity matrix proposed in Ng et al. (2002) should be counted as a step in the same direction. Yet recently, Kumar and Kannan (2010) proposed an interesting condition of separation between the centers which can lead to useful theoretic results on the relation between k-means and SVD.

3.3 Challenges for k-Means Criterion and Algorithm

3.3.1 Properties of the Criterion

There are a couple of geometric properties of the k-means criterion, which are quite easy to prove. The other properties mentioned below concern its ability to recover the clusters rather than the data.

3.3.1.1 Convexity of Cluster Domains

Each k-means cluster is located in a convex polytope linearly separated from polytopes of the other clusters.

This follows from the minimum distance rule. The \( k \)-th cluster lies in a halfspace \( \Omega_l \) separated from any \( l \)-th cluster by the hyperplane orthogonally crossing the interval between \( c_k \) and \( c_l \) in its middle point, and therefore, in the intersection of these subspaces \( \Omega_l \) over all \( l \neq k \).

3.3.1.2 Monotonicity of the Minimum with Respect to the Number of Clusters \( K \)

Given a dataset, denote by \( w_K \) the minimum value of \( W(S, c) \) with respect to all possible \( K \)-cluster partitions; then \( w_{K+1} \leq w_K \) for all \( K = 1, 2, \ldots, n - 1 \).

3.3.1.3 Dependence on the Feature Measurement Scales

Since k-means heavily relies on the squared Euclidean distance, it is highly dependent on the measurement scales of the variables. If, say, the scale of a feature changes 10 times, the
FIGURE 3.3
Two-modal variable on the right leads to two natural clusters, yet after normalization by the standard deviations it contributes less than the one-modal variable on the left.

Contribution of the feature to the squared distance changes 100 times. This leads to the idea of balancing the feature contributions either preliminarily or during the process of iterations in k-means. An approach to latter will be described in Section 3.3.6. The preliminary balancing usually is done with the so-called normalization of features so that afterwards their contributions become comparable. The most straightforward and frequently advised normalization is performed by dividing features by their standard deviations, so that all the normalized features have the same standard deviation.

Although quite sound in many cases, this normalization should not be applied when clustering data (Mirkin 2012). Indeed, the contribution of a feature with a multimodal distribution would relatively decrease under this normalization because the standard deviation of such a feature is greater than that of a unimodal feature (see Figure 3.3). This would be counterintuitive, because multimodal features are more useful at clustering than those unimodal. The cause is that the value of standard deviation depends both on the scale and shape of the distribution. Normalization over feature ranges in some cases would better suit the balancing purposes. Indeed, experimental evidence supports the use of feature ranges rather than standard deviations for normalization (Milligan and Cooper 1988, Steinley and Brusco 2007).

3.3.1.4 Propensity to Balance the Sizes of Clusters

Consider one-dimensional points A, B, C with coordinates 1, 2, 6, respectively. Assume that 50 entities are put at A, the other 50 are put at B, and only one entity is in C. Take \( K = 2 \), and see which split minimizes the k-means criterion: A and B versus C or A versus B and C. In the former case, A and B are in one cluster so that the center of the 100-point merged cluster will be at point 1.5. Therefore, the squared Euclidean distance from any point in A or B to the center is 0.25, the square of the difference 0.5. The summary distance from all 100 entities in A merged with B to the center is \( 100 \times 0.25 = 25 \), which is exactly the value of k-means criterion for the partition \( \{A \cup B, C\} \). The other split puts together 51 entities in B and C, with their center being at \( CB = (50 \times 2 + 6)/51 = 2.0784 \). The summary squared Euclidean distance from this point to the 51 entities in the merged cluster is equal to \( d = 50 \times 0.784^2 + 3.9216^2 = 15.6863 \). This clearly shows that k-means favors the latter partition, \( \{A, B \cup C\} \) over the former one, which contradicts the intuition that the nearest points A and B should be put together to make a good 2-clustering.

