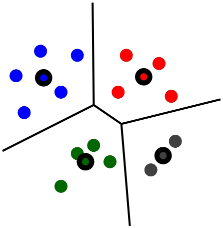
k-Means has Polynomial Smoothed Complexity

David Arthur
Google

Bodo Manthey
University of Twente
The Netherlands

Heiko Röglin
universität bonn



Pearls of Algorithms
Winter 2010/11

Data Clustering

Data Clustering:

Input: set of objects X

Ouput: partition of X into k classes $C_1, \ldots, C_k$

Goal: maximize similarity among objects in the same class

Data Clustering

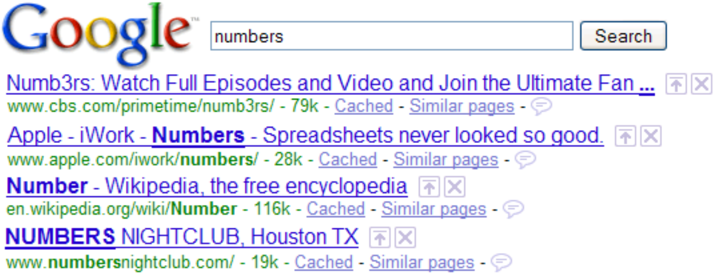
Data Clustering:

Input: set of objects X

Ouput: partition of X into k classes $C_1, \ldots, C_k$

Goal: maximize similarity among objects in the same class

First Example: Clustering of Web Pages



Data Clustering

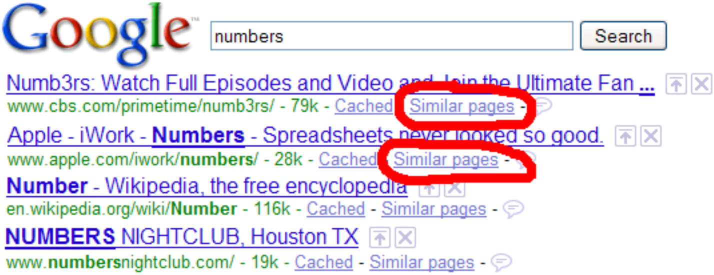
Data Clustering:

Input: set of objects X

Ouput: partition of X into k classes $C_1, \ldots, C_k$

Goal: maximize similarity among objects in the same class

First Example: Clustering of Web Pages



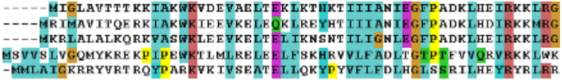
Data Clustering — Further Examples

Second Example: Color Reduction

Second Example: Color Reduction

Second Example: Color Reduction

Third Example: Protein Clustering



Outline

Main Questions

1. **Data Clustering**
   What is the *k*-means method?
2. **Smoothed Analysis**
   What can we do when worst case analysis is too pessimistic?
3. **Smoothed Analysis of the *k*-Means Method**
   What is the smoothed complexity of the *k*-means method?
4. **Extensions and Conclusions**

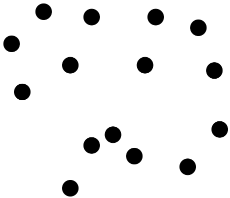
Outline

Main Questions

1. **Data Clustering**
   What is the *k*-means method?
2. **Smoothed Analysis**
   What can we do when worst case analysis is too pessimistic?
3. **Smoothed Analysis of *k*-Means Method**
   What is the smoothed complexity of the *k*-means method?
4. **Extensions and Conclusions**

Data Clustering

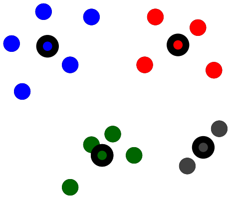
Input: point set $X \subseteq \mathbb{R}^d$, $|X| = n$
number of clusters k



Data Clustering

Input: point set $X \subseteq \mathbb{R}^d$, $|X| = n$
number of clusters k

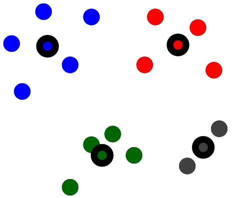
Output: partition $C_1, \ldots, C_k$
centers $c_1, \ldots, c_k \in \mathbb{R}^d$



