Datenstrukturen, Algorithmen und Programmierung 2 (DAP2)

Approximationsalgorithmen

Das (diskrete) k-Center Clustering Problem

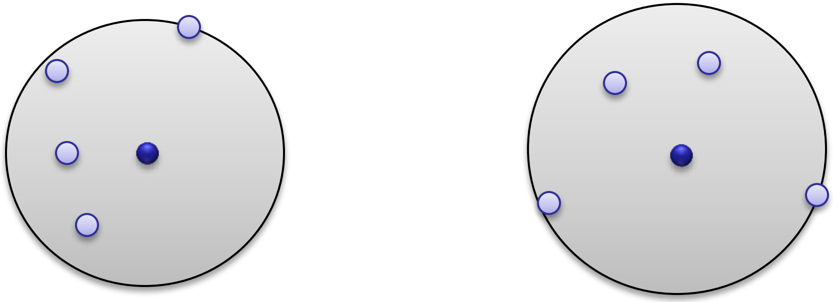
- Gegeben: Menge P von n Punkten in der Ebene
- Gesucht: Menge $C \in P$ von k Zentren, so dass die maximale Distanz der Punkte zum nächstgelegenen Zentrum, d.h.
  $\mathrm{cost}(P, C) = \max_{p \in P} d(p, C)$ minimiert wird, wobei
- $\mathrm{dist}(p, C) = \max_{q \in C} \mathrm{dist}(p, q)$ und
- $\mathrm{dist}(p, q)$ bezeichnet den Abstand (Euklidische Distanz) von p und q

Approximationsalgorithmen

Beispiel

Approximationsalgorithmen

Beispiel



Alternative Formulierung

Finde k Scheiben mit Zentrum aus P, die alle Punkte abdecken und deren Maximaler Radius minimiert wird.

Approximationsalgorithmen

Typische Anwendung

- Punkte symbolisieren Städte
- Will Mobilfunkmasten mit möglichst geringer Leistung aufstellen, so dass alle Städte versorgt sind

Allgemeiner

- Punkte (typischerweise in höheren Dimensionen) sind „Beschreibungen von Objekten"
- Will Objekte in Gruppen von ähnlichen Objekten unterteilen

Approximationsalgorithmen

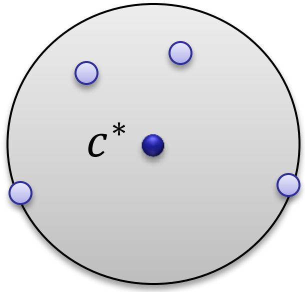
Ein Gedankenexperiment

- Nehmen wir an, wir kennen die Kosten r einer optimalen Lösung, d.h. wir wissen, dass man mit Scheiben mit Radius r und Zentrum aus P die Punkte abdecken kann
- Wir werden zeigen, dass wir dann einen einfachen 2-Approximationsalgorithmus finden können

Approximationsalgorithmen

Idee

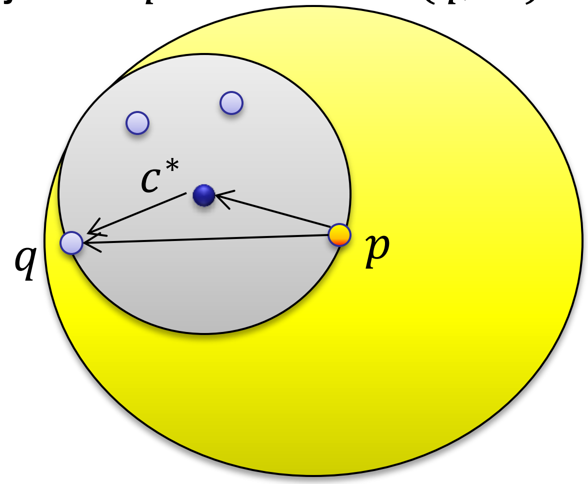
- Wir nutzen Existenz von Lösung C^* mit Radius (Kosten) r
- Betrachte Punkt $p \in P$
- Dann gibt es Zentrum $c^* \in C^*$ mit $\mathrm{dist}(p, c^*) \leq r$
- Nehmen wir nun p als Zentrum anstelle von c^* und verdoppeln wir den Radius, so decken wir jeden Punkt q ab, der von c^* mit Radius r abgedeckt wurde, d.h.
- für jedes $q \in P$ mit $\mathrm{dist}(q, c^*) \leq r$ gilt $\mathrm{dist}(p, q) \leq \mathrm{dist}(p, c^*) + \mathrm{dist}(c^*, q) \leq 2r$