3.3.1.5 Square Error Versus Consensus

The fact above that the square-error criterion can be counterintuitive made the current author to look at consensus clustering (see Kuncheva and Vetrov (2005)) as a potential supplement to the square-error criterion. In Shestakov and Mirkin (2012), Gaussian cluster sets of different degrees of intermix were generated; given such a dataset, k-means starting
from random \( K \) entities as centers was run and then postprocessed with two options: (i) BSE: choosing the best of obtained clusterings according to the k-means square-error criterion \( W(S, c) \), (ii) ECC: finding an ensemble consensus clustering according to the algorithm described in Mirkin (2012). The ensemble consensus clustering criterion operates with incidence matrices of partitions: given a partition \( S \) on \( I \), its incidence matrix \( Z = (z_{ik}) \) has its entries \( z_{ik} = 1 \) if \( i \) belongs to \( k \)th class of \( S \) and \( z_{ik} = 0 \), otherwise. Given a number of partitions on \( I \), an ECC partition is defined as such a partition \( S \) on \( I \) that projections of the incidence matrices of all given partitions on the linear subspace \( L(Z) \) spanning its incidence matrix \( Z \) are as close to themselves as possible. When the ECC goodness of fit is the summary squared error, this criterion is equivalent to the kernel-wise criterion Equation 3.12 in which the inner product \( \langle x_i, x_j \rangle \) is substituted by the consensus matrix value \( a_{ij} \) (Mirkin 2012). The consensus matrix value \( a_{ij} \) is the number of those among the given partitions in which \( i \) and \( j \) belong to the same cluster. The ECC algorithm outperformed its competition in a series of experiments (Shestakov and Mirkin 2012). It appears, on average, that the ECC clusterings are much closer to those generated than the corresponding BSE clusterings (see Shestakov and Mirkin (2012) and Section 5.4.2 in Mirkin (2012)).

### 3.3.2 Different Clustering Structures

Method k-means partitions the dataset into \( K \) nonoverlapping clusters. The clusters should be nonempty, although, if an initial center is far away from both the other centers and the entity points, it might remain empty since no entity is nearer to it than to other centers. To warrant that this may not happen, it is usually sufficient to put initial centers in data points. In some problems, partition can be considered an overly rigid structure: it requires each entity point to belong to one and only one cluster. There can be situations when entities could belong to more than one cluster or to none of them. There are at least four types of structures that can deal with the multiple memberships:

1. Mixture of distributions mentioned in Section 3.1.2—this will be extensively covered in Chapter 3.1.
2. Fuzzy cluster partition: the full membership value of an entity can be divided among several clusters so that each cluster gets a corresponding share of the membership and the shares total to unity. An extension of k-means to this will be covered in Chapter 5.5.
3. Overlapping clusters: clusters may be crisp and still may have nonempty intersection. An extension of the model Equation 3.5 to that case is considered in Depril et al. (2008).
4. Hierarchical clusters: clusters may be crisp and form a hierarchical tree, so that each cluster, which is a proper subset of all the entity set, is part of a larger cluster in the tree. Then k-means can be a viable tool for building such a hierarchy divisively with what is called Bisecting k-means (Mirkin 1996, 2012, Steinbach et al. 2000). This subject is out of scope of this chapter.

The case of incomplete clustering can be covered with the model in Equation 3.5 by using an incomplete matrix \( Z \), similarly to the case of Anomalous Pattern (AP) clustering in which \( Z \) consists of just one column (see Section 3.3.3).

Another type of different clustering structures emerges when a cluster center is not just a point in the feature space but forms a structure of its own (regression-wise clustering, PCA-wise clustering, etc.). These are out of scope of this chapter too.
3.3.3 Initialization of k-Means

To initialize k-means, one needs to specify:

1. The number of clusters, \( K \)
2. Initial centers (sometimes referred to as seeds), \( c_1, c_2, \ldots, c_K \)

This generated scores of different proposals that essentially boil down to two generic ideas:
(1) investigate the data structure (Steinley and Brusco 2007), (2) use multiple data of the process under consideration, including domain knowledge, as well as a combination of these.

The data structure can be effectively investigated at all the stages of clustering process: [A] preclustering, [B] while-clustering, and [C] postclustering.

Most efforts by researchers in finding a “right” number of clusters \( K \) have been devoted to option [C], postprocessing, in the format of multiple runs of k-means starting from random initial centers with a follow-up analysis of relative changes in the criterion values when moving from \( K \) to \( K = K + 1 \) or \( K - 1 \); see more on this in Chapter 6.1.