Input: point set $X \subseteq \mathbb{R}^d$, $|X| = n$
number of clusters k

Output: partition $C_1, \ldots, C_k$
centers $c_1, \ldots, c_k \in \mathbb{R}^d$

Goal: $\min \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2$

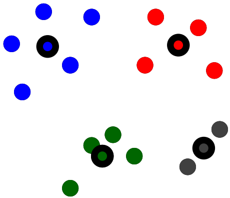


Data Clustering

Input: point set $X \subseteq \mathbb{R}^d$, $|X| = n$
number of clusters k

Output: partition $C_1, \ldots, C_k$
centers $c_1, \ldots, c_k \in \mathbb{R}^d$

Goal: $\min \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2$



Theory: The problem is NP-hard, but a PTAS exists.
(running time is exponential in k)

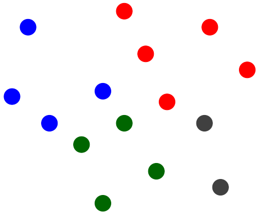
Practice: k-Means Method.

k-Means Method

Local search based on two observations:

1. clusters C_i fixed

$\Rightarrow$ centers $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$

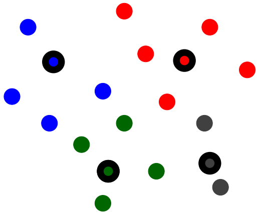


k-Means Method

Local search based on two observations:

1. clusters C_i fixed

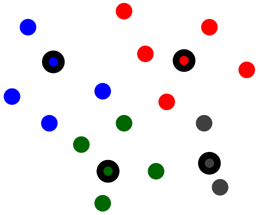
$\Rightarrow$ centers $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$



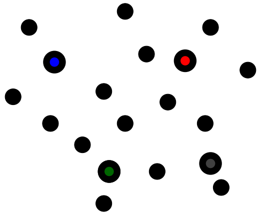
k-Means Method

Local search based on two observations:

1. clusters C_i fixed
   $\Rightarrow$ centers $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$



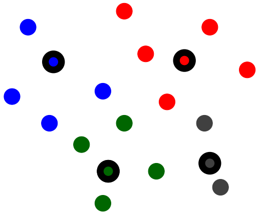
2. centers c_i fixed
   $\Rightarrow$ clusters C_i fixed



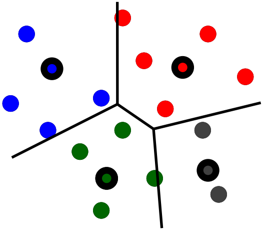
k-Means Method

Local search based on two observations:

1. clusters C_i fixed
$\Rightarrow$ centers $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$



2. centers c_i fixed
$\Rightarrow$ clusters C_i fixed

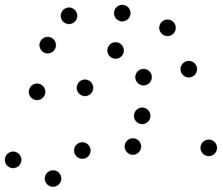


k-Means Method

k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

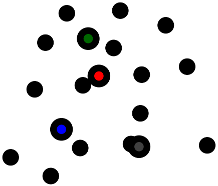
1. **choose $c_1, \ldots, c_k \in \mathbb{R}^d$**
2. Repeat
3. partition X into $C_1, \ldots, C_k$
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. Until stable



k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

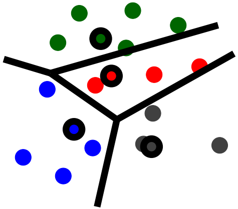
1. choose $c_1, \dots, c_k \in \mathbb{R}^d$
2. Repeat
3. **partition X into $C_1, \dots, C_k$**
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. Until stable



k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

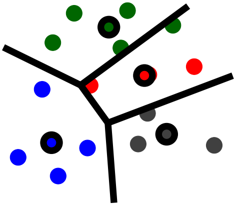
1. choose $c_1, \ldots, c_k \in \mathbb{R}^d$
2. Repeat
3. partition X into $C_1, \ldots, C_k$
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. Until stable



k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

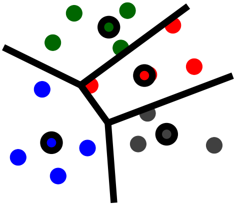
1. choose $c_1, \ldots, c_k \in \mathbb{R}^d$
2. Repeat
3. **partition X into $C_1, \ldots, C_k$**
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. Until stable



k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

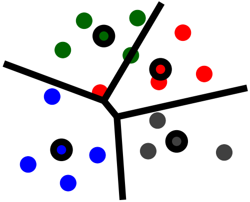
1. choose $c_1, \ldots, c_k \in \mathbb{R}^d$
2. Repeat
3. partition X into $C_1, \ldots, C_k$
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. Until stable



k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

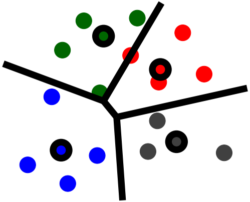
1. choose $c_1, \ldots, c_k \in \mathbb{R}^d$
2. Repeat
3. **partition X into $C_1, \ldots, C_k$**
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. Until stable



k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

1. choose $c_1, \ldots, c_k \in \mathbb{R}^d$
2. Repeat
3. partition X into $C_1, \ldots, C_k$
4. $\boldsymbol{c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x}$
5. Until stable