Approximationsalgorithmen

Idee

- Wir nutzen Existenz von Lösung C^* mit Radius (Kosten) r
- Betrachte Punkt $p \in P$
- Dann gibt es Zentrum $c^* \in C^*$ mit $\mathrm{dist}(p, c^*) \leq r$
- Nehmen wir nun p als Zentrum anstelle von c^* und verdoppeln wir den Radius, so decken wir jeden Punkt q ab, der von c^* mit Radius r abgedeckt wurde, d.h.
- für jedes $q \in P$ mit $\mathrm{dist}(q, c^*) \leq r$ gilt $\mathrm{dist}(p, q) \leq \mathrm{dist}(p, c^*) + \mathrm{dist}(c^*, q) \leq 2r$



Approximationsalgorithmen

k-Center1(P, k)

1. $C \leftarrow \emptyset$; $P' \leftarrow P$
2. **while** $P' \neq \emptyset$ **do**
3. Wähle beliebigen Punkt $p \in P'$
4. $C \leftarrow C \cup \{p\}$
5. Lösche alle Punkte aus P' mit Distanz höchstens $2r$ von p
6. **if** $|C| \leq k$ **then return** C
7. **else return** „Es gibt keine Menge von k Zentren mit Radius r“

Approximationsalgorithmen

k-Center1(P,k)

1. $C \leftarrow \emptyset$; $P' \leftarrow P$
2. **while** $P' \neq \emptyset$ **do**
3. Wähle beliebigen Punkt $p \in P'$
4. $C \leftarrow C \cup \{p\}$
5. Lösche alle Punkte aus P' mit Distanz höchstens $2r$ von p
6. **if** $|C| \leq k$ **then return** C
7. **else return** „Es gibt keine Menge von k Zentren mit Radius r"

Offensichtlich gilt

Jede Menge von k Zentren, die der Algorithmus zurückgibt, hat Kosten höchstens $2r$.

Approximationsalgorithmen

Lemma 82

Wenn Algorithmus k-$\mathrm{Center1}$ mehr als k Zentren auswählt, dann gilt für jede Menge $C^* \subseteq P$ von k Zentren, dass $\mathrm{cost}(P, C^*) > r$ ist.

Approximationsalgorithmen

Lemma 82

Wenn Algorithmus k-$\mathrm{Center1}$ mehr als k Zentren auswählt, dann gilt für jede Menge $C^* \subseteq P$ von k Zentren, dass $\mathrm{cost}(P, C^*) > r$ ist.

Beweis (durch Widerspruch)

- Annahme: Es gibt C^* mit $\mathrm{cost}(P, C^*) \leq r$ und $|C^*| \leq k$ und $|C| > k$.

Approximationsalgorithmen

Lemma 82

Wenn Algorithmus k-$\mathrm{Center1}$ mehr als k Zentren auswählt, dann gilt für jede Menge $C^* \subseteq P$ von k Zentren, dass $\mathrm{cost}(P, C^*) > r$ ist.

Beweis (durch Widerspruch)

- Annahme: Es gibt C^* mit $\mathrm{cost}(P, C^*) \leq r$ und $|C^*| \leq k$ und $|C| > k$.
- Sei C die Menge der Zentren, die k-$\mathrm{Center1}$ auswählt

Approximationsalgorithmen

Lemma 82

Wenn Algorithmus k-$\mathrm{Center1}$ mehr als k Zentren auswählt, dann gilt für jede Menge $C^* \subseteq P$ von k Zentren, dass $\mathrm{cost}(P, C^*) > r$ ist.

Beweis (durch Widerspruch)

- Annahme: Es gibt C^* mit $\mathrm{cost}(P, C^*) \le r$ und $|C^*| \le k$ und $|C| > k$.
- Sei C die Menge der Zentren, die k-$\mathrm{Center1}$ auswählt
- Da $C \subseteq P$ gibt es für jedes $p \in C$ (mindestens) ein $c^* \in C^*$ mit $\mathrm{dist}(p, c*) \le r$
- Wir nennen c^* nah zu p

Approximationsalgorithmen

Lemma 82

Wenn Algorithmus k-$\mathrm{Center1}$ mehr als k Zentren auswählt, dann gilt für jede Menge $C^* \subseteq P$ von k Zentren, dass $\mathrm{cost}(P, C^*) > r$ ist.

