How to select clustering algorithms?

Margareta Ackerman

Based on joint work with Ben-David and Loker (NIPS 2010) and Ben-David, Branzei and Loker (AAAI 2012)
Clustering is one of the most widely used tools for exploratory data analysis.

Social Sciences
Biology
Astronomy
Computer Science

All apply clustering to gain a first understanding of the structure of large data sets.
“While the interest in and application of cluster analysis has been rising rapidly, the abstract nature of the tool is still poorly understood” (Wright, 1973)

“There has been relatively little work aimed at reasoning about clustering independently of any particular algorithm, objective function, or generative data model” (Kleinberg, 2002)

Both statements still apply today.
Inherent Obstacles: Clustering is ill-defined

Clustering aims to assign data into groups of similar items

*Beyond that, there is very little consensus on the definition of clustering*
• Clustering is *inherently ambiguous*
  – There may be multiple reasonable clusterings
  – There is usually no ground truth
• There are many clustering algorithms with different (often implicit) objective functions
• Different algorithms have radically different input-output behaviour
Differences in Input/Output Behavior of Clustering Algorithms
Differences in Input/Output Behavior of Clustering Algorithms
There are a wide variety of clustering algorithms, which can produce very different clusterings.

How should a user decide which algorithm to use for a given application?
Users rely on cost related considerations: running times, space usage, software purchasing costs, etc...

There is inadequate emphasis on input-output behaviour
A framework that lets a user utilize prior knowledge to select an algorithm

- Identify properties that distinguish between different *input-output behaviour* of clustering paradigms
- The properties should be:
  1) Intuitive and “user-friendly”
  2) Useful for distinguishing clustering algorithms
Framework for Algorithm Selection

The goal is to understand fundamental differences between clustering methods, and convey them formally, clearly, and as simply as possible.
Property-based classification for fixed $k$
Ackerman, Ben-David, and Loker, NIPS 2010

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Single linkage</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Average linkage</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Complete linkage</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>K-means</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>K-medoids</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Min-Sum</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ratio-cut</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Normalized-cut</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Kleinberg’s axioms for fixed $k$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Single linkage</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Average linkage</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Complete linkage</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>K-means</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>K-medoids</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Min-Sum</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ratio-cut</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Normalized-cut</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Kleinberg’s Axioms are consistent when $k$ is given
Single-linkage satisfies everything

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Despite much work on clustering properties, some basic questions remained unanswered.

Consider some of the most popular clustering methods: $k$-means, single-linkage, average-linkage, etc...

- What are the advantages of $k$-means over other methods?
- We were missing some key properties.
There are three fundamental categories that clearly delineate some essential differences between common clustering methods.

The strength of these categories lies in their simplicity.
Every element is associated with a real valued weight, representing its “mass” or “importance”.

Generalizes the notion of element duplication.
Other Reasons to Add Weight: An Example

• Apply clustering to facility allocation, such as the placement of police stations in a new district.

• The distribution of stations should enable quick access to most areas in the district.

• Accessibility of different institutions to a station may have varying importance.

• The weighted setting enables a convenient method for prioritizing certain landmarks.
Traditional clustering algorithms can be readily translated into the weighted setting by considering their behavior on data containing element duplicates.
A *weight function* 
\[ w: X \rightarrow \mathbb{R}^+ \]
defines the weight of every element.

A *distance function* 
\[ d: X \times X \rightarrow \mathbb{R}^+ \cup \{0\} \]
is the distance defined between the elements.
(w[X],d) denotes weighted data

A Partitional Algorithm maps
Input: (w[X],d,k)
to
Output: a k-clustering of X
Towards Basic Categories

Range(X,d)

Range(A(X, d,k)) = \{C \mid \exists w \text{ s.t. } C = A(w[X], d)\}

The set of clusterings that \( A \) outputs on \((X, d)\) over all possible weight functions.
A is weight-robust if for all \((X, d)\),
\[|\text{Range}(X,d)| = 1.\]

\(A\) never responds to weight.
A is **weight-sensitive** if for all \((X, d)\),
\[|\text{Range}(X,d)| > 1.\]

\(A\) always responds to weight.
An algorithm \( A \) is **weight-considering** if

1) There exists \((X, d)\) where \(|\text{Range}(X,d)| = 1\).

2) There exists \((X, d)\) where \(|\text{Range}(X,d)| > 1\).

\( A \) responds to weight on some data sets, but not others.
Summary of Categories

\[ \text{Range}(A(X, d)) = \{C \mid \exists \ w \text{ such that } A(w[X], d) = C \} \]
\[ \text{Range}(A(X, d)) = \{D \mid \exists \ w \text{ such that } A(w[X], d) = D \} \]

**Weight-robust**: for all \((X, d)\), \(|\text{Range}(X,d)| = 1\).

**Weight-sensitive**: for all \((X, d)\), \(|\text{Range}(X,d)| > 1\).

**Weight-considering**:

1) \( \exists (X, d) \text{ where } |\text{Range}(X,d)| = 1 \).
2) \( \exists (X, d) \text{ where } |\text{Range}(X,d)| > 1 \).
Connecting To Applications

The desired category depends on the application.

In the facility allocation example above, a weight-sensitive algorithm may be preferred.

In phylogeny, where sampling procedures can be highly biased, weight robustness may be desired.
## Classification

<table>
<thead>
<tr>
<th>Weight Robust</th>
<th>Partitional</th>
<th>Hierarchical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min Diameter</td>
<td><strong>K-center</strong></td>
<td><strong>Single Linkage</strong></td>
</tr>
<tr>
<td>K-center</td>
<td></td>
<td><strong>Complete Linkage</strong></td>
</tr>
<tr>
<td>Weight Sensitive</td>
<td><strong>K-means, k-medoids, k-median, min-sum</strong></td>
<td><strong>Ward’s Method</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Bisecting K-means</strong></td>
</tr>
<tr>
<td>Weight Considering</td>
<td><strong>Ratio Cut</strong></td>
<td><strong>Average Linkage</strong></td>
</tr>
</tbody>
</table>
We show that k-means (the objective function) is weight-sensitive.