With option [A], preprocessing, a viable idea is to locate “anomalous” patterns in the dataset and put initial centers in those. This concurs with the equivalent k-means clustering criterion in Equation 3.10. A relatively simple method starting from the two most distant entities and adding one by one further farthest entities does not always work, probably because it can dwell on outliers which are not characteristic for the structure (Mirkin 2012).

A good method should warrant not only an “anomalous” point but a pattern, that is, a “dense” set of points, around it as well.

Such is the AP method that finds APs one by one by locally maximizing criterion
\[
 r(z) = z^T A z / z^T z = n(S_1) a(S_1),
\]
where \( S_1 \) is the sought cluster, \( a(S_1) \), the average within-\( S_1 \) similarity \( a_{ij} \), \( n(S_1) \), the number of entities in \( S_1 \), and
\[
 A = (a_{ij}) = (< x_i, x_j >),
\]
the matrix of similarities (Chiang and Mirkin 2010, Mirkin 2012). It puts the space origin into the grand mean of the dataset or any other “reference” points, and the initial center of AP into the point that is farthest away from the origin. Then it applies iterations of 2-Means to the origin and the center, at which the origin is never moved. The stabilized cluster around the center is the AP sought (see Figure 3.4). After it is removed from the dataset, the next AP is found at the remainder, etc. When no entities remain in the remainder, the singleton patterns are removed, and those larger than that are used to both set \( K \) and initialize k-means (ik-means). In a series of experiments with a wide set of cluster structures, this method has appeared superior over many other methods in terms of the cluster recovery, although the number of clusters it generates is overly high sometimes (Chiang and Mirkin 2010).

A popular procedure Build (Kaufman and Rousseeuw 1990) for selecting initial seeds proceeds in a manner resembling that of the iterated AP. The seeds here must belong to the dataset, the number \( K \) is prespecified, and the center of gravity is taken as the first seed.

Option [B], while-clustering, has received a boost recently in (Tasoulis et al. 2010) and Kovaleva and Mirkin (2013). It appears, a process of divisive clustering can be stopped at the “right” clusters by applying an additional criterion of stopping the division process if a (proportion of the) unidimensional projection(s) of the within-cluster density function has no visible minima.

Consider the second generic idea, the usage of multiple data of the process under consideration, including domain knowledge. This aspect was neglected previously, probably, because of lacking multiple data in most cases. Yet the availability of multiple data suggests a formalized view of a most important clustering criterion, quite well known
to all practitioners in the field—consistency between clusters and other aspects of the phenomenon in question. Long considered as a purely intuitive and thus thoroughly unscientific matter, this emerges currently as a powerful device, first of all in bioinformatics studies. Dotan-Cohen et al. (2009) and Freudenberg et al. (2009) show how the knowledge of biomolecular functions embedded in the so-called Gene Ontology can be used to cut functionally meaningful clusters. Mirkin et al. (2010) derives the number and location of clusters by combining three different sources of information on genes: protein sequence, gene arrangement, and reconstructed evolutionary histories of genes. The more knowledge of different aspects of real-world phenomena emerges, the greater importance of the consistency criterion in deciding of the right number of clusters.

3.3.4 Getting a Deeper Minimum

Although some recent mathematics studies suggest that the number of iterations in a run of k-means can be very large (Vattani 2009), this is not what happens in practical computations. Typically, the number of iterations is small and the final centers are not far away from the initial ones. That is, the stationary “minima” of $W(S,c)$ achieved with k-means are not deep.

There are two different views on this. One relates to the perspective in which clustering is just a tool for building classifications. In this perspective, k-means expresses stages of typology making, at which the criterion is considered not as something that must be minimized at any cost but rather a search direction. The centers should express prototypes that are to be domain knowledge driven. k-means is but a tool to adjust them to real data so that the closer the final prototypes to the initial ones, the better the knowledge reflects the reality. What is important in this perspective, though, is defining an appropriate, rather than random, initial setting.

