Clustering with General Costs

A Digression from “Factor Discovery Through Optimal Transport”

Given by Daniel Wang on 07/27/2023
Mentors: Esteban G. Tabak, Andrew Lipnick, Nina Mortensen, Ryan Shijié Dù
The Roadmap

✦ Clusters as Discrete Factors
  ✦ Relaxing The Problem to k-Means

✦ k-Means: The Standard Algorithm

✦ k-GenCenters: An Extension of k-Means to General Costs
  ✦ Introduce k-Medians
  ✦ k-Means \textit{vs} k-Medians
  ✦ Improving initialization
  ✦ Must-link constraints
Clusters as Discrete Factors
A relaxation of the factor discovery problem

We seek factors $z$ and a map $y = T(x, z)$ that solve

$$
\max_z \left\{ \min_{y=T(x,z)} \int c(x, y) \rho(x|z) \gamma(z) \, dx \, dz \right\} \quad \text{s.t.} \quad y \perp z
$$
Clusters as Discrete Factors
A relaxation of the factor discovery problem

We seek factors \( z \) and a map \( y = T(x, z) \) that solve

\[
\max_z \left\{ \min_{y=T(x,z)} \int c(x, y) \rho(x \mid z) \gamma(z) \, dx \, dz \quad \text{s.t.} \quad y \perp z \right\}.
\]

In the case of a discrete-valued \( z \), a natural relaxation of the independence condition is that

\[
\bar{y} = \bar{y}(z) \quad \forall \ z
\]
Clusters as Discrete Factors
A relaxation of the factor discovery problem

We seek factors $z$ and a map $y = T(x, z)$ that solve

$$\max_z \left\{ \min_{y=T(x,z)} \left[ \int c(x, y)\rho(x | z)\gamma(z) \, dx \, dz \right] \right\} \quad \text{s.t.} \quad y \perp z$$

Premises of relaxation:

1) $\bar{y} = \bar{y}(z) \ \forall \ z$
Clustering as Discrete Factors
A relaxation of the factor discovery problem

We seek factors $z$ and a map $y = T(x, z)$ that solve

$$\max_z \left\{ \min_{y=T(x,z)} \int c(x, y)\rho(x\mid z)\gamma(z) \, dx \, dz \quad \text{s.t.} \quad y \perp z \right\}.$$ 

Premises of relaxation:

1) $\bar{y} = \bar{y}(z) \ \forall \ z$
2) $c(x, y) = \|x - y\|^2$
Clusters as Discrete Factors
A relaxation of the factor discovery problem

We seek factors $z$ and a map $y = T(x, z)$ that solve

$$\max_z \left\{ \min_{y=T(x,z)} \int c(x,y) \rho(x \mid z) \gamma(z) \, dx \, dz \quad \text{s.t.} \quad y \perp z \right\}. $$

Premises of relaxation:

1) $\overline{y} = \overline{y}(z)$ $\forall$ $z$

2) $c(x, y) = \|x - y\|^2$

We seek $I_k$ that solve the data-driven formulation,

$$\max_{I_k} \left\{ \sum_{k=1}^{p} [I_k] \|\overline{y} - \overline{x}(z_k)\|^2 \right\},$$

where $I_k$ is a set containing the identities of points attributable to the class $z_k$. 

Clusters as Discrete Factors
A relaxation of the factor discovery problem

We seek $I_k$ that solve the data-driven formulation,

$$\max_{I_k} \left\{ \sum_{k=1}^{p} [I_k] \|\bar{y} - \bar{x}(z_k)\|^2 \right\}.$$
Clusters as Discrete Factors

A relaxation of the factor discovery problem

We seek $I_k$ that solve the data-driven formulation,

$$\max_{I_k} \left\{ \sum_{k=1}^{p} [I_k] \| \bar{y} - \bar{x}(z_k) \|^2 \right\}.$$

Since $\bar{y} = \bar{x}$ and since

$$\sum_{k=1}^{p} \sum_{i \in I_k} \| x^i - \bar{x}(z_k) \|^2 + \sum_{k=1}^{p} [I_k] \| \bar{x} - \bar{x}(z_k) \|^2 = \sum_{i=1}^{n} \| x^i - \bar{x} \|^2,$$

our problem is equivalent to

$$\min_{I_k} \sum_{k=1}^{p} \sum_{i \in I_k} \| x^i - \bar{x}(z_k) \|^2.$$
Clusters as Discrete Factors
A refresher on the arithmetic mean

Given a set of numbers \( \{x^i \in \mathbb{R}\}_{i=1}^N \), the arithmetic mean is

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x^i
\]
Clusters as Discrete Factors
A refresher on the arithmetic mean

Given a set of numbers \( \{x^i \in \mathbb{R}\}_{i=1}^N \), the arithmetic mean is

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^N x^i
\]

\[
= \arg\min_{\hat{x}} \sum_{i=1}^N |x^i - \hat{x}|^2,
\]

which, performed component-wise with vectors \( x^i \), is precisely the centroid from k-Means.