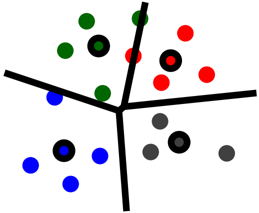


k-Means Method

k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

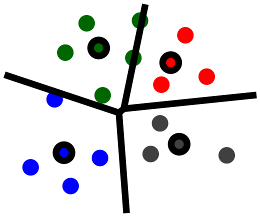
1. choose $c_1, \ldots, c_k \in \mathbb{R}^d$
2. Repeat
3. **partition X into $C_1, \ldots, C_k$**
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. Until stable



k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

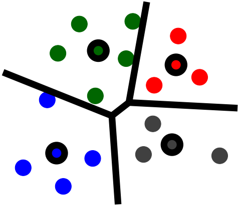
1. choose $c_1, \ldots, c_k \in \mathbb{R}^d$
2. Repeat
3. partition X into $C_1, \ldots, C_k$
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. Until stable



k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

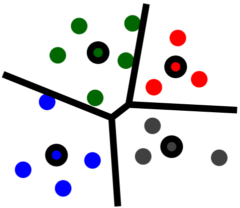
1. choose $c_1, \ldots, c_k \in \mathbb{R}^d$
2. Repeat
3. **partition X into $C_1, \ldots, C_k$**
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. Until stable



k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

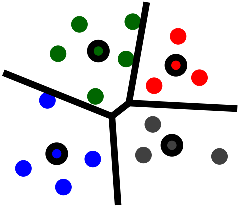
1. choose $c_1, \ldots, c_k \in \mathbb{R}^d$
2. Repeat
3. partition X into $C_1, \ldots, C_k$
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. Until stable



k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

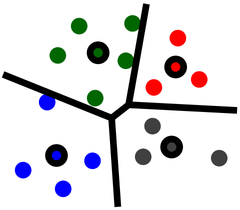
1. choose $c_1, \ldots, c_k \in \mathbb{R}^d$
2. Repeat
3. partition X into $C_1, \ldots, C_k$
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. **Until stable**



k-Means Method

Input: $X \subseteq \mathbb{R}^d$, k

1. choose $c_1, \ldots, c_k \in \mathbb{R}^d$
2. Repeat
3. partition X into $C_1, \ldots, C_k$
4. $c_i \leftarrow \frac{1}{|C_i|} \cdot \sum_{x \in C_i} x$
5. Until stable



"by far the most popular clustering algorithm used in scientific and industrial applications" (Berkhin 2002)

"in practice the number of iterations is generally much less than the number of points" (Duda et al. 2001)

k-Means Method – Theoretical Results

Running Time

Upper Bound: **At most $(k^2 n)^{kd}$ iterations.**

No clustering can occur twice.

Lower Bound: **At least $2^{\Omega(k)}$ iterations** for $d \geq 2$.

[Andrea Vattani (SoCG'09)]

k-Means Method – Theoretical Results

Running Time

Upper Bound: **At most $(k^2 n)^{kd}$ iterations.**

No clustering can occur twice.

Lower Bound: **At least $2^{\Omega(k)}$ iterations** for $d \geq 2$.

[Andrea Vattani (SoCG'09)]

Quality

Local optima can be arbitrarily bad.

k-Means Method – Theoretical Results

Running Time

Upper Bound: **At most $(k^2 n)^{kd}$ iterations.**

No clustering can occur twice.

Lower Bound: **At least $2^{\Omega(k)}$ iterations** for $d \geq 2$.

[Andrea Vattani (SoCG'09)]

Quality

Local optima can be arbitrarily bad.

$\Rightarrow$ Huge discrepancy between theory and practice.

(Focus of this talk: running time.)

