k-Means

Autor und Herausgeber

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland (ORCID 0000-0001-7538-0990)
Editors: Konstantina Olioumtsevits (ORCID 0000-0003-2697-8659), Ekkehard Schnoor (ORCID 0000-0001-8252-1499), Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Zusammenfassung

k-Means ist eine hartes Clustering-Methode für Datenpunkte mit numerischen Merkmalsvektoren. Jeder Datenpunkt wird genau einem von k Cluster zugewiesen, wobei jeder durch einen Clusterzentrum repräsentiert wird, nämlich dem Cluster, dessen Clusterzentrum in der euklidischer Abstand am nächsten liegt. Die Clusterzentren werden so gewählt, dass der clustering error minimiert wird, der durchschnittliche quadrierte euklidischer Abstand zwischen einem Datenpunkt und seinem nächsten Clusterzentrum. Das resultierende k-Means-Optimierungsproblem ist eine Instanz von empirische Risikominimierung (ERM) unter Verwendung einer spezifischen Wahl für die Verlustfunktion. Dieses Optimierungsproblem ist nicht-konvex und NP-schwer und wird typischerweise mit dem Lloyd's algorithm gelöst. Dieser Algorithmus besteht aus einer fixed-point iteration, die den clustering error niemals erhöht und nach endlich vielen Iterationen einen Fixpunkt erreicht. Das Ersetzen der quadrierten euklidischer Abstand durch die euklidischer Abstand selbst ergibt k-Mediane, deren Mittelungsschritt einen geometric median (GM) berechnet und die weniger empfindlich gegenüber Ausreißer sind.

Definition

Ein Foto eines Berges wird in kleine quadratische Bereiche geschnitten, und jeder Bereich wird durch zwei Zahlen beschrieben: wie grün und wie blau er im Durchschnitt ist (Abb. 1). Vegetation, Berg und Dunst sowie heller Himmel bilden drei Cluster solcher Bereiche, die Datenpunkte sind, charakterisiert durch einen Merkmalsvektor $\mathbf{x} \in \mathbb{R}^d$. k-Mittelwerte ist eine hartes Clustering-Methode für solche Datenpunkte (Hastie et al., 2009, Abschn. 14.3.6; MacQueen, 1967).

Die Idee von k-Mittelwerte ist es, jeden Datenpunkt eines Datensatz $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$ genau einem von k Cluster zuzuweisen, die mit $c = 1, \ldots, k$ indiziert sind. Jeder Cluster wird durch einen Clusterzentrum $\mathbf{w}^{(c)} \in \mathbb{R}^d$ repräsentiert, und ein Datenpunkt wird dem Cluster zugewiesen, dessen Clusterzentrum in der euklidischer Abstand am nächsten liegt. Die Qualität einer gegebenen Wahl von Clusterzentren und der resultierenden hartes Clustering wird durch den clustering error gemessen: den durchschnittlichen quadrierten euklidischer Abstand zwischen einem Datenpunkt und seinem nächsten Clus-

terzentrum,

$$\frac{1}{m}\sum_{r=1}^{m}\min_{c=1,\ldots,k}\left\|\mathbf{x}^{(r)}-\mathbf{w}^{(c)}\right\|_2^2.$$

Das k-Mittelwerte-Prinzip besteht darin, Clusterzentren zu wählen, die den clustering error minimieren,

$$\min_{\mathbf{w}^{(1)},\ldots,\mathbf{w}^{(k)}}\ \frac{1}{m}\sum_{r=1}^{m}\min_{c=1,\ldots,k}\left\|\mathbf{x}^{(r)}-\mathbf{w}^{(c)}\right\|_2^2.$$

Die innere Minimierung weist jeden Datenpunkt seinem nächsten Clusterzentrum zu, und die äußere Minimierung verschiebt die Clusterzentren (Abb. 1).