Beweis (durch Widerspruch)

- Annahme: Es gibt C^* mit $\mathrm{cost}(P, C^*) \leq r$ und $|C^*| \leq k$ und $|C| > k$.
- Sei C die Menge der Zentren, die k-$\mathrm{Center1}$ auswählt
- Da $C \subseteq P$ gibt es für jedes $p \in C$ (mindestens) ein $c^* \in C^*$ mit $\mathrm{dist}(p, c*) \leq r$
- Wir nennen c^* nah zu p
- Behauptung: Kein c^* kann nah zu zwei $p \in C$ sein (Beweis später)

Approximationsalgorithmen

Lemma 82

Wenn Algorithmus k-$\mathrm{Center1}$ mehr als k Zentren auswählt, dann gilt für jede Menge $C^* \subseteq P$ von k Zentren, dass $\mathrm{cost}(P, C^*) > r$ ist.

Beweis (durch Widerspruch)

- Annahme: Es gibt C^* mit $\mathrm{cost}(P, C^*) \leq r$ und $|C^*| \leq k$ und $|C| > k$.
- Sei C die Menge der Zentren, die k-$\mathrm{Center1}$ auswählt
- Da $C \subseteq P$ gibt es für jedes $p \in C$ (mindestens) ein $c^* \in C^*$ mit $\mathrm{dist}(p, c*) \leq r$
- Wir nennen c^* nah zu p
- Behauptung: Kein c^* kann nah zu zwei $p \in C$ sein (Beweis später)
- Damit folgt: $|C^*| \geq |C|$ und somit Widerspruch zu $|C^*| \leq k$ und und $|C| > k$.

Approximationsalgorithmen

Lemma 82

Wenn Algorithmus k-$\mathrm{Center1}$ mehr als k Zentren auswählt, dann gilt für jede Menge $C^* \subseteq P$ von k Zentren, dass $\mathrm{cost}(P, C^*) > r$ ist.

Beweis (durch Widerspruch)

- Annahme: Es gibt C^* mit $\mathrm{cost}(P, C^*) \leq r$ und $|C^*| \leq k$ und $|C| > k$.
- Sei C die Menge der Zentren, die k-$\mathrm{Center1}$ auswählt
- Da $C \subseteq P$ gibt es für jedes $p \in C$ (mindestens) ein $c^* \in C^*$ mit $\mathrm{dist}(p, c *) \leq r$
- Wir nennen c^* nah zu p
- Behauptung: Kein c^* kann nah zu zwei $p \in C$ sein (Beweis später)
- Damit folgt: $|C^*| \geq |C|$ und somit Widerspruch zu $|C^*| \leq k$ und und $|C| > k$.

Approximationsalgorithmen

Lemma 82

Wenn Algorithmus k-$\mathrm{Center1}$ mehr als k Zentren auswählt, dann gilt für jede Menge $C^* \subseteq P$ von k Zentren, dass $\mathrm{cost}(P, C^*) > r$ ist.

Beweis (durch Widerspruch)

- Behauptung: Kein c^* kann nah zu zwei $p \in C$ sein
- Beweis der Behauptung:

Approximationsalgorithmen

Lemma 82

Wenn Algorithmus k-$\mathrm{Center1}$ mehr als k Zentren auswählt, dann gilt für jede Menge $C^* \subseteq P$ von k Zentren, dass $\mathrm{cost}(P, C^*) > r$ ist.

Beweis (durch Widerspruch)

- Behauptung: Kein c^* kann nah zu zwei $p \in C$ sein
- Beweis der Behauptung:
- Alle Paare von Zentren p,q aus C haben Abstand $> 2r$

Approximationsalgorithmen

Lemma 82

Wenn Algorithmus k-$\mathrm{Center1}$ mehr als k Zentren auswählt, dann gilt für jede Menge $C^* \subseteq P$ von k Zentren, dass $\mathrm{cost}(P, C^*) > r$ ist.

Beweis (durch Widerspruch)

- Behauptung: Kein c^* kann nah zu zwei $p \in C$ sein
- Beweis der Behauptung:
- Alle Paare von Zentren p,q aus C haben Abstand $> 2r$
- Wäre nun für ein Zentrum c^* $\mathrm{dist}(p, c^*) \leq r$ und $\mathrm{dist}(q, c^*) \leq r$, so würde $\mathrm{dist}(p, q) \leq \mathrm{dist}(p, c^*) + \mathrm{dist}(c^*, q) = \mathrm{dist}(p, c^*) + \mathrm{dist}(c^*, q) \leq 2r$ gelten. Widerspruch!

Approximationsalgorithmen

Lemma 82

Wenn Algorithmus k-$\mathrm{Center1}$ mehr als k Zentren auswählt, dann gilt für jede Menge $C^* \subseteq P$ von k Zentren, dass $\mathrm{cost}(P, C^*) > r$ ist.