A is **weight-separable** if for any data set \((X, d)\) and subset \(S\) of \(X\) with at most \(k\) points, \(\exists w\) so that \(A(w[X],d,k)\) separates all points of \(S\).

**Fact:** Every algorithm that is weight-separable is also weight-sensitive.
Weighted $k$-means objective function

Given a clustering $\{C_1, C_2, \ldots, C_k\}$, the weighted $k$-means objective function is

$$
\sum_{i=1}^{k} \sum_{x \in C_i} w(x) \| x - c_i \|^2
$$

Where $c_i$ is the mean of $C_i$. That is,

$$
c_i = \frac{1}{\sum_{x \in C_i} w(x)} \sum_{x \in C_i} x w(x)
$$
**Theorem:** $k$-means is weight-sensitive.

**Proof:**
- Show that $k$-means is weight-separable
- Consider any $(X,d)$ and $S \subset X$ on at least $k$ points
- Increase weight of points in $S$ until each belongs to a distinct cluster.
These algorithms are invariant to element duplication.

Ex. Single linkage (Kruskle’s algorithm for minimum spanning tree)

As the minimum spanning tree is independent of the weight of the points, single-linkage is weight robust.
We will show that Average-Linkage is Weight Considering.

We could also characterize the precise conditions under which it is sensitive to weight.

Recall:
An algorithm $A$ is weight-considering if
1) There exists $(X, d)$ where $|\text{Range}(X,d)| = 1$.
2) There exists $(X, d)$ where $|\text{Range}(X,d)| > 1$. 
Weighted Average Linkage

• Average Linkage starts by creating a cluster for every element.
• It then repeatedly merges the “closest” clusters using the following linkage function, until exactly $k$ clusters remain:

$$\ell_{AL}(X_1, X_2, d, w) = \frac{\sum_{x \in X_1, y \in X_2} d(x, y) \cdot w(x) \cdot w(y)}{w(X_1) \cdot w(X_2)}$$
Data where Average Linkage ignore weights:

For k=2, average-linkage outputs the clustering \{{A,B}, \{C,D\}\} regardless of the weights (or number of occurrences) of these points.
Average Linkage is Weight Considering

Data where Average Linkage responds to weights:

Weights are all 1:

\[
\{\{A, B, C, D\},\{E\}\}
\]

Dark points have much higher weights than light points:

\[
\{\{A, B\},\{C, D, E\}\}
\]
When is average linkage sensitive to weight?

Turns out that it can be characterized!
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.

Not nice
When is Average Linkage Sensitive to Weight?

It can be shown that average-linkage ignores weights (for fixed $k$) on data that has a (unique) nice $k$-clustering.

Furthermore, it responds to weight when there are no nice $k$-clusterings.

There is a more elegant result in the hierarchical clustering setting.
What about heuristics?

• The above analysis for $k$-means and similar methods is for their corresponding objective functions.

• Unfortunately, optimal partitions are NP-hard to find. In practice, heuristics such as the Lloyd method are used.
### Heuristics Classification

<table>
<thead>
<tr>
<th>Weight Sensitive</th>
<th>Partitional</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Lloyd with random initialization</td>
<td></td>
</tr>
<tr>
<td>• K-means++</td>
<td></td>
</tr>
<tr>
<td>• PAM</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weight Considering</th>
<th>Partitional</th>
</tr>
</thead>
<tbody>
<tr>
<td>• The Lloyd Method with Further Centroid Initialization</td>
<td></td>
</tr>
</tbody>
</table>

Note that the more popular heuristics respond to weights in the same way as the $k$-means and $k$-medoids objective functions.
### Heuristics Classification

<table>
<thead>
<tr>
<th>Weight Sensitive</th>
<th>Partitional</th>
</tr>
</thead>
</table>
| Lloyd with random initialization | • Lloyd with random initialization  
  • K-means++  
  • PAM |

<table>
<thead>
<tr>
<th>Weight Considering</th>
<th>Partitional</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Lloyd Method with Further Centroid Initialization</td>
<td>• The Lloyd Method with Further Centroid Initialization</td>
</tr>
</tbody>
</table>

Just like Average-Linkage, the Lloyd Method with Furthest Centroid initialization responds to weight only on data without nice clusterings.
## Classification

<table>
<thead>
<tr>
<th>Weight Robust</th>
<th>Partitional</th>
<th>Hierarchical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Min Diameter</em></td>
<td><em>Single Linkage</em></td>
</tr>
<tr>
<td></td>
<td><em>K-center</em></td>
<td><em>Complete Linkage</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weight Sensitive</th>
<th>Partitional</th>
<th>Hierarchical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>K-means</em></td>
<td><em>Ward’s Method</em></td>
</tr>
<tr>
<td></td>
<td><em>k-medoids</em></td>
<td><em>Bisecting K-means</em></td>
</tr>
<tr>
<td></td>
<td><em>k-median</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Min-Sum</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Randomized Lloyd</em></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weight Considering</th>
<th>Partitional</th>
<th>Hierarchical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Ratio Cut</em></td>
<td><em>Average Linkage</em></td>
</tr>
<tr>
<td></td>
<td><em>Lloyd with Furthest Centroid</em></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

• We introduced a framework for choosing clustering algorithms based on their input-output behavior
• We saw three categories describing how algorithms respond to weights
• The same results apply in the non-weighted setting for data duplicates