Incremental clustering

Margareta Ackerman

Based on joint work with Sanjoy Dasgupta
What Is Incremental Clustering?

- Incremental clustering algorithms process the data one element at a time.
- They usually only store a small number of elements, such as a constant number.

- In contrast, batch methods store all of the data.
Sequential k-means

Set $T = (t_1, \ldots, t_k)$ to the first $k$ data points
Initialize the counts $n_1, \ldots, n_k$ to 1
Repeat:
  Acquire the next example, $x$
  If $t_i$ is the closest center to $x$:
    Increment $n_i$
    Replace $t_i$ by $t + (1/n_i)(x-t_i)$
Sequential nearest-neighbour

Set $T = (t_1, \ldots, t_k)$ to the first $k$ data points
Repeat:
   Get the next point $x$ and add it to $T$
   Let $t, t'$ be the two closest points in $T$
   Replace $t, t'$ by either of these two points
Due to increasing volumes of data, it is no longer possible to keep all the data in memory at the same time. So, many users are forced to transition to incremental methods. Ex. Recommendation systems
An even bigger gap between theory and practice: Incremental clustering

Meanwhile, theoretical foundations of incremental clustering are virtually unexplored.
We start with the following basic questions:

Are incremental methods as powerful as batch clustering algorithms?
The purpose of clustering algorithms is to detect clustering structure.

We would like to explore whether online algorithms are as good as batch methods in detecting clustering structure.

That is, we ask whether clustering structures that can be detected offline are also possible to discover using an incremental technique.
We will see that incremental clustering methods are *strictly weaker* than batch algorithms in their ability to detect clustering structure.

Then, we show how allowing extra clusters enables incremental methods to detect structures that are otherwise inaccessible.
A notion of clusterability defined by Balcan, Blum and Vempala (STOC, ’08)

This notion captures elementary clustering structure, and is easy to detect using batch methods.
A clustering is **nice** if every point is closer to all points within its cluster than to all other points.
A clustering is *nice* if every point is closer to all points within its cluster than to all other points.
A clustering is *nice* if every point is closer to all points within its cluster than to all other points.
A clustering algorithm is **nice-detecting**, if given any positive $k$ and any data set that has a unique nice $k$-clustering $C$, the algorithm outputs $C$. 
In the batch model, nice clusterings are easy to detect.

- Single-linkage is nice-detecting

- Other well-known algorithms, such as average-linkage and complete-linkage, also detect nice clusterings.
Our impossibility result

**Theorem** [Ackerman and Dasgupta, 2014]:
There is no incremental, memory-bound clustering algorithm that is nice-detecting.
We demonstrate the existence of a data set that has a nice clustering, yet no incremental, memory-bound clustering algorithm can detect it.
The data set consists of two nearly identical sets of points, $A$ and $B$, each having $n$ points.

There is going to a unique nice $3$-clustering of this data set, which will be one of:
- $A$ is a cluster, and $B$ is split into $2$ clusters, or
- $B$ is a cluster, and $A$ is split into $2$ clusters
Both $A$ and $B$ have the following structure:

There is a special point $x$
All other points have distance 1 to each other, and distance 2 to $x$. 

Many equidistant points at distance 1 from each other and distance 2 from $x$. 
So far, here is what our data looks like:
There are two nice clusterings in this data set:
Proof sketch

There are two nice clusterings in this data set:
We now add one more point to $A$ and to $B$ that will force the nice 3-clustering to be unique. These will be the last two points that the algorithm sees.
Point $y$ will have the same distances as all other points (that are not $x$), except for one “special distance.”

Many equidistant points, distance 1 from each other, and distance 2 from $x$.

This distance is set to either 1 or 2.

$y$ has distance 2 to $x$, and distance 1 to all but possibly one point in $A$. 

Proof sketch
Proof sketch

We make this “special distance” 2 in exactly one of A or B.