Beweis (durch Widerspruch)

- Behauptung: Kein c^* kann nah zu zwei $p \in C$ sein
- Beweis der Behauptung:
- Alle Paare von Zentren p,q aus C haben Abstand $> 2r$
- Wäre nun für ein Zentrum c^* $\mathrm{dist}(p, c^*) \leq r$ und $\mathrm{dist}(q, c^*) \leq r$, so würde $\mathrm{dist}(p, q) \leq \mathrm{dist}(p, c^*) + \mathrm{dist}(c^*, q) = \mathrm{dist}(p, c^*) + \mathrm{dist}(c^*, q) \leq 2r$ gelten. Widerspruch!

Approximationsalgorithmen

Was, wenn wir r nicht kennen?

- Wir wissen nicht, welche Punkte Distanz größer als $2r$ von den bisher ausgewählten Zentren haben
- Idee: Wähle immer den am weitesten entfernten Punkt, d.h. der $\mathrm{dist}(p, C)$ maximiert
- Gibt es einen Punkt mit $\mathrm{dist}(p, C) > 2r$, dann ist es dieser

Approximationsalgorithmen

k-Center2(P, k)

1. Wähle beliebigen Punkt $p \in P$ und setze $C \leftarrow \{p\}$
3. **while** $|C| < k$ **do**
3. Wähle Punkt $p \in P$, der $\mathrm{dist}(p, C)$ maximiert
4. $C \leftarrow C \cup \{p\}$
5. **return** C

Approximationsalgorithmen

Satz 83

Algorithmus $\mathrm{k\text{-}Center2}$ ist ein 2-Approximationsalgorithmus für das diskrete k-Center Clustering Problem. Algorithmus $\mathrm{k\text{-}Center2}$ kann mit Laufzeit $\mathbf{O}(nk)$ implementiert werden.

Beweis

- Zunächst zur Laufzeit:

Approximationsalgorithmen

Satz 83

Algorithmus $\mathrm{k\text{-}Center2}$ ist ein 2-Approximationsalgorithmus für das diskrete k-Center Clustering Problem. Algorithmus $\mathrm{k\text{-}Center2}$ kann mit Laufzeit $\mathbf{O}(nk)$ implementiert werden.

Beweis

- Zunächst zur Laufzeit:
- Um den Algorithmus in $\mathbf{O}(nk)$ Zeit zu implementieren, müssen wir jeden Schleifendurchlauf in $\mathbf{O}(n)$ Zeit erledigen können. Dazu speichern wir uns für jeden Punkt p den Wert $\mathrm{dist}(p, C)$.

Approximationsalgorithmen

Satz 83

Algorithmus $\mathrm{k\text{-}Center2}$ ist ein 2-Approximationsalgorithmus für das diskrete k-Center Clustering Problem. Algorithmus $\mathrm{k\text{-}Center2}$ kann mit Laufzeit $\mathbf{O}(nk)$ implementiert werden.

Beweis

- Zunächst zur Laufzeit:
- Um den Algorithmus in $\mathbf{O}(nk)$ Zeit zu implementieren, müssen wir jeden Schleifendurchlauf in $\mathbf{O}(n)$ Zeit erledigen können. Dazu speichern wir uns für jeden Punkt p den Wert $\mathrm{dist}(p, C)$.
- Wird nun ein neues Zentrum c in C eingefügt, so müssen wir nur für jeden Punkt überprüfen, ob $\mathrm{dist}(p, c) < \mathrm{dist}(p, C)$ ist und ggf. $\mathrm{dist}(p, C)$ aktualisieren. Dies geht insgesamt in $\mathbf{O}(n)$ Zeit.

Approximationsalgorithmen

Satz 83

Algorithmus $\mathrm{k\text{-}Center2}$ ist ein 2-Approximationsalgorithmus für das diskrete k-Center Clustering Problem. Algorithmus $\mathrm{k\text{-}Center2}$ kann mit Laufzeit $\mathbf{O}(nk)$ implementiert werden.

Beweis

- Zunächst zur Laufzeit:
- Um den Algorithmus in $\mathbf{O}(nk)$ Zeit zu implementieren, müssen wir jeden Schleifendurchlauf in $\mathbf{O}(n)$ Zeit erledigen können. Dazu speichern wir uns für jeden Punkt p den Wert $\mathrm{dist}(p, C)$.
- Wird nun ein neues Zentrum c in C eingefügt, so müssen wir nur für jeden Punkt überprüfen, ob $\mathrm{dist}(p, c) < \mathrm{dist}(p, C)$ ist und ggf. $\mathrm{dist}(p, C)$ aktualisieren. Dies geht insgesamt in $\mathbf{O}(n)$ Zeit.