Outline

Main Questions

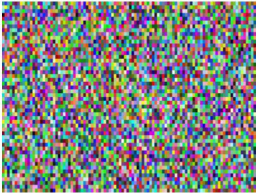
1. **Data Clustering**
   What is the *k*-means method?
2. **Smoothed Analysis**
   What can we do when worst case analysis is too pessimistic?
3. **Smoothed Analysis of *k*-Means Method**
   What is the smoothed complexity of the *k*-means method?
4. **Extensions and Conclusions**

Smoothed Analysis

- Worst-case analysis is too pessimistic.
  Typical instances are not adversarial.

Smoothed Analysis

- Worst-case analysis is too pessimistic.
  Typical instances are not adversarial.
- Average-case analysis unrealistic.
  What is the **right probability distribution**?



Smoothed Analysis

- Worst-case analysis is too pessimistic. Typical instances are not adversarial.
- Average-case analysis unrealistic. What is the **right probability distribution**?

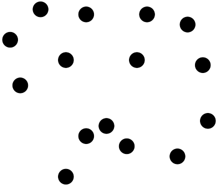
Smoothed Analysis

- Worst-case analysis is too pessimistic.
  Typical instances are not adversarial.
- Average-case analysis unrealistic.
  What is the **right probability distribution**?

Smoothed Analysis

Less powerful adversary:

1. Adversary chooses instance I.



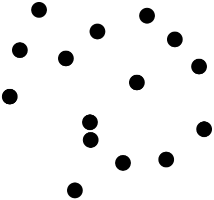
Smoothed Analysis

- Worst-case analysis is too pessimistic. Typical instances are not adversarial.
- Average-case analysis unrealistic. What is the **right probability distribution**?

Smoothed Analysis

Less powerful adversary:

1. Adversary chooses instance *I*.
2. Small amount of random noise is added to *I*.
   $\sigma =$ amount of noise



Smoothed Analysis

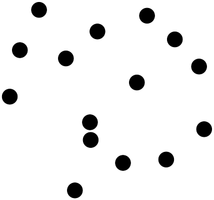
- Worst-case analysis is too pessimistic. Typical instances are not adversarial.
- Average-case analysis unrealistic. What is the **right probability distribution**?

Smoothed Analysis

Less powerful adversary:

1. Adversary chooses instance I.
2. Small amount of random noise is added to I.

   $\sigma =$ amount of noise



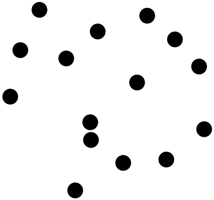
$$T(n, \sigma) = \max_{I, |I| = n} \mathbf{E}(\text{running time}(\mathrm{per}_\sigma(I)))$$

Smoothed Analysis

- Worst-case analysis is too pessimistic.
  Typical instances are not adversarial.
- Average-case analysis unrealistic.
  What is the **right probability distribution**?

Smoothed Analysis

Less powerful adversary:



1. Adversary chooses instance I.
2. Small amount of random noise is added to I.
   $\sigma =$ amount of noise

$$T(n,\sigma) = \max_{I,|I|=n} \mathbf{E}(\text{running time}(\mathrm{per}_\sigma(I)))$$

- models, e.g., measurement errors or numerical imprecision
- Smoothed compl. low $\Rightarrow$ bad performance unlikely in practice

Outline

Main Questions

1. **Data Clustering**
   What is the *k*-means method?
2. **Smoothed Analysis**
   What can we do when worst case analysis is too pessimistic?
3. **Smoothed Analysis of the *k*-Means Method**
   What is the smoothed complexity of the *k*-means method?
4. **Extensions and Conclusions**

Smoothed Analysis of k-Means

Model: Every point is perturbed by independent d-dimensional Gaussian with standard deviation σ.

$$T(n, \sigma) = \max_{X, |X|=n} \mathbf{E}(\#\mathrm{Iterations}(\mathrm{per}_\sigma(X)))$$

Smoothed Analysis of k-Means

Model: Every point is perturbed by independent d-dimensional Gaussian with standard deviation σ.

$$T(n, \sigma) = \max_{X, |X|=n} \mathbf{E}(\#\mathrm{Iterations}(\mathrm{per}_\sigma(X)))$$

Arthur, Vassilvitskii (FOCS 2006)

Smoothed number of iterations is **at most** $\mathrm{poly}(n^k, 1/\sigma)$.