The second perspective is not so much concerned with typology making but rather with minimization of the criterion. In this perspective, the goal is to globally minimize the criterion. This is a hard problem (Drineas et al. 2004). This is why various heuristics are used to get a better algorithm than the batch k-means. First of all, attempts should be mentioned that change the structure of neighborhoods:
1. Incremental version of k-means by moving one entity at a time (MacQueen 1967).
2. Variable neighborhood versions such as J-Means (Hansen and Mladenovic 2001).
3. Nature inspired approaches:
   a. Genetic algorithms in which partitions are taken as the “chromosomes” (Chang et al. 2009, Krishna and Murty 1999, Lu 2004);
   b. Evolution and Differential Evolution algorithms (Naldi et al. 2011, Paterlini and Krink 2006) in which the population is represented by cluster centers rather than partitions, which allows for smoother changes in it through additions of small random changes
   c. Particle Swarm Optimization algorithms (Lam et al. 2013, Van der Merwe and Engelbrecht 2003) in which the cluster centers are subject to moving with taking into account the best places the population had encountered (as bees supposedly do)
   d. Other population behavior modeling like ant-colony optimization (Saatchi and Hung 2005, Tsai et al. 2011).

Xavier and Xavier (2011) proposes considering a smoothed version of criterion \( W(S,c) \) which allows to lessen the number of local minima in the function. It also makes a useful criterion-based distinction between “boundary” and “core” regions; the latter being rather stable in the iterative process.

### 3.3.5 Interpretation Aids

Two conventional tools for interpreting k-means clustering results, partition \( S \) and cluster centroids \( c = \{c_1, \ldots, c_K\} \), are

1. Analysis of cluster centroids \( c_k \), or “representative” entities
2. Analysis of bivariate distributions between cluster partition \( S = \{S_k\} \) and categorical features related to the dataset

In fact, (2) can be considered part of (1) since, with the zero-one dummy coding of categories, cross-classification frequencies are but cluster centroids.

The decomposition Equation 3.6 leads to a number of less conventional aids based on contributions of cluster-feature pairs \( B_{kh} = n_k c_{kh}^2 \) to the data scatter (Mirkin 2012). Such a value has a simple intuitive meaning if the features have been preliminarily centered. Indeed, in this case, the value \( n_k c_{kh}^2 \) is proportional to the squared difference between the grand mean and within-cluster mean of variable \( h \): the further away the within cluster mean from the grand mean, the greater the contribution. Curiously, for centered dummy variables representing categories, these sum to conventional cross-classification association measures such as Pearson chi-squared, depending on the dummies’ normalization (Mirkin 2012).

Somewhat less intuitive is the choice of a cluster representative, a “prototypic” entity, based on the decomposition Equation 3.6. In empirical domains, such as mineralogy or oenology, indication of a “prototypic” entity makes much sense. The contribution of an entity \( i \) to the explained part of the data scatter, \( B(S,c) \) Equation 3.7, appears to be equal to \((x_i, c_k)\) where \( c_k \) is the center of cluster containing \( i \). Therefore, the prototype should maximize the inner product, rather than minimize the distance, to its cluster’s center. In the
case when the data have been centered, this choice better follows tendencies represented in \( c_k \) versus the grand mean than the conventional choice according to the distance.

### 3.3.6 Feature Weighting, Three-Stage k-Means, and Minkowski Metric

Drawbacks of k-means method have been mentioned above such as the lack of advice on the choice of the number of clusters and location of the initial centers. The latter is aggravated by the fact that k-means usually reaches very superficial minima of the criterion. One more drawback of the method is that k-means method cannot distinguish noise, or irrelevant, features from those useful ones and, therefore, is defenseless against them. To address this, a three-stage version of k-means involving feature weighting was proposed in Makarenkov and Legendre (2001) and further developed in (Huang et al. 2008). These have been integrated in a three-stage Minkowski k-means approach (Amorim and Mirkin 2012). The approach adjusts the feature scales in such a way that those supporting the found partition get greater weights over those less relevant ones. Here is an outline of the approach.