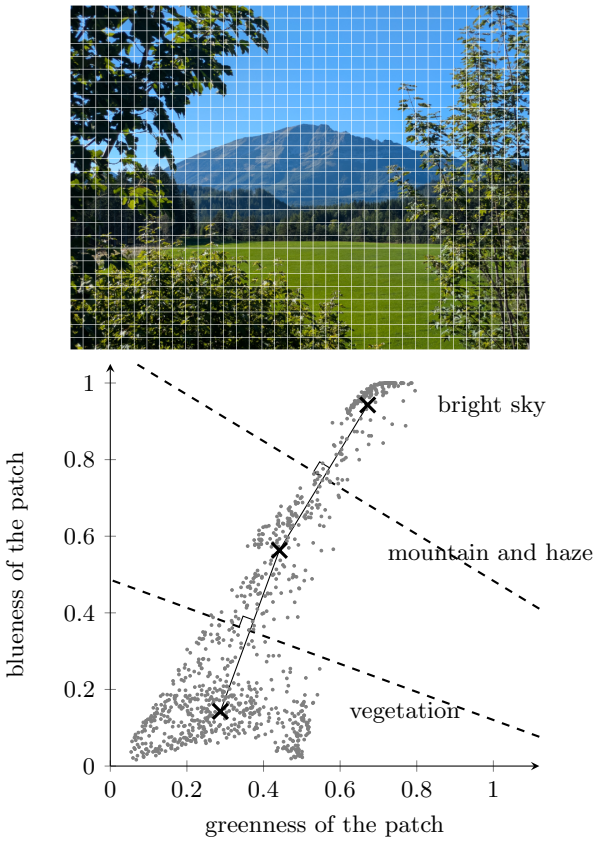


Abbildung 1: Oben: die Fotografie des Ötscher-Massivs mit dem darauf gezeichneten Patch-Gitter; jeder Patch ist ein Datenpunkt. Unten: die 972 Patches, jeder beschrieben durch seine durchschnittliche Grünheit und seine durchschnittliche Bläue, beide auf einer Skala von 0 bis 1. Die drei Kreuze sind die Clusterzentren, die von k-Mittelwerte mit $k = 3$ gefunden wurden, beschriftet an ihrer Höhe, und die gestrichelten Linien sind die Grenzen zwischen den Cluster: Jeder Patch wird dem Cluster zugewiesen, dessen Clusterzentrum am nächsten ist. Die dünnen Linien verbinden zwei Clusterzentren; jede Grenze kreuzt eine solche Linie in ihrem Mittelpunkt im rechten Winkel, markiert durch das kleine Quadrat. Daten generiert von `pythondemos/kmeans.py`

Das k-Mittelwerte-Optimierungsproblem hat keine geschlossene Lösung, und es exakt zu lösen ist NP-schwer (Mahajan et al., 2009). Die Standardmethode zur annähernden Lösung ist der Lloyd's algorithm (Lloyd, 1982), der zwischen zwei Schritten wechselt. Der Zuweisungsschritt weist jeden Datenpunkt seinem nächsten Clusterzentrum zu,

$$\widehat{c}^{(r)} := \operatorname*{arg\,min}_{c=1,\ldots,k} \left\| \mathbf{x}^{(r)} - \mathbf{w}^{(c)} \right\|_2 \quad \text{für } r = 1, \ldots, m, \tag{1}$$

und der Mittelungsschritt verschiebt jeden Clusterzentrum zum Mittelwert der ihm zugewiesenen Datenpunkte,

$$\mathbf{w}^{(c)} := \frac{1}{\left|\{r : \widehat{c}^{(r)} = c\}\right|} \sum_{r:\widehat{c}^{(r)}=c} \mathbf{x}^{(r)} \quad \text{für } c = 1, \ldots, k. \tag{2}$$

Die beiden Schritte werden wiederholt, bis sich die Zuweisung (1) nicht mehr ändert. Algorithmus 1 beschreibt die Methode vollständig.

Algorithm 1 k-Means, Lloyd's algorithm (Hastie et al., 2009, Algorithmus 14.1)

Input: Merkmalsvektoren $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$, Cluster count k, initial Clusterzentren $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(k)}$