In fact, our relaxed, data-driven optimization problem is equivalent to k-Means.

\[
\min_{I_k} \sum_{k=1}^P \sum_{i \in I_k} \|x^i - \bar{x}(z_k)\|^2
\]
**k-Means**

the standard algorithm
k-Means
the standard algorithm

k-Means
the standard algorithm
k-Means
the standard algorithm
k-Means
the standard algorithm

![Naive k-means clustering](image-url)
k-Means
the standard algorithm
k-Means
the standard algorithm
k-GenCenters
Coming To A Repository Near You…
k-GenCenters

An Extension of k-Means to General Costs

The k-GenCenters module...

• is styled after sklearn.cluster.KMeans
• has multiple initialization options, including kGenCenters++
• can perform variations on k-Means using
  ✦ Any $L^p$ norm $c(x, \hat{x}) = \left( \sum_{d=1}^{D} |x_d - \hat{x}_d|^p \right)^{\frac{1}{p}}$
  ✦ Any power of the Euclidean distance $c(x, \hat{x}) = \|x - \hat{x}\|_2^n$
  ✦ Any future cost functions contributed by the community
• can generate Voronoi diagrams
k-GenCenters

An Extension of k-Means to General Costs
k-GenCenters
An Extension of k-Means to General Costs
k-Medians
A refresher on the median

Given a set of numbers \( \{x^i \in \mathbb{R}\}_{i=0}^{N} \), the median is

\[
\text{Median} = x^{N/2}
\]

\[
= \arg\min_{\hat{x}} \sum_{i=0}^{N} |x^i - \hat{x}|
\]
k-Medians
A refresher on the median

Given a set of numbers \( \{ x^i \in \mathbb{R} \}_{i=0}^N \), the median is

\[
\text{Median} = x^{|\frac{N}{2}} = \arg\min_{\hat{x}} \sum_{i=0}^{N} |x^i - \hat{x}|
\]

Naïve k-Medians finds the component-wise medians of vectors \( x^i \), solving

\[
\arg\min_{\hat{x}} \sum_{i=0}^{N} \| x^i - \hat{x} \|_1.
\]
k-Medians
A refresher on the median

Given a set of numbers \( \{x^i \in \mathbb{R}\}_{i=0}^{N} \), the median is

\[
\text{Median} = x^{i_{\frac{N}{2}}}
\]

\[
= \arg\min_{\hat{x}} \sum_{i=0}^{N} |x^i - \hat{x}|
\]

Naïve k-Medians finds the component-wise medians of vectors \( x^i \), solving

\[
\arg\min_{\hat{x}} \sum_{i=0}^{N} \|x^i - \hat{x}\|_1.
\]
k-Medians
A refresher on the median

Given a set of numbers \( \{ x^i \in \mathbb{R} \}_{i=0}^{N} \), the median is

\[
\text{Median} = x_{\frac{N}{2}} = \arg\min_{\hat{x}} \sum_{i=0}^{N} |x^i - \hat{x}|.
\]

Naïve k-Medians finds the component-wise medians of vectors \( x^i \), solving

\[
\arg\min_{\hat{x}} \sum_{i=0}^{N} ||x^i - \hat{x}||_1.
\]
k-Medians
A refresher on the median

Given a set of numbers \( \{x^i \in \mathbb{R}\}_{i=0}^{N} \), the median is

\[
\text{Median} = \hat{x}^N
\]

\[
= \arg\min_{\hat{x}} \sum_{i=0}^{N} |x^i - \hat{x}|
\]

Naïve k-Medians finds the component-wise medians of vectors \( x^i \), solving

\[
\arg\min_{\hat{x}} \sum_{i=0}^{N} \|x^i - \hat{x}\|_1
\]

We want the more natural “geometric median”:

\[
\arg\min_{\hat{x}} \sum_{i=0}^{N} \|x^i - \hat{x}\|_2
\]
k-Means vs k-Medians
Performance with outliers

Robustness to Outliers: k-Means and k-Medians

Feature 1
Feature 2
True Labels
k-Means
Accuracy: 85.4%
k-Medians
Accuracy: 92.2%
k-Means vs k-Medians