In data analysis, \( p \)-th power of Minkowski distance between vectors \( x = (x_v) \) and \( y = (y_v) \) is defined by formula

\[
d^p(x, y) = \sum_{v \in V} |x_v - y_v|^p
\]

Minkowski metric k-means clustering criterion is specified as

\[
W_p(S, c) = \sum_{k=1}^{K} \sum_{i \in S_k} d^p(x_i, c_k) = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{v \in V} |x_{iv} - c_{kv}|^p
\]

(3.14)

Obviously, at \( p = 2 \) this is the conventional k-means criterion Equation 3.2.

To formulate the Batch k-means algorithm in this case, one needs an algorithm for finding Minkowski’s center, that is, \( c \) minimizing

\[
D_p(c) = \sum_{i=1}^{n} |x_i - c|^p
\]

(3.15)

for any series of reals \( x_1, x_2, \ldots, x_n \) (see Figure 3.5).

To include the issue of feature weighting in k-means, let us reformulate the criterion as follows. Given a standardized data matrix \( X = (x_{ih}) \), where \( i \in I \) are entities and \( h \in H \)

![Graphs of Minkowski \( D_p(c) \) function for the series \( I = \{2, 2, 3, 4, 5, 1\} \) at \( p = 1.5 \), on the left, and \( p = 0.5 \), on the right.](image)
features, the nonnegative weights \( w_h \), satisfying condition \( \sum_{h \in H} w_h = 1 \), serve as rescaling factors so that the criterion Equation 3.14 is reformulated over the rescaled data \( x'_{ih} = w_h x_{ih} \) and rescaled centroids \( c'_{kh} = w_h c_{kh} \):

\[
W_p(S, c, w) = \sum_{k=1}^{K} \sum_{i \in S_k} d^p(x'_i, c'_k) = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{h \in H} w_h^p |x_{ih} - c_{kh}|^p
\]

Criterion Equation 3.16 holds to the generic form of k-means criterion, which is in stark contrast to other approaches (for a review, see (Amorim and Mirkin 2012)). The alternating minimization of the criterion Equation 3.16 over three groups of variables, corresponding to \( S, c, \) and \( w \), respectively, leads to a three-step version of k-means. The computation starts similarly to batch k-means, from \( K \) tentative centroids \( c_1, c_2, \ldots, c_K \) and initial weights all equal to each other, \( w_h = 1/|H| \).

Here is a formulation of the algorithm, referred to as MWK-Means in (Amorim and Mirkin 2012), for alternating minimization of criterion Equation 3.16 at a given \( p \):

1. Given centroids \( c_k \) and weights \( w_h \), update the cluster assignment of entities by using the minimum distance rule with distance defined as \( p \)-power of Minkowski metric \( d^p(x_i, c_k) = \sum_{h \in H} w_h^p |x_{ih} - c_{kh}|^p \);
2. Given clusters \( S_k \) and weights \( w_h \), update centroid \( c_k = (c_{kh}) \) of each cluster \( S_k \) as its Minkowski center so that, at each \( h \), \( c_{kh} \) is defined as \( w^*_h c \) where \( c \) minimizes an item in Minkowski’s distance power, \( \sum_{i=1}^{N} w_h|x_{ih} - c|^p \) according to Equation 3.15;
3. Given clusters \( S_k \) and centroids \( c_k \), update weights according to formula

\[
w_h = \frac{1}{\sum_{h' \in H} [D_{hp} / D_{h'p}]^{1/p}}
\]

where \( D_{hp} = \sum_{k=1}^{K} \sum_{i \in S_k} |x_{ih} - c_{kh}|^p \).

The formulation of the MWK-Means algorithm suggests that the feature weights can be made cluster-specific both in the criterion Equation 3.16 and the algorithm (Amorim and Mirkin 2012).

It appears that cluster recovery can improve indeed, at some datasets drastically, at an appropriate \( p \) depending on the dataset. Moreover, an appropriate version of anomalous pattern method can be used to define the number of clusters \( K \). For example, at the celebrated 150 × 4 Iris dataset, the Minkowski three-stage clustering method makes only five errors at \( p = 1.2 \)—an absolute minimum of all the supervised and unsupervised methods reported (Amorim and Mirkin 2012). The issue of choosing an appropriate value of Minkowski exponent \( p \) can be addressed by learning in a semi-supervised manner if cluster labels are supplied at about 5% of the entities (Amorim and Mirkin 2012). In fact, (Amorim and Mirkin 2012) also reports of a successful experiment in determining the right \( p \) in a nonsupervised manner.