1: **repeat**

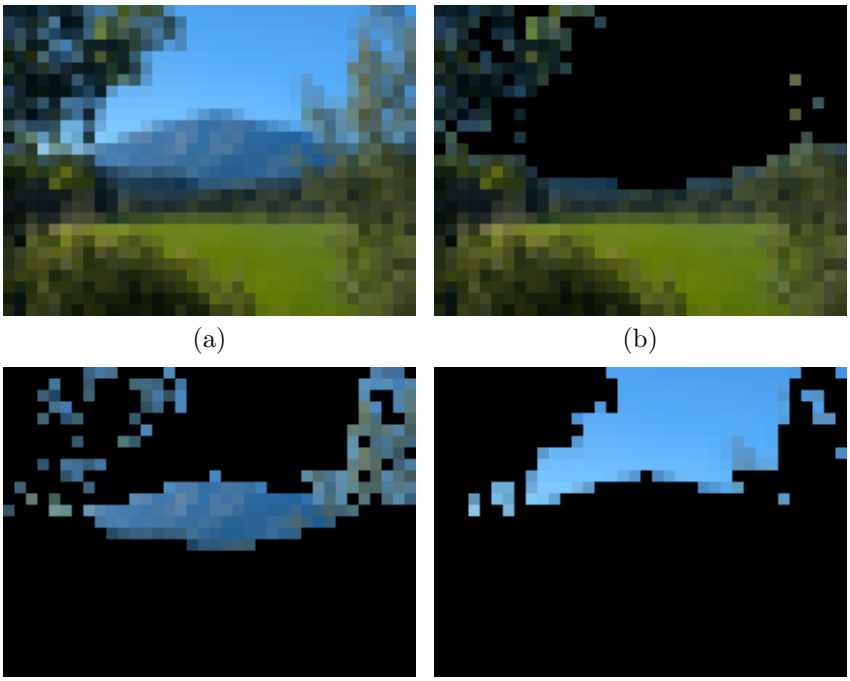
2: **assignment:** for each $r = 1, \ldots, m$, assign $\mathbf{x}^{(r)}$ to the Cluster $\arg\min_{c=1,\ldots,k} \left\|\mathbf{x}^{(r)} - \mathbf{w}^{(c)}\right\|_2$, as in (1)

3: **averaging:** for each $c = 1, \ldots, k$, set $\mathbf{w}^{(c)}$ to the Mittelwert of the Merkmalsvektoren assigned to Cluster c, as in (2)

4: **until** no assignment changes

Output: Clusterzentren $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(k)}$ and the assignment they induce

Angewendet auf die 972 Bereiche von Abb. 1 mit $k = 3$, trennt k-Mittelwerte Vegetation, Berg und Dunst sowie hellen Himmel. Jeder Bereich, den es dem blauesten Clusterzentrum zuweist, liegt in der oberen Hälfte des Fotos. Die gelernten Clusterzentren definieren eine hartes Clustering: Jeder Datenpunkt gehört zu genau einem Cluster, dem mit dem nächsten Clusterzentrum. Abb. 2 zeigt diese Zuweisung auf dem Foto selbst.



(a) (b) (c) (d)

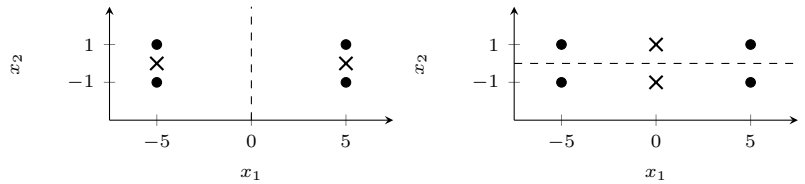
Abbildung 2: (a) Die Fotografie aus Abb. 1 in Patch-Auflösung, jeder Patch ein Datenpunkt. Dieselbe Fotografie dreimal, wobei jeweils nur die Patches beibehalten werden, die einem Cluster zugewiesen sind: Vegetation (b), Berg und Dunst (c) und heller Himmel (d). Jeder Patch ist in genau einer Kopie hell und in den anderen beiden schwarz, sodass die drei Kopien sich wieder zu (a) addieren. Daten generiert von `pythondemos/kmeans.py`

Das k-Mittelwerte-Prinzip ist eine Instanz von empirische Risikominimierung (ERM) mit einer bestimmten Wahl für den Hypothesenraum und die Verlustfunktion. Der Hypothesenraum $\mathcal{H}_k$ besteht aus allen Abbildungen $h : \mathbb{R}^d \to \mathbb{R}^d$ der stückweise konstanten Form

$$h(\mathbf{x}) = \mathbf{w}^{(\,c^\star(\mathbf{x}))}, \qquad c^\star(\mathbf{x}) = \underset{c=1,\ldots,k}{\arg\min} \left\| \mathbf{x} - \mathbf{w}^{(c)} \right\|_2,$$

