

Masterarbeit

**Kernel k-means Methoden zur  
spektralen Clusteranalyse von  
Graphen**

Lukas Pradel

10. Dezember 2014

Gutachter:

Prof. Dr. Christian Sohler

Dipl.-Inf. Melanie Schmidt

Technische Universität Dortmund

Fakultät für Informatik

Lehrstuhl II - Effiziente Algorithmen und Komplexitätstheorie

<http://ls2-www.cs.tu-dortmund.de/>

Hier kommt eine Danksagung hin!





# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlegende Definitionen und Algorithmen</b>	<b>2</b>
2.1	Clustering und $k$ -means . . . . .	2
2.2	Graphen und Clusteranalyse von Graphen . . . . .	4
2.3	Kernel-Methoden und spektrales Clustering . . . . .	4
	<b>Literatur</b>	<b>6</b>
	<b>Erklärung</b>	<b>9</b>



# 1 Einleitung

## 2 Grundlegende Definitionen und Algorithmen

In diesem Kapitel definieren wir die für unsere Zwecke relevanten Begriffe im Kontext der Clusteranalyse und führen die wichtigen grundlegenden Algorithmen ein, deren Ideen für uns im Folgenden noch von Bedeutung sein werden. Wir gehen dabei nach Themengebieten geordnet vor: In Abschnitt 2.1 skizzieren wir kurz das Themengebiet der Clusteranalyse, definieren die üblichen Zielfunktionen und stellen zwei bedeutende Algorithmen vor. Abschnitt 2.2 führt kurz in die Graphentheorie sowie die Clusteranalyse von Graphen ein. In diesem Abschnitt werden wir zudem die von der klassischen Clusteranalyse sehr unterschiedlichen Optimierungskriterien für die Clusteranalyse von Graphen herausstellen. Schließlich fassen wir in Abschnitt 2.3 die wichtigsten Methoden und Algorithmen aus dem Bereich der spektralen Clusteranalyse zusammen und stellen zudem die wesentlichen Konzepte von Kernel-Methoden vor.

### 2.1 Clustering und $k$ -means

Clusteranalyse oder „Clustering“ beschäftigt sich mit der Einteilung von Objekten in Gruppen („Cluster“), sodass sich die Objekte innerhalb eines Clusters gemäß eines bestimmten Optimierungskriteriums ähnlich sind und von Objekten eines anderen Clusters unterscheiden. Es existieren zahlreiche grundsätzlich verschiedene Ansätze, Clusteringprobleme zu lösen. Wir beschränken uns in dieser Arbeit auf *partitionierende* Clusteringprobleme und -verfahren. Bei diesen soll eine Menge von  $d$ -dimensionalen Punkten, welche der erste Teil der Eingabe ist, gemäß einer Cluster-Zielfunktion möglichst optimal in genau  $k$  Cluster unterteilt werden, wobei  $k$  der ganzzahlige zweite Teil der Eingabe ist.

Für die Zielfunktion, welche die Nähe oder Ferne von Punkten zueinander quantifiziert, sind bei Eingabepunkten aus  $\mathbb{R}^d$  Metriken naheliegend. Intuitiv ist dabei die euklidische Distanz, welche als Zielfunktion für die beiden bekanntesten Clustering-Problemstellungen dient.

**Definition 2.1.1 ( $k$ -median und  $k$ -means).** Sei  $P \subset \mathbb{R}^d$  und  $k \in \mathbb{N}^+$ . Das  $k$ -median-Problem besteht darin, eine Menge von  $k$  (Cluster-)Zentren  $C = \{c_1, \dots, c_k\}$  mit  $c_i \in \mathbb{R}^d$  zu finden, sodass der folgende Term minimal wird:

$$\sum_{p \in P} \min_{c_i \in C} \|p - c_i\|$$

Das  $k$ -means-Problem unterscheidet sich nur darin, dass bei diesem die Summe der *quadrierten* euklidischen Distanzen zum jeweils nächstgelegenen Zentrum minimiert



werden soll, das heißt, dass der folgende Term minimiert werden soll:

$$\sum_{p \in P} \min_{c_i \in C} \|p - c\|^2$$

Beim *gewichteten*  $k$ -means-Problem werden den Eingabepunkten zusätzlich mit einer Funktion  $w : P \rightarrow \mathbb{R}$  Gewichte zugewiesen. Die zu minimierende Zielfunktion lautet dann entsprechend

$$\sum_{p \in P} \min_{c_i \in C} w(p) \|p - c\|^2$$

Sowohl das  $k$ -Median-Problem [MS84] als auch das  $k$ -means-Problem [ADHP09] sind optimal NP-schwer lösbar. Typischerweise werden zur Lösung daher approximative oder heuristische Algorithmen eingesetzt. Die bekannteste und bis heute sehr erfolgreiche Heuristik für das  $k$ -means-Problem ist der Algorithmus von Lloyd [Llo82]. Der Algorithmus wählt initial  $k$  zufällige Punkte aus der Eingabemenge oder sogar beliebige Punkte aus  $\mathbb{R}^d$  als initiale Clusterzentren. Anschließend wird jedem Punkt das am nächsten gelegene Zentrum zugewiesen. Dadurch entstehen die initialen Cluster mit ihren jeweiligen Zentren. Im zweiten Schritt wird das neue Zentrum eines jeden Clusters als der geometrische Zentroid des Clusters gewählt. Die Zuweisung von Punkten zum nächstgelegenen Cluster und die Neuberechnung der neuen Zentren werden solange alterniert, bis die Lösung konvergiert, also wenn sich die Zuordnungen der Punkte nicht mehr ändern. In der Praxis wird gelegentlich auch nach einer festen Anzahl von Iterationen terminiert.

