K-Clustering
Coresets

Sagi Hed 9th Dec 2008

- "Bi-Criteria Linear-Time Approximations for Generalized k-Mean/Median/Center" (Danny Feldman, Amos Fiat, Micha Sharir, Danny Segev)
- "Smaller Coresets for k-Median an k-Means Clustering" (Sariel Har-Peled, Akash Kushal)
Coresets Approach

• **Not exactly sublinear time algorithm!**

• Instead of working on input, work on an alternative, very small, but indistinguishable input

• Create small coreset in streaming algorithm

• Coreset will allow infinite various queries in sub-linear time..
Streaming

What is A Streaming Algorithm?
• Data continuously generated and total volume is extremely large
• Algorithm examines arriving items once, and discards
• Update internal state fast per new item ($O(1)$ to $poly \ log(n)$)
• Use small space ($poly \ log(n)$)

In other words, can run whenever new data arrives, even if $n$ is too large for memory.
What is a Coreset?
A subset of input, such that we can get a good approximation to
the original input by solving the problem directly on the coreset.

Since the coreset is very small, typically poly-logarithmic, we can
do this even with exhaustive search.
K-Clustering

\[ P \equiv \text{set of } n \text{ points in } \mathbb{R}^d \]
\[ C \equiv \text{set of } k \text{ points in } \mathbb{R}^d \]
\[ \text{dist}(p, C) \equiv \text{distance from } p \text{ to the nearest point in } C \]

\[ \text{Cost}_{P}(C) \equiv \sum_{p \in P} \text{dist}(p, C) \quad \text{K-Median Clustering} \]
\[ \text{Cost}_{P}(C) \equiv \sum_{p \in P} [\text{dist}(p, C)]^2 \quad \text{K-Means Clustering} \]
\[ \text{Cost}_{P}(C) \equiv \max_{p \in P} \{ \text{dist}(p, C) \} \quad \text{K-Center Clustering} \]
K-Clustering

\[ P \equiv \text{set of } n \text{ points in } \mathbb{R}^d \]

- **K-Center/Median/Mean Clustering**
  Find \( C, |C| = k \), s.t. \( \text{Cost}_P(C) \) is minimal…

- **K-Center/Median/Mean Queries**
  Given \( C \), what is \( \text{Cost}_P(C) \)?

  *Exact K-Clustering is NP-hard… (exponential in \( k \))*
(k, ε) Coreset

(k, ε)-coreset ≡ set S of points in \( R^d \) where for every C, |C| = k

\[(1 - \varepsilon) \cdot \text{Cost}_P(C) \leq \text{Cost}_S(C) \leq (1 + \varepsilon) \cdot \text{Cost}_P(C)\]

But \( \text{OPT}(S) \neq \text{OPT}(P) \)...

\[\text{Cost}_S(\text{OPT}(S)) \leq \text{Cost}_S(\text{OPT}(P)) \leq (1 + \varepsilon) \cdot \text{Cost}_P(\text{OPT}(P))\]
Our Goal

- Streaming algorithm for \((k, \varepsilon)\)-coreset with \(O(k\varepsilon^{-d} \cdot \log(n))\) points.

- With this small coreset we can answer k-queries or find a k-clustering in sub-linear time, without ever reading the original input again.

- We assume \(k\) and \(d\) is constant.

Answer queries on coreset with exhaustive search –

- Answer k-queries in \(O(k^2 \cdot d \cdot \varepsilon^{-d} \cdot \log(n))\) time.

- Find k-clustering in roughly \(O( [k\varepsilon^{-d} \cdot \log(n)]^{kd+1} )\) time (number of Voronoi partitions is bounded).
K-Clustering Coresets

Order Of Business:

1. Coreset Streaming Reduction
2. Bi-Criteria K-Clustering Approximation
3. K-Center Coresets Using Bi-Criteria K-Clustering
4. K-Means/Median Coresets Using Hand Waving
Coreset Streaming Reduction

Input
A \((k, \varepsilon)\)-coreset creation algorithm for k-clustering, creating coresets of size \(O(k(1/\varepsilon)^d)\) in \(O(d \cdot n \cdot k)\)