parametrisiert durch die k Clusterzentren $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(k)} \in \mathbb{R}^d$. Jede solche Hypothese bildet jeden Merkmalsvektor auf seinen nächsten Clusterzentrum ab. Der Bereich des Merkmalsraum, der einem Clusterzentrum zugeordnet wird, ist durch Hyperebenen begrenzt, eine für jeden anderen Clusterzentrum (die gestrichelten Linien von Abb. 1). Somit ist jeder von k-Mittelwerte gefundene Cluster ein konvexer Bereich. Der Verlust, der durch einen Merkmalsvektor $\mathbf{x}$ entsteht, ist die quadrierte euklidischer Abstand zu seinem nächsten Clusterzentrum, $L(\mathbf{x}, h) = \|\mathbf{x} - h(\mathbf{x})\|_2^2$. Der clustering error wird dann als

durchschnittlicher Verlust, d.h. als empirical risk, über den gesamten Datensatz erhalten. Die Minimierung des clustering error ist daher ERM über $\mathcal{H}_k$.
Der clustering error ist eine nicht-konvexe Funktion der Clusterzentren: mit einem einzelnen Datenpunkt bei $x = 0$ und zwei skalaren Clusterzentren $w^{(1)}, w^{(2)} \in \mathbb{R}$, ist er $\min\big((w^{(1)})^2, (w^{(2)})^2\big)$, sodass sowohl $(1, 0)$ als auch $(0, 1)$ globale Optima mit dem Wert 0 sind, aber ihr Mittelpunkt $(0.5, 0.5)$ 0.25 ergibt. Die Nicht-Konvexität lässt den Lloyd's algorithm mit mehr als einem Fixpunkt zurück. Man betrachte die vier Datenpunkte $(\pm 5, \pm 1)^\top$ und $k = 2$ (Abb. 3). Ausgehend von einem Clusterzentrum links und einem rechts gruppiert der Zuordnungsschritt die beiden linken und die beiden rechten Datenpunkte, und der Mittelungsschritt liefert die Clusterzentren $(-5, 0)^\top$ und $(5, 0)^\top$, mit einem clustering error von 1. Ausgehend von einem Clusterzentrum oben und einem unten liefern die beiden Schritte $(0, 1)^\top$ und $(0, -1)^\top$, mit einem clustering error von 25. Beide Paare sind Fixpunkte: mit den Clusterzentren $(\pm 5, 0)^\top$ ist jeder Datenpunkt näher am Clusterzentrum auf seiner eigenen Seite, da $1 < \sqrt{101}$, und mit $(0, \pm 1)^\top$ ist jeder Datenpunkt näher am Clusterzentrum auf seiner eigenen Ebene, da $5 < \sqrt{29}$. Keiner der Starts bewegt sich, und der zweite hat den 25-fachen clustering error des ersten. Da der Lloyd's algorithm an einem solchen Fixpunkt, einem lokalen minimum des clustering error, anhalten kann, wird er typischerweise mehrmals mit verschiedenen Initialisierungen ausgeführt, wobei der Lauf mit dem kleinsten clustering error beibehalten wird (Hastie et al., 2009, Sect. 14.3.6). Eine Variante von k-Mittelwerte, die als k-means++ bezeichnet wird, verwendet eine spezifische Konstruktion der anfänglichen Clusterzentren.



(a) start left/right: error 1 (b) start above/below: error 25

Abbildung 3: Zwei Fixpunkte des Lloyd's algorithm für die vier Datenpunkte $(\pm 5, \pm 1)^\top$ (gefüllte Kreise) und $k = 2$. (a) Die Clusterzentren $(\pm 5, 0)^\top$ (Kreuze) mit der Entscheidungsgrenze der Nächste-Zentroid-Regel, der x_2-Achse (gestrichelte Linie); clustering error 1. (b) Die Clusterzentren $(0, \pm 1)^\top$ (Kreuze) mit der Grenze entlang der x_1-Achse (gestrichelte Linie); clustering error 25. Jedes Paar ist der Mittelwert der Datenpunkte auf seiner eigenen Seite seiner Grenze, sodass sich keines bewegt.

