Clustering in Data Streams: Improving BIRCH

Project: Practical Theory for Clustering Algorithms

Johannes Blömer, Daniel Kuntze, Kathrin Bujna (Paderborn)
Christian Sohler, Melanie Schmidt (Dortmund)

Hendrik Fichtenberger, Marc Gillé, Chris Schwiegelshohn
Clustering: Grouping of similar objects according to some distance measure

The $k$-means Problem
Clustering: Grouping of similar objects according to some distance measure

The k-means Problem

- Given a point set $P \subseteq \mathbb{R}^d$, compute a set $C \subseteq \mathbb{R}^d$ with $|C| = k$ centers which minimizes $\text{cost}(P, C) = \sum_{p \in P} \min_{c \in C} ||c - p||^2$, the sum of the squared distances.
Clustering: Grouping of similar objects according to some distance measure

The $k$-means Problem
- Given a point set $P \subseteq \mathbb{R}^d$,
- compute a set $C \subseteq \mathbb{R}^d$ with $|C| = k$ centers
Clustering: Grouping of similar objects according to some distance measure

The $k$-means Problem

- Given a point set $P \subseteq \mathbb{R}^d$,
- compute a set $C \subseteq \mathbb{R}^d$ with $|C| = k$ centers
- which minimizes
  \[
  \text{cost}(P, C) = \sum_{p \in P} \min_{c \in C} ||c - p||^2,
  \]
  the sum of the squared distances.
In Data Streams:
- Points arrive in a stream one after the other
- arbitrary order
- only one pass over the data allowed
- limited storage capacity
... in Data Streams:
- Points arrive in a stream one after the other
- arbitrary order
- only one pass over the data allowed
- limited storage capacity

In Practice: BIRCH as a well-known heuristic
In Theory: Coreset Theory
Data Stream Clustering in Practice and Theory

**Clustering Feature**

**Fact**

The sum of the squared distances satisfies the equation

\[
\sum_{p \in P} \|p - z\|^2 = \sum_{p \in P} \|p - \mu\|^2 + |P| \|\mu - z\|^2
\]

where \(\mu\) is the centroid of \(P\).
The sum of the squared distances satisfies the equation

$$\sum_{p \in P} ||p-z||^2 = \sum_{p \in P} ||p-\mu||^2 + |P|||\mu-z||^2$$

where $\mu$ is the centroid of $P$. 

---

Clustering Feature

Fact

The sum of the squared distances satisfies the equation

$$\sum_{p \in P} ||p-z||^2 = \sum_{p \in P} ||p-\mu||^2 + |P|||\mu-z||^2$$

where $\mu$ is the centroid of $P$. 

---

Clustering in Data Streams: Improving BIRCH
The sum of the squared distances satisfies the equation

$$\sum_{p \in P} ||p-z||^2 = \sum_{p \in P} ||p-\mu||^2 + |P|||\mu-z||^2$$

where $\mu$ is the centroid of $P$. 
Clustering Feature

Fact

The sum of the squared distances satisfies the equation

$$\sum_{p \in P} ||p - z||^2 = \sum_{p \in P} ||p - \mu||^2 + |P||\mu - z||^2$$

where $\mu$ is the centroid of $P$. 
The sum of the squared distances satisfies the equation

$$\sum_{p \in P} ||p - z||^2 = \sum_{p \in P} ||p - \mu||^2 + |P||\mu - z||^2$$

where $\mu$ is the centroid of $P$. 

Clustering in Data Streams: Improving BIRCH
Clustering in Data Streams: Improving BIRCH

**Fact**

The sum of the squared distances satisfies the equation

\[
\sum_{p \in P} ||p - z||^2 = \sum_{p \in P} ||p - \mu||^2 + ||P|| ||\mu - z||^2
\]

where \( \mu \) is the centroid of \( P \).