Output
A **Streaming** \((k, \varepsilon)\)-coreset creation algorithm for k-center clustering, creating coresets of size \(O(k(1/\varepsilon)^d \log(n))\), that works in time \(O(polylog(k, 1/\varepsilon))\) per insertion
Lemma 1
C₁ is a \((k, \varepsilon)\)-coreset for \(P₁\)
C₂ is a \((k, \varepsilon)\)-coreset for \(P₂\)
\(\Rightarrow C₁ \cup C₂\) is a \((k, \varepsilon)\)-coreset for \(P₁ \cup P₂\)

Lemma 2
C₁ is a \((k, \varepsilon₁)\)-coreset for \(P\)
C₂ is a \((k, \varepsilon₂)\)-coreset for \(C₁\)
\(\Rightarrow C₂\) is a \((k, O(\varepsilon₁+\varepsilon₂))\)-coreset for \(P\)
Coreset Streaming

- Operate coreset on existing coresets
- Maintain $\log(n)$ coresets at a time

$p_i = \epsilon / [c(j+1)^2]$

$\prod(1+p_i) \leq 1 + \frac{\epsilon}{2}$ for a large enough $c$

$\Rightarrow \bigcup Q_i$ is a $(k, \frac{\epsilon}{2})$-coreset

$Q_3 = \bullet$

$Q_2 =$

$Q_1 =$

$O(k(1/\epsilon)^d)$

$Q_3 =$

$(k,p_3)$-coreset

$(k,p_2)$-coreset

$(k,p_1)$-coreset

$\ldots$
Coreset Streaming

Space Complexity
Space for algorithm is –
\[ \sum_{1 \leq j \leq \log(n)} O\left(\left[\frac{c(j+1)^{2d}}{\varepsilon^d}\right]\right) = O(\text{polylog}(n)) \]

Coreset Size
Build \( R_i = (k, \varepsilon/6)\)-coreset from every \( Q_i \)
\( U R_i \) = still a \((k, \varepsilon)\)-coreset for all points
\[ |U R_i| = \sum_{1 \leq j \leq \log(n)} O\left(\frac{(6/\varepsilon)^d}{\varepsilon^d}\right) = O\left(\frac{(1/\varepsilon)^d \log(n)}{\varepsilon^d}\right) \]
=> actual coreset (for exhaustive search) is smaller

Time Complexity
Requires an amortized analysis similar to that of a binary counter
\[ O(\text{polylog}(k, 1/\varepsilon)) \text{ per insertion} \]
Bi-Criteria K-Clustering

Clustering Approximation
C is a $\beta$-approximation for clustering $\equiv$
$\text{Cost}_P(C) \leq \beta \cdot \text{Cost}_P(OPT)$

Goal
With high probability obtain a
2-approximation clustering C
$|C| = O(k \cdot \log^2(n))$
*in time $O(d \cdot n \cdot k \cdot \log(n))$*

*With meticulous calculations can be improved to
$|C| = O(k \cdot \log(n) \cdot \log\log(n))$
*and time $O(d \cdot n \cdot k)$*
Algorithm
1. \( F = \emptyset, t = 1 \)
2. Until we cover all points:
   1. \( F_t = \text{Sample } O(k \cdot \log(n)) \) points at random
   2. \( F = F \cup F_t \)
   3. \( P_t = \text{the } \lfloor |P|/2 \rfloor \text{ points from } P \) that are closest to \( F_t \)
   4. Remove \( P_t \) from \( P \)
   5. \( t++ \)
3. Output \( F \)
**Bi-Criteria K-Clustering**

**Algorithm**
1. While (P is not empty, t++):
   1. \( F = F \cup F_t \) (≡ Sample \( O(k \cdot \log(n)) \) points at random)
   2. Remove \( P_t = \) the \( |P|/2 \) points from P that are closest to \( F_t \)

**Analysis**

Running Time – \( O(n \cdot d \cdot k \cdot \log(n)) \)

\( |F| = O(k \cdot \log^2(n)) \)
Bi-Criteria K-Clustering

Correctness

$F^* \equiv$ optimal centers for clustering, $|F| = k$

$p$ is “bad” for $F_t \equiv \text{dist}(p, F_t) > 2 \cdot \text{dist}(p, F^*)$
$p$ is “good” for $F_t \equiv \text{dist}(p, F_t) \leq 2 \cdot \text{dist}(p, F^*)$
Bi-Criteria K-Clustering

Lemma 1

With high probability, number of bad points for $F_t \leq |P_t| / 8$
Lemma 2
With high probability, for every bad point for $F_t$ which is discarded with $P_t$, we can match a distinct good point for $F_t$ which is outside $P_t$. In fact, this point is discarded with $P_{t+1}$.