Der Lloyd's algorithm ist eine fixed-point iteration. Man sammle die Clusterzentren in einem Vektor $\mathbf{w} := \big(\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(k)}\big)$. Eine Iteration, der Zuordnungsschritt (1) gefolgt vom Mittelungsschritt (2), wendet einen Operator $\mathcal{F}$ auf die aktuellen Clusterzentren an, sodass die Methode die fixed-point iteration $\mathbf{w}^{(t+1)} = \mathcal{F}\big(\mathbf{w}^{(t)}\big)$ für $t = 0, 1, \ldots$ ist. Die Fixpunkte von $\mathcal{F}$ sind Cluster-

zentren, die jeweils mit dem Mittelwert der ihnen zugewiesenen Datenpunkte übereinstimmen. Die convergence folgt aus einem monotonen Abstieg und nicht aus einer contractive operator-Eigenschaft: Jeder der beiden Schritte minimiert den clustering error über eine Menge von Variablen, während die andere festgehalten wird, sodass die Anwendung von $\mathcal{F}$ den clustering error niemals erhöht. Eine Nicht-Erhöhung allein schließt ein Durchlaufen von Partitionen mit gleichem clustering error nicht aus. Wenn ein Datenpunkt sein Clusterzentrum im Zuordnungsschritt (1) nur dann ändert, wenn ein anderes strikt näher ist, senkt jede Änderung der Partition den clustering error strikt, sodass keine Partition zweimal besucht wird. Ein endlicher Datensatz lässt nur endlich viele Partitionen zu, sodass die Partition nach endlich vielen Iterationen aufhört sich zu ändern, und der Mittelungsschritt die Clusterzentren dann unverändert lässt: Die Iterierten haben einen Fixpunkt erreicht (Selim and Ismail, 1984). Auf den Patches von Abb. 1 erreichte der Lloyd's algorithm einen Fixpunkt nach 12 Iterationen, und der clustering error erhöhte sich dabei nie.

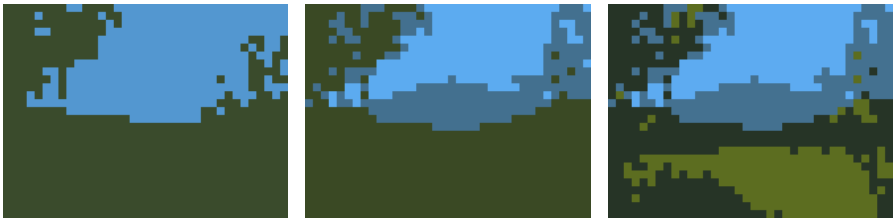
Der Lloyd's algorithm für k-Mittelwerte ist der Grenzwert des EM-Algorithmus (EM) Algorithmus für ein Gaußsches Mischmodell (GMM) unter zusätzlichen Randbedingungen. Man fixiere jede Cluster Wahrscheinlichkeit des GMM auf $1/k$ und jede Kovarianzmatrix auf $\sigma^2\mathbf{I}$, ein Vielfaches der EinheitsMatrix mit einer gemeinsamen σ^2. Die A-posteriori-Verteilung des Cluster-Index hängt dann von $\mathbf{x}$ nur über ihre euklidische Abstände zu den Mittelwerte ab, und wenn σ^2 gegen 0 schrumpft, konzentriert sie sich auf die nächste. Im Grenzwert ($\sigma^2 \to 0$) wird der E-Schritt des EM zum Zuordnungsschritt (1) und der M-Schritt zum Mittelungsschritt (2) des Lloyd's algorithm (Bishop, 2006, Sect. 9.3.2). Auf den Patches von Abb. 1 mit $\sigma^2 = 10^{-4}$ ordnet der EM Algorithmus 99.4% der Patches wie k-Mittelwerte zu. Die Kovarianzmatrizen und die Cluster Wahrscheinlichkeiten freizulassen, ermöglicht es einem GMM, eine Zugehörigkeitsgrad bereitzustellen, wo k-Mittelwerte jeden Datenpunkt einem Cluster zuweist.

Die Verlustfunktion ist das, was k-Mittelwerte von seinen Verallgemeinerungen trennt, und sie entscheidet, was der Mittelungsschritt des Lloyd's algorithm berechnet. Ersetzt man die quadrierte euklidischer Abstand durch die euklidischer Abstand selbst, $L(\mathbf{x}, h) = \|\mathbf{x} - h(\mathbf{x})\|_2$, erhält man das k-Medians-Optimierungsproblem. Der Zuordnungsschritt (1) bleibt unverändert, da das nächste Clusterzentrum unter beiden Verluste dasselbe ist, aber der Mittelungsschritt (2) wird zum geometric median (GM) der einem Cluster zugewiesenen Datenpunkte, der keine geschlossene Form hat und durch Weiszfelds fixed-point iteration berechnet wird (Weiszfeld, 1937). Der Gewinn ist die robustness gegenüber Ausreißer: Das Hinzufügen von 10 weit entfernten Patches zum Vegetations-Cluster von Abb. 1 verschiebt deren Mittelwert um 0.044, aber deren GM um 0.004, während k-Median 98.5% der Patches wie k-Mittelwerte zuweist.