Manthey, Röglin (SODA 2009)

Smoothed number of iterations is **at most**

- $\mathrm{poly}(n^{\sqrt{k}}, 1/\sigma)$
- $k^{kd} \cdot \mathrm{poly}(n, 1/\sigma)$

Smoothed Analysis of *k*-Means

Model: Every point is perturbed by independent *d*-dimensional Gaussian with standard deviation σ.

$$T(n, \sigma) = \max_{X, |X| = n} \mathbf{E}(\#\mathrm{Iterations}(\mathrm{per}_\sigma(X)))$$

Arthur, Vassilvitskii (FOCS 2006)

Smoothed number of iterations is **at most** $\mathrm{poly}(n^k, 1/\sigma)$.

Manthey, Röglin (SODA 2009)

Smoothed number of iterations is **at most**

- $\mathrm{poly}(n^{\sqrt{k}}, 1/\sigma)$
- $k^{kd} \cdot \mathrm{poly}(n, 1/\sigma)$

Our Result (FOCS 2009)

Smoothed number of iterations is $\mathrm{poly}(n, 1/\sigma)$.

General Approach

1. Initial Potential: After first iteration whp
   $\Phi = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2 = O(\mathrm{poly}(n))$

General Approach

1. Initial Potential: After first iteration whp
   $\Phi = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2 = O(\mathrm{poly}(n))$
2. Define smallest possible improvement:

$$\Delta = \min_{\text{iteration: } \mathcal{C} \to \text{succ}(\mathcal{C})} (\Phi(\mathcal{C}) - \Phi(\text{succ}(\mathcal{C}))) \ .$$

$\Rightarrow$ **at most $O(\mathrm{poly}(n)/\Delta)$ steps**.

General Approach

1. Initial Potential: After first iteration whp $\Phi = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2 = O(\mathrm{poly}(n))$
2. Define smallest possible improvement:

$$\Delta = \min_{\text{iteration: } \mathcal{C} \to \text{succ}(\mathcal{C})} \left(\Phi(\mathcal{C}) - \Phi(\text{succ}(\mathcal{C})) \right) .$$

$\Rightarrow$ **at most $O(\mathrm{poly}(n)/\Delta)$ steps**.

In the worst case: Δ arbitrarily small.

General Approach

1. Initial Potential: After first iteration whp
   $\Phi = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2 = O(\mathrm{poly}(n))$
2. Define smallest possible improvement:

$$\Delta = \min_{\text{iteration: } \mathcal{C} \to \text{succ}(\mathcal{C})} (\Phi(\mathcal{C}) - \Phi(\text{succ}(\mathcal{C}))) \ .$$

$\Rightarrow$ **at most $O(\mathrm{poly}(n)/\Delta)$ steps**.

In the worst case: Δ arbitrarily small.

Lemma

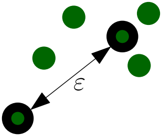
For $d \geq 2$, for every $X \subseteq [0, 1]^d$, in the model of smoothed analysis:

$$\mathbf{E}\left[\frac{1}{\Delta}\right] = \mathrm{poly}(n, 1/\sigma) \ .$$

$$\Rightarrow \max_{X, |X|=n} \mathbf{E}(\#\mathrm{Iterations}(\mathrm{per}_\sigma(X))) = \mathrm{poly}(n, 1/\sigma) \ .$$

When does the potential drop?

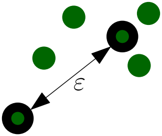
1) center moves by ε



$\Rightarrow$ improvement by ε^2

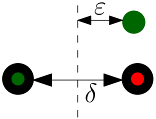
When does the potential drop?

1) center moves by ε



$\Rightarrow$ improvement by ε^2

2) point with distance ε to bisector changes assignment

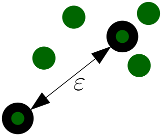


$\Rightarrow$ improvement by $2\varepsilon\delta$

When does the potential drop?

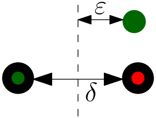
When does the potential drop?

1) center moves by ε



$\Rightarrow$ improvement by ε^2

2) point with distance ε to bisector changes assignment



$\Rightarrow$ improvement by $2\varepsilon\delta$

Goal: Show that in every iteration

1. either a center moves significantly
2. or a reassigned point is significantly far from bisector.

How large is Δ?

Configuration $\mathcal{C}$ is **ε-bad** if $\Phi(\mathcal{C}) - \Phi(\text{succ}(\mathcal{C})) \leq \varepsilon$.

Naive approach: Union Bound over all configurations.

$$\mathbf{Pr}\left[\exists \text{Configuration } \mathcal{C} : \mathcal{C} \text{ is } \varepsilon\text{-bad}\right] \leq \sum_{\text{Configuration } \mathcal{C}} \mathbf{Pr}\left[\mathcal{C} \text{ is } \varepsilon\text{-bad}\right]$$

How large is Δ?