Proof of Lemma 2

$\#\{\text{points in } P_{t+1} \text{ good for } F_t\} \geq |P_{t+1}| - \#\{\text{points bad for } F_t\}$

Using Lemma 1,
$\geq |P_t|/2 - |P_t|/8 \geq 3/8 |P_t|$

$\geq \#\{\text{points bad for } F_t\}$

Since the point is discarded with $P_{t+1}$, it is distinct.
**Correctness**
For any bad point \( b \) for some \( F_t \), let \( g \equiv \) its matching distinct good point –
\[
\text{dist}(b, F) \leq \text{dist}(b, F_t) \leq \text{dist}(g, F_t) \leq 2 \cdot \text{dist}(g, F^*)
\]

**K-Median Clustering**
\[
\text{Cost}_P(F) = \sum_{p \in P} \text{dist}(p, F)
\]
\[
\leq \sum_g \text{dist}(g, F) + \sum_b \text{dist}(b, F)
\]
\[
\leq \sum_g 2 \cdot \text{dist}(g, F^*) + \sum_g 2 \cdot \text{dist}(g, F^*)
\]
\[
\leq 2 \sum_g \text{dist}(g, F^*) \leq 2 \cdot \text{Cost}_P(F^*) \]

**K-Means/Center Clustering**
Same proof…
Bi-Criteria K-Clustering

**Proof of Lemma 1**

During iteration $t$ –

$f^*_i \equiv$ the $i$th center in $F^*$

$Q_i \equiv |P_t|/8k$ points closest to $f_i$

Probability that sampled point is not in $Q_i = (1 - 1/8k)$

$\Rightarrow$ Probability that none of the sampled points are in $Q_i = (1-1/8k)^{8k\log(n)\cdot c}$

$\leq e^{-c\cdot \log(n)} = 1/n^c$

Probability that at least one of $Q_i$ has no sampled points in at least one of the $\log(n)$ iterations is, using union bound

$\leq k \cdot \log(n)/n^c \leq 1/n^{c-2}$

$\Rightarrow$ Probability that all $Q_i$ have sampled points in all iterations $\geq 1 - 1/n^{c-2}$

For a large enough $n$, this expression is arbitrarily large
Proof of Lemma 1

$q \equiv$ point outside the $|P_i|/8k$ closest of any $f_i$
$f^* \equiv q$'s closest point in $F^*$
$p \equiv$ a sampled point closest to $f^*$ (within the $|P_i|/8k$ closest)

$dist(q,F_i) \leq dist(q,p) \leq dist(q,f^*) + dist(f^*,p) \leq 2 \cdot dist(q,f^*) = 2 \cdot dist(q,F^*)$

$\Rightarrow p$ is a “good“ point!

$\Rightarrow$ at most $|P_i|/8$ “bad” points!
Bi-Criteria K-Clustering

**Careful**

Can amplify the probability more by making attempts and choosing $F$ with lowest $Cost_P(F)$.

Need to be careful with probabilities and streaming reduction.

- Generalizes to $j$-flats in $d$ dimensions instead of points (open problem to construct coresets for these)
K-Center Clustering
Coreset

Goal

Obtain a \((k, \varepsilon)\) center-clustering coreset \(S\)

\[(1-\varepsilon)\text{Cost}_P(C) \leq \text{Cost}_S(C) \leq (1+\varepsilon)\text{Cost}_P(C)\]

\(\text{Cost}_A(B) \equiv \max_{p \in A} \{ \text{dist}(p, B) \}\)

\(|S| = O(k^{\sqrt{d}/\varepsilon^d})\)

- Can show it's impossible to not be exponential in \(d\).
- Just for \(k\)-clustering it's possible to get a coreset of size \(O(k\varepsilon^{-2})\)
K-Center Clustering

Coreset

**Construction**
Use K\(\log(n)\)-Clustering on \(P\) to create \(C\)

\(|C| = O(k\log^2(n))\)

\(Cost_P(C) \leq 8 \cdot Cost_P(OPT)\)
K-Center Clustering

Coreset

Construction
Build $2 \cdot \text{Cost}_P(C) \times 2 \cdot \text{Cost}_P(C) \times \ldots \times d$-dimensional grid around points in $C$.
Divide to “squares” $\varepsilon \text{Cost}_P(C)/(8\sqrt{d}) \times \varepsilon \text{Cost}_P(C)/(8\sqrt{d}) \times \ldots$
K-Center Clustering