---

**Algorithmus 1:** Algorithmus von Lloyd

---

**Eingabe:**  $P \subseteq \mathbb{R}^d, k \in \mathbb{N}^+$

**Ausgabe:**  $k$ -means-Clustering von  $P$

- 1 Wähle zufällig  $k$  Zentren  $c_1^{(0)}, \dots, c_k^{(0)}$  aus  $P$  oder  $\mathbb{R}^d$
  - 2  $S_i^{(0)} \leftarrow \{p \in P : \|p - c_i^{(0)}\|^2 \leq \|p - c_{i'}^{(0)}\|^2 \forall i' \in \{1, \dots, k\}\}$
  - 3 **repeat**
  - 4    $c_i^{(t)} \leftarrow \frac{1}{|S_i^{(t-1)}|} \sum_{p_j \in S_i^{(t-1)}} p_j$
  - 5    $S_i^{(t)} \leftarrow \{p \in P : \|p - c_i^{(t)}\|^2 \leq \|p - c_{i'}^{(t)}\|^2 \forall i' \in \{1, \dots, k\}\}$
  - 6 **until**  $S_i^{(t)} = S_i^{(t-1)}$
- 

Die asymptotische Laufzeit des Algorithmus beträgt  $\mathcal{O}(nkdi)$ , wobei  $i$  die Anzahl an durchgeführten Iterationen ist. Wenn der Algorithmus konvergiert und nicht durch eine feste Anzahl von Iterationen terminiert wird, wurde ein lokales Optimum gefunden, welches jedoch im Allgemeinen kein globales Optimum oder eine Approximation eines globalen Optimums ist. Die Güte des berechneten Clusterings hängt maßgeblich von der initialen Wahl der Cluster ab. Der Algorithmus

$k$ -means++ [AV07] setzt genau an dieser Stelle an: er berechnet auf einfache, aber dennoch geschickte Art und Weise die initialen Cluster und führt anschließend mit diesen die übrigen Schritte von Lloyds Algorithmus durch. Der Algorithmus wählt zunächst ein einzelnes Clusterzentrum  $c_1$  zufällig gleichverteilt aus der Eingabe-Punktmenge  $P$  und wählt alle weiteren Clusterzentren sukzessive nach der folgenden Vorschrift, bis insgesamt  $k$  Zentren gewählt wurden. Im Weiteren bezeichnen wir mit  $D(x)$  für einen Punkt  $x$  aus der Eingabe-Punktmenge  $P$  die geringste Distanz von  $x$  zum nächstgelegenen bereits gewählten Zentrum. In jeder Iteration wird als nächstes Zentrum  $c_i$  der Punkt  $x' \in P \setminus \{c_1, \dots, c_{i-1}\}$  mit Wahrscheinlichkeit  $\frac{D(x')^2}{\sum_{x \in P} D(x)}$  gewählt.

---

**Algorithmus 2:**  $k$ -means++

---

**Eingabe:** :  $P \subseteq \mathbb{R}^d, k \in \mathbb{N}^+$

**Ausgabe:** :  $k$  initiale Clusterzentren für  $P$

- 1 Wähle  $c_1$  zufällig gleichverteilt aus  $P$
- 2 **for**  $i \leftarrow 1$  **to**  $k$  **do**
- 3     Wähle den Punkt  $x' \in P \setminus \{c_1, \dots, c_{i-1}\}$  als Zentrum  $c_i$  mit  
       Wahrscheinlichkeit  $\frac{D(x')^2}{\sum_{x \in P} D(x)}$
- 4 Führe Lloyds Algorithmus mit den initialen Clusterzentren  $c_1, \dots, c_k$  aus.

---

Die  $k$  Zentren, die von  $k$ -means++ ausgewählt werden, sind eine  $\mathcal{O}(\log k)$ -Approximation für das  $k$ -means-Problem, die durch die anschließende Ausführung von Lloyds Algorithmus noch zu einem lokalen Optimum verbessert werden.

## 2.2 Graphen und Clusteranalyse von Graphen

## 2.3 Kernel-Methoden und spektrales Clustering



## Literatur

- [ADHP09] ALOISE, Daniel ; DESHPANDE, Amit ; HANSEN, Pierre ; POPAT, Preyas: NP-hardness of Euclidean sum-of-squares clustering. In: *Machine Learning* (2009), S. 245–248
- [AV07] ARTHUR, David ; VASSILVITSKII, Sergei: k-means++: The advantages of careful seeding. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, 2007, S. 1027–1035
- [Llo82] LLOYD, Stuart P.: Least squares quantization in PCM. In: *IEEE Transactions on Information Theory* 28 (1982), Nr. 2, S. 129–136
- [MS84] MEGIDDO, Nimrod ; SUPOWIT, Kenneth J.: On the Complexity of Some Common Geometric Location Problems. In: *SIAM J. Comput.* 13 (1984), Nr. 1, S. 182–196





Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Dortmund, den 10. Dezember 2014

Lukas Pradel