Configuration $\mathcal{C}$ is **ε-bad** if $\Phi(\mathcal{C}) - \Phi(\text{succ}(\mathcal{C})) \leq \varepsilon$.
Naive approach: Union Bound over all configurations.

$$\mathbf{Pr}\left[\exists \text{Configuration } \mathcal{C} : \mathcal{C} \text{ is } \varepsilon\text{-bad}\right] \leq \sum_{\text{Configuration } \mathcal{C}} \mathbf{Pr}\left[\mathcal{C} \text{ is } \varepsilon\text{-bad}\right]$$

Problem: Too many configurations: k^n.

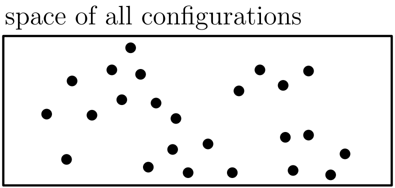
How large is Δ?

Configuration $\mathcal{C}$ is **ε-bad** if $\Phi(\mathcal{C}) - \Phi(\text{succ}(\mathcal{C})) \leq \varepsilon$.

Naive approach: Union Bound over all configurations.

$$\textbf{Pr}\left[\exists \text{Configuration } \mathcal{C} : \mathcal{C} \text{ is } \varepsilon\text{-bad}\right] \leq \sum_{\text{Configuration } \mathcal{C}} \textbf{Pr}\left[\mathcal{C} \text{ is } \varepsilon\text{-bad}\right]$$

Problem: Too many configurations: k^n.

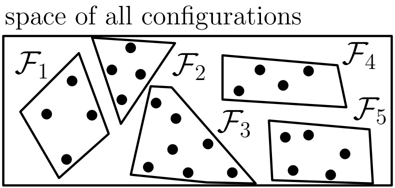


How large is Δ?

Configuration $\mathcal{C}$ is **ε-bad** if $\Phi(\mathcal{C}) - \Phi(\text{succ}(\mathcal{C})) \leq \varepsilon$.
Naive approach: Union Bound over all configurations.

$$\mathbf{Pr}\left[\exists \text{Configuration } \mathcal{C} : \mathcal{C} \text{ is } \varepsilon\text{-bad}\right] \leq \sum_{\text{Configuration } \mathcal{C}} \mathbf{Pr}\left[\mathcal{C} \text{ is } \varepsilon\text{-bad}\right]$$

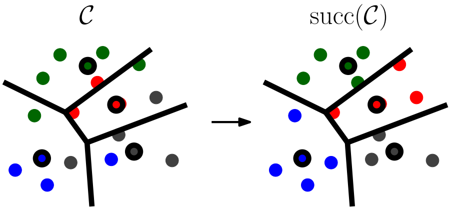
Problem: Too many configurations: k^n.

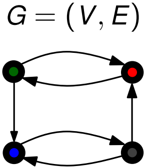


$$\mathbf{Pr}\left[\exists \text{Configuration } \mathcal{C} : \mathcal{C} \text{ is } \varepsilon\text{-bad}\right] \leq \sum_{i=1}^{5} \mathbf{Pr}\left[\mathcal{F}_i\right]$$

Transition Graph $G = (V, E)$:

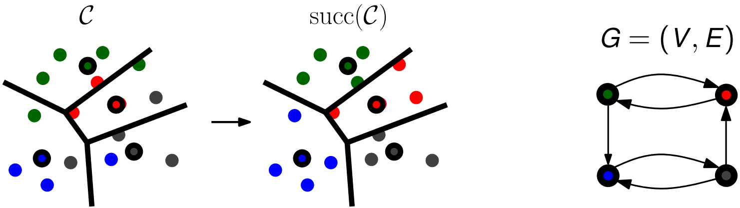
V: clusters E: labeled directed edge for each reassigned point





Transition Graph $G = (V, E)$:

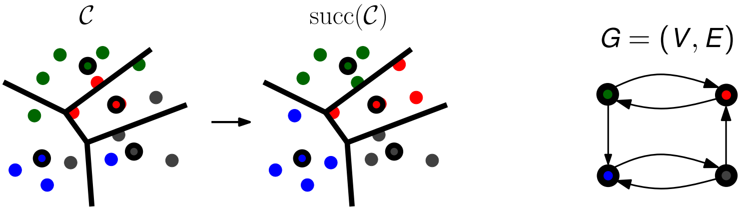
V: clusters E: labeled directed edge for each reassigned point



Approach: Union bound over different transition blueprints.