Approximationsalgorithmen

Satz 83

Algorithmus $\mathrm{k\text{-}Center2}$ ist ein 2-Approximationsalgorithmus für das diskrete k-Center Clustering Problem. Algorithmus $\mathrm{k\text{-}Center2}$ kann mit Laufzeit $\mathbf{O}(nk)$ implementiert werden.

Beweis

- Wir bezeichnen nun mit r die Kosten einer optimalen Lösung C^*.
  Annahme: Algorithmus liefert Menge C mit Kosten $> 2r$.

Approximationsalgorithmen

Satz 83

Algorithmus $\mathrm{k\text{-}Center2}$ ist ein 2-Approximationsalgorithmus für das diskrete k-Center Clustering Problem. Algorithmus $\mathrm{k\text{-}Center2}$ kann mit Laufzeit $\mathbf{O}(nk)$ implementiert werden.

Beweis

- Wir bezeichnen nun mit r die Kosten einer optimalen Lösung C^*.
  Annahme: Algorithmus liefert Menge C mit Kosten $> 2r$.
- Dann gibt es einen Punkt p mit $\mathrm{dist}(p, C) > 2r$.

Approximationsalgorithmen

Satz 83

Algorithmus $\mathrm{k\text{-}Center2}$ ist ein 2-Approximationsalgorithmus für das diskrete k-Center Clustering Problem. Algorithmus $\mathrm{k\text{-}Center2}$ kann mit Laufzeit $\mathbf{O}(nk)$ implementiert werden.

Beweis

- Wir bezeichnen nun mit r die Kosten einer optimalen Lösung C^*.
  Annahme: Algorithmus liefert Menge C mit Kosten $> 2r$.
- Dann gibt es einen Punkt p mit $\mathrm{dist}(p, C) > 2r$.
- Da der Algorithmus immer den Punkt auswählt, der maximalen Abstand zu den bisher ausgewählten Zentren hat, haben alle ausgewählten Zentren Abstand $> 2r$ zur den bisher ausgewählten Zentren

Approximationsalgorithmen

Satz 83

Algorithmus $\mathrm{k\text{-}Center2}$ ist ein 2-Approximationsalgorithmus für das diskrete k-Center Clustering Problem. Algorithmus $\mathrm{k\text{-}Center2}$ kann mit Laufzeit $\mathbf{O}(nk)$ implementiert werden.

Beweis

- Wir bezeichnen nun mit r die Kosten einer optimalen Lösung C^*. Annahme: Algorithmus liefert Menge C mit Kosten $> 2r$.
- Dann gibt es einen Punkt p mit $\mathrm{dist}(p, C) > 2r$.
- Da der Algorithmus immer den Punkt auswählt, der maximalen Abstand zu den bisher ausgewählten Zentren hat, haben alle ausgewählten Zentren Abstand $> 2r$ zur den bisher ausgewählten Zentren
- Somit würde Algorithmus $\mathrm{k\text{-}Center1}$ auf dieser Eingabe mehr als k Zentren zurückgeben. Damit gilt nach Lemma 82 $\mathrm{cost}(P, C^*) > r$. Widerspruch!

Approximationsalgorithmen

Satz 83

Algorithmus $\mathrm{k\text{-}Center2}$ ist ein 2-Approximationsalgorithmus für das diskrete k-Center Clustering Problem. Algorithmus $\mathrm{k\text{-}Center2}$ kann mit Laufzeit $\mathbf{O}(nk)$ implementiert werden.

Beweis

- Wir bezeichnen nun mit r die Kosten einer optimalen Lösung C^*.
 Annahme: Algorithmus liefert Menge C mit Kosten $> 2r$.
- Dann gibt es einen Punkt p mit $\mathrm{dist}(p, C) > 2r$.
- Da der Algorithmus immer den Punkt auswählt, der maximalen Abstand zu den bisher ausgewählten Zentren hat, haben alle ausgewählten Zentren Abstand $> 2r$ zur den bisher ausgewählten Zentren
- Somit würde Algorithmus $\mathrm{k\text{-}Center1}$ auf dieser Eingabe mehr als k Zentren zurückgeben. Damit gilt nach Lemma 82 $\mathrm{cost}(P, C^*) > r$. Widerspruch!

Approximationsalgorithmen

Zusammenfassung & Kommentare

- Man kann viele Probleme approximativ schneller lösen als exakt
(für die drei Beispiele sind keine Algorithmen mit Laufzeit $\mathbf{O}(n^c)$ für eine Konstante c bekannt)
- Gierige Algorithmen sind häufig Approximationsalgorithmen