Many equidistant points, distance 1 from each other, and distance 2 from x.

y has distance 2 to x, and distance 1 to all but possibly one point in A.
If the special distance is set to $2$ in $A$, then this is the unique nice $3$-clustering.
If the special distance is set to 2 in $B$, then this is the unique nice 3-clustering.
Proof sketch

• In order to identify the nice 3-clustering, the algorithm needs to know which of A and B has the “special distance” set to 2.
• However, the special distance connects y to any one of \( n-2 \) points.
• Note that all but x and y are indistinguishable points.
• So, an algorithm would have to remember all \( n-2 \) points in order to identify the special distance.
• Since the algorithm is memory-bounded, this is not possible when \( n \) is large enough.
• What if we allow the algorithm to output extra clusters?
• That is, instead of finding the unique nice clustering, we only ask for refinement of it.
$C$ is a **refinement** of $C'$ if $C'$ can achieved by merging clusters in $C$. 
Refinement

$C$ is a \textbf{refinement} of $C'$ if $C'$ can achieved by merging clusters in $C$. 
Finding nice clustering with extra clusters

**Theorem** [Ackerman and Dasgupta, 2014]:

There exists an incremental clustering algorithm that identifies nice $k$-clusterings by using $2^{k-1}$ clusters.
Unfortunately, $2^{k-1}$ is only useful for very small values of $k$.

Can extra clusters help us identify cluster structure in a more practical way?

What is if we allow Sequential k-means to use extra clusters?
**Sequential k-means**

Set $T = (t_1, \ldots, t_k)$ to the first $k$ data points.

Initialize the counts $n_1, \ldots, n_k$ to 1.

Repeat:

1. Acquire the next example, $x$.
2. If $t_i$ is the closest center to $x$:
   - Increment $n_i$.
   - Replace $t_i$ by $t + (1/n_i)(x-t_i)$. 

Recall: Sequential k-means
Sequential $k$-means is not nice-detecting.

In fact, it cannot even detect perfect clusterings.
Sequential k-means with extra clusters

A slightly stronger assumption on the clustering structure allows sequential k-means with extra clusters to succeed.
A clustering $C=\{C_1, C_2, ..., C_k\}$ is **convex-nice** if for all $i \neq j$,

any $x,y$ in the convex hull of $C_i$,
and any $z$ in the convex hull of $C_j$,

$$d(y,x) < d(z,x).$$

All convex-nice clusterings are nice, but not all nice clusterings are convex nice.
A clustering is **nice** if every point is closer to all points within its cluster than to all other points.

All convex-nice clusterings are nice.

A nice, and convex-nice clustering.
Convex-nice doesn’t imply nice

Nice, but not convex nice.

$$ \mathbf{x}_1 = (1,0,0) $$

$$ \mathbf{x}_2 = (-1,0,0) $$

$$ \mathbf{x}_3 = (0,1,\sqrt{2}+\varepsilon) $$

$$ \mathbf{x}_4 = (0,-1,\sqrt{2}+\varepsilon) $$

$$ C = \{ \{ \mathbf{x}_1, \mathbf{x}_2 \}, \{ \mathbf{x}_3, \mathbf{x}_4 \} \} $$

x = (0,-1,\sqrt{2}+\varepsilon) and y = (0,1,\sqrt{2}+\varepsilon) are in the convex hull of \{x_3, x_4\}

z = (0,0,0) is in the convex hull of \{x_1, x_2\}

and yet x is closer to z than to y.
Theorem:

Let $X$ be a data set that has a convex-nice $k$-clustering $C$. Let $\beta$ denote the proportion of points in the smallest cluster in $C$.

If the points are ordered uniformly at random, then for any $\ell > k$, sequential $\ell$-means will return a refinement of $C$ with probability at least $1 - k \varepsilon^{\beta \ell}$. 
Proof Sketch

Compute the probability that the initial $k$ centers each lie in a distinct cluster of $C$ (works out to at least $1 - k\varepsilon - \beta\varepsilon$).

Since the clustering is convex nice, each center update keeps the centers in the convex hull of $C$.

So the final centers each lie in a district cluster of $C$, inducing this clustering.