Transition Graph $G = (V, E)$:

V: clusters E: labeled directed edge for each reassigned point



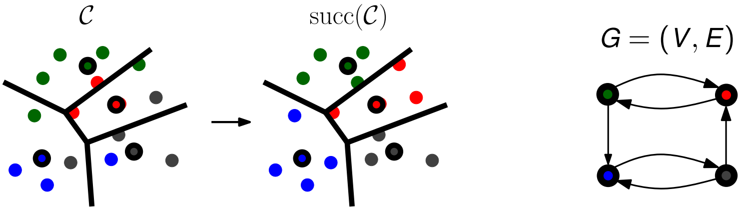
Approach: Union bound over different transition blueprints.

- First glance: Natural idea.
- Second glance: Not enough information.
  E.g.: No information about positions of centers and bisectors.

Transition Graph

Transition Graph $G = (V, E)$:

V: clusters $\quad$ E: labeled directed edge for each reassigned point



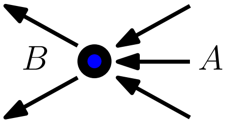
Approach: Union bound over different transition blueprints.

- First glance: Natural idea.
- Second glance: Not enough information.
  E.g.: No information about positions of centers and bisectors.
- Third glance: **Enough information!**

Approximate Centers

Goal: Show that in every iteration

1. either a center moves significantly
2. or a reassigned point is significantly far from bisector.



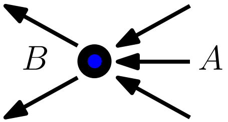
$$C \to (C \cup A) \setminus B$$

$$\|\mathrm{cm}((C \cup A) \setminus B) - \mathrm{cm}(C)\| \geq \frac{1}{n^2} \left\| \mathrm{cm}(C) - \overbrace{\left(\frac{|B|\,\mathrm{cm}(B) - |A|\,\mathrm{cm}(A)}{|B| - |A|} \right)}^{\mathrm{approx}(A,B)} \right\|$$

Approximate Centers

Goal: Show that in every iteration

1. either a center moves significantly
2. or a reassigned point is significantly far from bisector.



$$C \to (C \cup A) \setminus B$$

$$\|\text{cm}((C\cup A)\setminus B) - \text{cm}(C)\| \geq \frac{1}{n^2}\left\| \text{cm}(C) - \overbrace{\left(\frac{|B|\,\text{cm}(B) - |A|\,\text{cm}(A)}{|B| - |A|}\right)}^{\mathrm{approx}(A,B)} \right\|$$

small potential drop $\Rightarrow$ $\text{cm}(C)$ must be close to $\mathrm{approx}(A, B)$

Proof of Main Theorem

Theorem

Smoothed number of iterations is $\mathbf{poly}(n, 1/\sigma)$.

Proof of Main Theorem

Theorem

Smoothed number of iterations is $\mathbf{poly}(n, 1/\sigma)$.

- number of blueprints with m edges: $(k^2 n)^m$.

Proof of Main Theorem

Theorem

Smoothed number of iterations is $\mathbf{poly}(n, 1/\sigma)$.

- number of blueprints with m edges: $(k^2 n)^m$.
- probability that a fixed data point has distance at most ε from its approximate bisector: $\leq \varepsilon/\sigma$.

Proof of Main Theorem

Theorem

Smoothed number of iterations is $\mathbf{poly}(n, 1/\sigma)$.

- number of blueprints with m edges: $(k^2 n)^m$.
- probability that a fixed data point has distance at most ε from its approximate bisector: $\leq \varepsilon/\sigma$.
- probability that all m data points are ε-close to their approximate bisectors: $\leq (\varepsilon/\sigma)^m$

Proof of Main Theorem

Theorem

Smoothed number of iterations is $\mathbf{poly}(\boldsymbol{n}, \mathbf{1}/\boldsymbol{\sigma})$.

- number of blueprints with m edges: $(k^2 n)^m$.
- probability that a fixed data point has distance at most ε from its approximate bisector: $\leq \varepsilon/\sigma$.
- probability that all m data points are ε-close to their approximate bisectors: $\leq (\varepsilon/\sigma)^m$
- Union Bound:

$$\mathbf{Pr}\left[\exists\varepsilon\text{-bad blueprint}\right] \leq (k^2 n)^m \cdot (\varepsilon/\sigma)^m$$

Very unlikely for $\varepsilon = 1/\mathrm{poly}(n, 1/\sigma)$. □

Proof of Main Theorem

Theorem

Smoothed number of iterations is $\mathbf{poly}(\boldsymbol{n}, \mathbf{1}/\boldsymbol{\sigma})$.