Performance with outliers

Robustness to Outliers: k-Means and k-Medians

- **True Labels**
- **k-Means**
- **k-Medians**

<table>
<thead>
<tr>
<th>Feature 2</th>
<th>Feature 1</th>
<th><strong>Accuracy</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>50.5%</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>93.2%</strong></td>
</tr>
</tbody>
</table>
k-Means vs k-Medians
Performance on real-world datasets
k-Means vs k-Medians

Performance on real-world datasets
k-Means vs k-Medians
Performance on real-world datasets

E. Coli dataset (7 dimensions, 8 classes)
**k-Means vs k-Medians**

**Performance on real-world datasets**

Glass dataset (9 dimensions, 6 classes)
k-Means vs k-Medians

Performance on real-world datasets

Wine dataset (13 dimensions, 3 classes)
k-Means vs k-Medians
Performance on real-world datasets

Breast Cancer dataset (30 dimensions, 2 classes)
# k-Means vs k-Medians

A Performance Comparison

## Average Accuracy of k-Means and k-Medians (over 100 trials)

<table>
<thead>
<tr>
<th></th>
<th>k-Means</th>
<th>k-Medians</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>79.57%</td>
<td>78.73%</td>
</tr>
<tr>
<td>Seeds</td>
<td>91.71%</td>
<td>90.82%</td>
</tr>
<tr>
<td>E. Coli</td>
<td>74.47%</td>
<td>74.85%</td>
</tr>
<tr>
<td>Glass</td>
<td>44.84%</td>
<td>42.29%</td>
</tr>
<tr>
<td>Wine</td>
<td>94.31%</td>
<td>95.82%</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>90.80%</td>
<td>91.62%</td>
</tr>
</tbody>
</table>
Improving Initialization

Forgy
• Initialize the centers randomly

Random Partition
• Initialize the assignments of the data points randomly

k-Means++ (improved Forgy)
• Initialize the centers
• Incentivize distance from existing centers

\[ \mathbb{P}(z_k = x^i) = \frac{\min_{z_j < k} \|x^i - z_j\|^2}{\sum_{w=1}^{N} \min_{z_j < k} \|x^w - z_j\|^2} \]
Improving Initialization

Forgy

• Initialize the centers randomly

Random Partition

• Initialize the assignments of the data points randomly

\textbf{k-Means++ (improved Forgy)}

• Initialize the centers

• Incentivize distance from existing centers

\[ \mathbb{P}(z_k = x^i) = \frac{\min_{z_j < k} \| x^i - z_j \|^2}{\sum_{w=1}^{N} \min_{z_j < k} \| x^w - z_j \|^2} \]

\textbf{k-GenCenters++}

• Incentivize a \textit{custom cost} from existing centers

\[ \mathbb{P}(z_k = x^i) = \frac{\min_{z_j < k} c(x^i, z_j)}{\sum_{w=1}^{N} \min_{z_j < k} c(x^w, z_j)} \]
# Improving Initialization

## Breast Cancer Dataset:
Accuracies of Various Initializations (avg. over 100 trials)

<table>
<thead>
<tr>
<th>Initialization</th>
<th>k-Means</th>
<th>k-Medians</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forgy</td>
<td>90.80%</td>
<td>91.62%</td>
</tr>
<tr>
<td>Random Partition</td>
<td>90.73%</td>
<td>91.61%</td>
</tr>
<tr>
<td>++</td>
<td>90.94%</td>
<td>91.64%</td>
</tr>
<tr>
<td>Euclidean ++</td>
<td>90.90%</td>
<td>91.64%</td>
</tr>
<tr>
<td>Euclidean^2 ++</td>
<td>90.94%</td>
<td>91.65%</td>
</tr>
<tr>
<td>Euclidean^3 ++</td>
<td>90.99%</td>
<td>91.68%</td>
</tr>
</tbody>
</table>
Must-link Constraints
A form of semi-supervised clustering

Points *known to share a factor* (e.g. belonging to the same person) are “must-link”.
Must-link Constraints
A form of semi-supervised clustering

Points *known to share a factor* (e.g. belonging to the same person) are “must-link”.

At every assignment step, for every set of must-link identities $I_p$, we assign the whole set to

$$\arg\min_z \sum_{i \in I_p} c(x^i, z).$$
Must-link Constraints
A form of semi-supervised clustering

Points known to share a factor (e.g. belonging to the same person) are “must-link”.

At every assignment step, for every set of must-link identities $I_p$, we assign the whole set to

$$\arg\min_z \sum_{i \in I_p} c(x^i, z).$$

By this approach,

- the centers may be more robust against local minima
- the algorithm can be expected to converge faster
  - Further speedups may be accessible using a weighted surrogate
Acknowledgements

Prof. Esteban Tabak!
Andrew Lipnick and Nina Mortensen!

Olivia, Fortino, and Ryan!

Prof. Aleksandar Donev!