**Coreset Construction**

\[
Cost_P(C) = \max_{p \in P} \{ \text{dist}(p, C) \}
\]

=> all points contained in the grids

Running time is \(O(n)\)

\[
2Cost_P(C) = \varepsilon/(8\sqrt{d}) \cdot Cost_P(C)
\]
K-Center Clustering

Coreset

Construction
Coreset = pick one representative from every square
$|\text{Coreset}| = O\left( \sqrt{d/\varepsilon}^d \cdot k \cdot \log^2(n) \right)$
K-Center Clustering

Coreset

Correctness

$S \equiv \text{coreset}, \ p \equiv \text{point maximizing } \text{dist}(p,A), \ q \equiv p's \ representative$

$\text{Cost}_P(A) = \text{dist}(p,A) \leq \text{dist}(q, A) + (\sqrt{d})(\epsilon/8\sqrt{d})\text{Cost}_P(C)$

$\leq \text{Cost}_S(A) + \epsilon\text{Cost}_P(OPT) \leq \text{Cost}_S(A) + \epsilon\text{Cost}_P(A)$

$\implies (1-\epsilon)\text{Cost}_P(A) \leq \text{Cost}_S(A)$

2$\text{Cost}_P(C)$

$\epsilon/(8\sqrt{d}) \cdot \text{Cost}_P(C)$
K-Center Clustering

Coreset

Correctness

\[ \text{Cost}_p(A) = \text{dist}(p, A) \geq \text{dist}(q, A) - (\sqrt{d})(\varepsilon/8\sqrt{d})\text{Cost}_p(C) \]
\[ \geq \text{Cost}_s(A) - \varepsilon\text{Cost}_p(OPT) \geq \text{Cost}_s(A) - \varepsilon\text{Cost}_p(A) \]

\[ \Rightarrow (1+\varepsilon)\text{Cost}_p(A) \geq \text{Cost}_s(A) \]

\[ 2\text{Cost}_p(C) \]
\[ \varepsilon/(8\sqrt{d}) \cdot \text{Cost}_p(C) \]
K-Center Clustering

Coreset

\[ |\text{Coreset}| = O\left( \sqrt{d/\epsilon}^d \cdot k \cdot \log^2(n) \right) \]

*How to get rid of the \( \log^2(n) \)?*
It was inherited from number of cluster centers in the bi-criteria clustering approximation…

*Find a k-center clustering \( (1+\epsilon) \) approximation using the current coreset, then reassign the clustering (this time with exactly k centers) into the coreset creation…*
K-Means/Median Clustering Coreset

Can also create a \((k,\varepsilon)\) median/means-clustering coreset \(S\)

\[(1-\varepsilon)\text{Cost}_P(C) \leq \text{Cost}_S(C) \leq (1+\varepsilon)\text{Cost}_P(C)\]

\[\text{Cost}_A(B) \equiv \sum_{p \in A} \text{dist}(p,B)\]
\[\text{or}\]
\[\text{Cost}_A(B) \equiv \sum_{p \in A} [\text{dist}(p,B)]^2\]

\[|S| = O(k^2\varepsilon^{-d})\]

- Was improved to \(|S| = O(poly(d,k,1/\varepsilon))!!\)
- Again, just for \(k\)-clustering its possible to get a coreset of size independent of \(n\) or \(d\)
K-Means/Median Clustering Coreset

**Construction**
In K-Means/Median the coresets have to be weighted…

Similarly to k-centers, start with a bi-criteria clustering and create squares around each center. The squares are growing by factor 2.

Sample points from every square, but give a weight according to the number of points in that square.
Questions ??
Exercises

1. Consider the case of points in $\mathbb{R}^2$. Suppose you have an $O(\log(n))$ size coreset. Figure out how to find its optimal clustering using exhaustive search in $O(\log^{4k}(n))$ time.

2. Consider Bi-Criteria K-Clustering for the case of lines in $\mathbb{R}^2$. i.e. we want to find a set $C$ of $O(k^c \log^c(n))$ lines s.t. - $\text{Cost}_P(C) \leq c' \cdot \text{Cost}_P(\text{OPT})$
where distances are now from points to their nearest line.

Hint: The algorithm stays almost the same. When sampling points think how you should transform them into lines…