- number of blueprints with m edges: $(k^2 n)^m$.
- probability that a fixed data point has distance at most ε from its approximate bisector: $\leq \varepsilon/\sigma$.
- probability that all m data points are ε-close to their approximate bisectors: $\leq (\varepsilon/\sigma)^m$
- Union Bound:

$$\mathbf{Pr}\left[\exists\varepsilon\text{-bad blueprint}\right] \leq (k^2 n)^m \cdot (\varepsilon/\sigma)^m$$

Very unlikely for $\varepsilon = 1/\mathrm{poly}(n, 1/\sigma)$. □

Technical Difficulties: Data points are not independent from approx. bisectors, approximate centers not defined for balanced clusters, blueprints must have enough edges, . . .

Outline

Main Questions

1. **Data Clustering**
   What is the *k*-means method?
2. **Smoothed Analysis**
   What can we do when worst case analysis is too pessimistic?
3. **Smoothed Analysis of *k*-Means Method**
   What is the smoothed complexity of the *k*-means method?
4. **Extensions and Conclusions**

Kullback-Leibler Divergence

Text Classification and Bag-of-Words Model:

- S set of all words
- count words and normalize:
  prob. distribution $p\colon S \to [0,1]$

Kullback-Leibler Divergence

Text Classification and Bag-of-Words Model:

- S set of all words
- count words and normalize:
  prob. distribution $p\colon S \to [0,1]$

Kullback-Leibler divergence (relative entropy):

$$\mathrm{KLD}(p,q) = \sum_{i=1}^{d} p_i \log\left(\frac{p_i}{q_i}\right)$$

$=$ number of bits to encode p with Huffman code for q
$-$ number of bits to encode p with Huffman code for p

Bregman Divergences

Bregman divergences are distance measures that generalize squared Euclidean distances and the Kullback-Leibler divergence.

Ackermann, Blömer, Sohler (SODA 2008)

Approximation Scheme for special cases of Bregman divergences (e.g., Kullback-Leibler divergence).

Bregman Divergences

Bregman divergences are distance measures that generalize squared Euclidean distances and the Kullback-Leibler divergence.

Ackermann, Blömer, Sohler (SODA 2008)

Approximation Scheme for special cases of Bregman divergences (e.g., Kullback-Leibler divergence).

Manthey, Röglin (ISAAC 2009)

For any well-behaved Bregman divergence:
Smoothed number of iterations is **at most**

- $\mathrm{poly}(n^{\sqrt{k}}, 1/\sigma)$
- $k^{kd} \cdot \mathrm{poly}(n, 1/\sigma)$
- $\mathrm{poly}(n, 1/\sigma)$ if $d, k = O(\sqrt{\log n / \log \log n})$ (or $d = 1$)

Bregman Divergences

Bregman divergences are distance measures that generalize squared Euclidean distances and the Kullback-Leibler divergence.

Ackermann, Blömer, Sohler (SODA 2008)

Approximation Scheme for special cases of Bregman divergences (e.g., Kullback-Leibler divergence).

Manthey, Röglin (ISAAC 2009)

For any well-behaved Bregman divergence:
Smoothed number of iterations is **at most**

- $\mathrm{poly}(n^{\sqrt{k}}, 1/\sigma)$
- $k^{kd} \cdot \mathrm{poly}(n, 1/\sigma)$
- $\mathrm{poly}(n, 1/\sigma)$ if $d, k = O(\sqrt{\log n / \log\log n})$ (or $d = 1$)

Polynomial bound does not extend as it uses special properties of Gaussian perturbations.

Further Results and Open Questions

Summary:

- Worst-case instances are often fragile.
- Smoothed Analysis often leads to better understanding of observed practical behavior.

Further Results and Open Questions

Summary:

- Worst-case instances are often fragile.
- Smoothed Analysis often leads to better understanding of observed practical behavior.

Future Research:

- improve exponents for k-means (currently $\approx n^{30}$)
  better understanding of dynamics seems necessary for this
- explanation for good approximation ratio
- better analysis of Bregman divergences
- more systematic theory of smoothed local search
- Are all local search problems in PLS easy in smoothed analysis?

Thank you for your attention!

Questions?