### 3.4 Notes on Software for k-Means

Although the k-means method deserves to be coded as a stand-alone application supplied with powerful tools for getting a meaningful cluster structure by using a wide spectrum of
computation and interpretation devices, the method currently appears in a generic version in most popular computational platforms and packages. In this section, a brief description of implementations of the method will be given in computational platforms:

- Matlab (see http://www.mathworks.com/products/matlab/),
- Weka (see http://www.cs.waikato.ac.nz/ml/weka/),
- R (see http://www.r-project.org/),
- SPSS (see http://www-01.ibm.com/software/analytics/spss/).

In the end, an outline of currently available platforms for applying k-means for the analysis of “big data” will be given.

Matlab k-means clustering takes data matrix $X$ and the number of clusters $K$ as its inputs and performs a run of batch k-means clustering with the squared Euclidean distances by default. It allows using some different distances, first of all the city block and cosine, as well. Other user selected parameters include:

- An initialization method: selecting $K$ random rows from $X$, or a matrix of user-specified $K$ points in the feature space, or results of preliminary clustering of a 10% random subsample of the dataset.
- Number of runs with different initializations.
- The way of reacting to the fact that some cluster happens to get empty: this can be an “error” message or removal of empty clusters or making a singleton instead.
- The maximum number of iterations so that the computation stops upon reaching that number, 100, by default.
- A possibility of further minimization of the criterion by moving individual entities between clusters.

In Weka, k-means is implemented as SimpleKMeans command. This command automatically handles a mixture of categorical and numerical attributes. To this end, the program converts all nominal attributes into binary numeric attributes and normalizes scales of all the numeric features in the dataset to lie within the interval $[0, 1]$. When prompted, the program replaces all missing values for nominal and numeric attributes with the modes and means of features in the training data. Initialization is made randomly from among the observations. Instance weighting can be included if needed. The Weka SimpleKMeans algorithm uses Euclidean distance measure to compute distances between instances and centers. It can visualize the results with figures drawn for each cluster so that features are represented by lines radiating from the center.

In R, Project for statistical computing, k-means clustering is available in either batch version or incremental one. It works either with random $K$ observations taken as initial centers or with a user-specified set of centers. The maximum number of iterations can be specified, as well as the number of runs from random initializations. In the Weka version, $K = 1$ is allowed. If an empty cluster emerges, an “error” message appears.

The statistical package SPSS allows to use its rich system for both preprocessing of data and interpretation of results. Its implementation of k-means is similar to that in Matlab—all the parameters available in Matlab are available in SPSS, except for the way for handling missing values. There is a different system for handling missing data in SPSS. Additional features include the possibility for storing the clusters as a nominal variable, ANOVA table.
for clusters to allow to see what variables contribute most to a cluster, and the matrix of distances between cluster centers.

Currently, some efforts are devoted to parallelization of k-means computations to apply it to big data, that is, data with millions or more entities. The method is well suited for distributed computations. Say, a method in Zhao et al. (2009) is based on the idea that the dataset is partitioned into portions processed at different processors. An iteration is performed as follows: a central processor supplies a set of centers so that each processor computes distances from its portion entities to the centers and within cluster sums, after which it supplies the central processor with the sums and number of entities. Then the central processor sums all the received sums within clusters and computes the cluster centers. Further development of cloud computations as a commodity will show what versions of the algorithm will be required in the future.

Acknowledgments

This work has been supported by the Laboratory of Decision Choice and Analysis NRU HSE Moscow and, also, the Laboratory of Algorithms and Technologies for Networks Analysis NRU HSE Nizhny Novgorod (by means of RF government grant, ag. 11.G34.31.0057), as well as by grants from the Academic Fund of the NRU HSE in 2010–2013.

References


D. Depril, I. Van Mechelen, B. Mirkin 2008 Algorithms for additive clustering of rectangular data tables, Computational Statistics and Data Analysis, 52(11), 4923–4938.


Quadratic Error and k-Means


M. Steinbach, G. Karypis, V. Kumar 2000 A comparison of document clustering techniques, KDD Workshop on Text Mining.