Der Mittelungsschritt ist nicht allein an die quadrierte euklidischer Abstand gebunden: Für jede Bregman divergence $L(\mathbf{x}, h) = D_\phi\big(\mathbf{x}, h(\mathbf{x})\big)$ als Verlust minimiert der Mittelwert der zugewiesenen Datenpunkte immer noch deren summierte Verlust, sodass sich nur der Zuordnungsschritt ändert (Banerjee et al.,

2005, Prop. 1). Die euklidischer Abstand ist keine Bregman divergence, weshalb sie den Mittelungsschritt ändert. Beschränkt man jedes Clusterzentrum darauf, einer der ihm zugewiesenen Datenpunkte zu sein, erhält man k-Medoide, die nur die paarweisen Abstände zwischen Datenpunkte verwenden und daher auf Datenpunkte ohne numerische Merkmalsvektoren anwendbar sind (Hastie et al., 2009, Sect. 14.3.10; Kaufman and Rousseeuw, 1990).

Zurück zu k-Mittelwerte selbst: Abb. 4 zeigt den Effekt der Anzahl k der Cluster auf die Patches von Abb. 1. Jeder Patch ist mit der durchschnittlichen Farbe seines Cluster bemalt, was eine image segmentation ist. Mit $k = 2$ werden Himmel und Berg von der Vegetation getrennt. Mit $k = 3$ bilden Berg und Dunst ein eigenes Cluster. Mit $k = 4$ teilt sich die Vegetation in die sonnenbeschienene Wiese und das dunkle Laub. Der clustering error nimmt mit k ab, von 0.034 für $k = 2$ auf 0.018 für $k = 3$ und 0.011 für $k = 4$, und er erreicht 0, wenn jeder Datenpunkt sein eigenes Clusterzentrum ist.



(a) (b) (c)

Abbildung 4: Die Fotografie in Patch-Auflösung, jeder Patch mit der durchschnittlichen Farbe seines Cluster unter k-Mittelwerte bemalt, mit (a) $k = 2$, (b) $k = 3$ und (c) $k = 4$. In einem Graustufendruck unterscheiden sich die Cluster durch Helligkeit: Der Himmel ist am hellsten, Berg und Dunst liegen dazwischen, und die Vegetation ist am dunkelsten; in (c) liegen die sonnenbeschienene Wiese und der Berg in der Helligkeit nahe beieinander. Daten generiert von `pythondemos/kmeans.py`

Synonyme: k-Means-Clustering.

Siehe auch: hartes Clustering, Clustering, Cluster, Clusterzentrum, clustering error, Lloyd's algorithm, fixed-point iteration, k-means++, euklidischer Abstand, ERM, Hypothesenraum, Verlustfunktion, GMM, EM, soft clustering, Zugehörigkeitsgrad, GM, Bregman divergence.

Querverweise

Literatur

1. Bishop (2006). Pattern Recognition and Machine Learning. Springer Science+Business Media, New York, NY, USA.

2. Banerjee et al. (2005). Clustering with Bregman Divergences. J. Mach. Learn. Res., 6(58), pp. 1705–1749.

3. Kaufman and Rousseeuw (1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York, NY, USA.

4. Lloyd (1982). Least squares quantization in PCM. IEEE Trans. Inf. Theory, 28(2), pp. 129–137.

5. MacQueen (1967). Some methods for classification and analysis of multivariate observations. In: Proc. Fifth Berkeley Symp. Math. Statist. Probab., pp. 281–297. University of California Press.

6. Mahajan et al. (2009). The Planar k-Means Problem is NP-Hard. In: WALCOM: Algorithms and Computation, pp. 274–285. Springer-Verlag.

7. Selim and Ismail (1984). K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. IEEE Trans. Pattern Anal. Mach. Intell., 6(1), pp. 81–87.

8. Weiszfeld (1937). Sur le point pour lequel la somme des distances de n points donnés est minimum. Tôhoku Math. J., 43, pp. 355–386.

9. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.