- Simply store \( |P|, \sum_{p \in P} p \) and \( \sum_{p \in P} ||p||^2 \)
- \( \sum_{p \in P} ||p||^2 = \sum_{p \in P} ||p - \mu||^2 + ||P|| ||\mu||^2 \)
- \( \sum_{p \in P} ||p - z||^2 = \sum_{p \in P} ||p||^2 - ||P|| ||\mu||^2 + ||P|| ||\mu - z||^2 \)
BIRCH

- uses Clustering Features
BIRCH

- uses Clustering Features
- CFs are stored in a CF Tree, nodes contain the CFs
BIRCH
- uses Clustering Features
- CFs are stored in a CF Tree, nodes contain the CFs

Insertion of a new point
When a new point is added to the CF Tree
BIRCH

- uses Clustering Features
- CFs are stored in a CF Tree, nodes contain the CFs

Insertion of a new point

When a new point is added to the CF Tree

- BIRCH searches for the ‘closest’ CF according to

\[
\sum_{q \in (S \cup \{p\})} \left( q - \frac{\sum_{q \in (S \cup \{p\})} q}{|S|+1} \right)^2 - \sum_{q \in S} \left( q - \frac{\sum_{q \in S} q}{|S|} \right)^2
\]
**BIRCH**
- uses Clustering Features
- CFs are stored in a CF Tree, nodes contain the CFs

**Insertion of a new point**

When a new point is added to the CF Tree
- BIRCH searches for the ‘closest’ CF according to
  \[
  \sum_{q \in (S \cup \{p\})} (q - \frac{\sum_{q \in (S \cup \{p\})} q}{|S|+1})^2 - \sum_{q \in S} (q - \frac{\sum_{q \in S} q}{|S|})^2
  \]
- \(p\) is added to CF representing subset \(S^*\) if
  \[
  \sqrt{\frac{\sum_{p \in (S^* \cup \{p\})} (q - \mu_S)^2}{|S|+1}} \leq T \text{ for a given threshold}
  \]
Coreset Theory

Coresets

Given a set of points $P$, a weighted subset $S \subset P$ is a $(k, \epsilon)$-coreset if for all sets $C$ of $k$ centers it holds

$$|\text{cost}_w(S, C) - \text{cost}(P, C)| \leq \epsilon \text{cost}(P, C)$$

where $\text{cost}_w(S, C) = \sum_{p \in S} \min_{c \in C} w(p) ||p - c||^2$. 

Data Stream Clustering in Practice and Theory
Coreset Theory

Coresets

Given a set of points \( P \), a weighted subset \( S \subset P \) is a \((k, \epsilon)\)-coreset if for all sets \( C \) of \( k \) centers it holds

\[
|\text{cost}_w(S, C) - \text{cost}(P, C)| \leq \epsilon \text{cost}(P, C)
\]

where \( \text{cost}_w(S, C) = \sum_{p \in S} \min_{c \in C} w(p) ||p - c||^2 \).
Coreset Theory

Coresets

Given a set of points $P$, a weighted subset $S \subseteq P$ is a $(k, \epsilon)$-coreset if for all sets $C$ of $k$ centers it holds

$$|\text{cost}_w(S, C) - \text{cost}(P, C)| \leq \epsilon \text{cost}(P, C)$$

where $\text{cost}_w(S, C) = \sum_{p \in S} \min_{c \in C} w(p) ||p - c||^2$. 
Coreset Theory

Coresets

Given a set of points $P$, a weighted subset $S \subset P$ is a $(k, \epsilon)$-coreset if for all sets $C$ of $k$ centers it holds

$$|\text{cost}_w(S, C) - \text{cost}(P, C)| \leq \epsilon \text{cost}(P, C)$$

where $\text{cost}_w(S, C) = \sum_{p \in S} \min_{c \in C} w(p) ||p - c||^2$. 
Coreset Theory

Coresets

Given a set of points $P$, a weighted subset $S \subset P$ is a $(k, \epsilon)$-coreset if for all sets $C$ of $k$ centers it holds

$$|\text{cost}_w(S, C) - \text{cost}(P, C)| \leq \epsilon \text{ cost}(P, C)$$

where $\text{cost}_w(S, C) = \sum_{p \in S} \min_{c \in C} w(p) ||p - c||^2$. 
Coreset Theory

Coresets

Given a set of points $P$, a weighted subset $S \subseteq P$ is a $(k, \epsilon)$-coreset if for all sets $C$ of $k$ centers it holds

$$|\text{cost}_w(S, C) - \text{cost}(P, C)| \leq \epsilon \text{cost}(P, C)$$

where $\text{cost}_w(S, C) = \sum_{p \in S} \min_{c \in C} w(p) ||p - c||^2$. 
Coreset constructions

'01: Agarwal, Har-Peled and Varadarajan: Coreset concept

'02: Bădoiu, Har-Peled and Indyk: First coreset construction for clustering problems

'04: Har-Peled and Mazumdar, Coreset of size $O(k\epsilon^{-d} \log n)$, maintainable in data streams

'05: Frahling and Sohler: Coreset of size $O(k\epsilon^{-d} \log n)$, insertion-deletion data streams

'06: Chen: Coresets for metric and Euclidean $k$-median and $k$-means, polynomial in $d,n$ and $\epsilon^{-1}$

'07: Feldman, Monemizadeh, Sohler: weak coresets, poly($k$, $\epsilon^{-1}$)

'10: Langberg, Schulman: $\tilde{O}(d^2k^3/\epsilon^2)$

'11: Feldman, Langberg: $O(dk/\epsilon^2)$

Merge & Reduce: Coreset Construction $\rightsquigarrow$ Streaming Algorithms.
Data Stream Clustering in Practice and Theory

Clustering in Data Streams: Improving BIRCH
StreamKM++

- also an outcome of this SPP
- practical $k$-means streaming algorithm
- computes a coreset, moderate storage requirement
- better solutions than BIRCH, but slower
StreamKM++

- also an outcome of this SPP
- practical $k$-means streaming algorithm
- computes a coreset, moderate storage requirement
- better solutions than BIRCH, but slower

Motivations to Improve BIRCH

- Analyzable BIRCH is valuable
- Might outperform both StreamKM++ and BIRCH
- Hope of keeping good practical properties
When does BIRCH perform badly?
When does BIRCH perform badly?

Lemma
Depending on the threshold $T$, BIRCH either needs $\Omega(|P|d)$ CFs or it computes no constant factor approximation.

(For a generalized example)
When does BIRCH perform badly?

Lemma

Depending on the threshold $T$, BIRCH either needs $\Omega\left(\frac{|P|}{d}\right)$ CFs or it computes no constant factor approximation.

(For a generalized example)
Lessons from Coreset Theory

- Base insertion decision on induced error
- Error can be bound if the clustering cost of a CF is small
- Use packing arguments to bound number of CFs
Lessons from Coreset Theory

- Base insertion decision on induced error
- Error can be bound if the clustering cost of a CF is small
- Use packing arguments to bound number of CFs

Insertion of a point

1. Levels of clustering features, Start at top level
2. Search for closest CF
3. Point has to lie within the radius of the CF (Radius decreases by constant factor per level)
4. Add point if clustering cost of CF stays below $\frac{f(\varepsilon)}{k \cdot g_i} \cdot OPT$
5. If insertion fails, open a new CF or go one level down
Small Change, Huge Effect
Analysis: Quality

- Inspired by known coreset constructions
- Distinguish between points close to optimal centers (→ packing argument)
- and far away centers (error neglectable to clustering cost)
Small Change, Huge Effect

Analysis: Number of CFs

1. **Bound number of levels:**
   Constant Factor between Radii $\rightarrow$ number of points until full doubles $\rightarrow$ logarithmic in the number of points

2. **Number of full CFs:**
   can be bound by lower bound on their clustering cost

3. **Two types of non-full CFs:**
   Children of full CFs ($\rightarrow$ bound carries over)

4. and non-full CFs on the first level ($\rightarrow$ packing argument)
Analysis: Number of CFs

1. Bound number of levels:
   Constant Factor between Radii $\rightarrow$ number of points until full doubles $\rightarrow$ logarithmic in the number of points

2. Number of full CFs:
   can be bound by lower bound on their clustering cost

3. Two types of non-full CFs:
   Children of full CFs ($\rightarrow$ bound carries over)

4. and non-full CFs on the first level ($\rightarrow$ packing argument)

And if $OPT$ is not known?

- dynamically increase threshold
- Analysis still works, but gets more involved
Theorem

The modified BIRCH algorithm computes a \((k, \varepsilon)\)-coreset if \(OPT\) is known and can be modified for the case that \(OPT\) is not known. The size of the coreset is

\[
\mathcal{O}\left(\left(\frac{k}{f(\varepsilon)}\right)^d + 2^{c \cdot d} \cdot \frac{k}{f(\varepsilon)} \cdot \log n \log^2 \log n\right).
\]
And what is still missing... 
...is the experimental analyses. This is the next step :-}
Thank you for your attention!