Clustering and Model Integration under the Wasserstein Metric

Jia Li
Department of Statistics
Penn State University
Clustering

• Data represented by vectors or pairwise distances.

• Methods
  • Top-down approaches
    • K-means
    • Statistical methods based on mixture models
  • Bottom-up approaches
    • Dendrogram clustering
  • Optimization vs. statistical modeling

• Applications
  • Exploratory study
  • Data reduction
Discrete Distributions

- Non-fixed support
- Generalize vectors
- Sparse histogram

$\Omega$: Discrete sample space

$p(v)$: Probability measure

$$\sum_{v \in \Omega} p(v) = 1$$

$$V = \{(v^{(1)}, p(v^{(1)})), \ldots, (v^{(t)}, p(v^{(t)}))\}$$

- Application 1: Cloud maps
- Application 2: Protein
Kantorovich-Wasserstein Metric

\[ X \sim Q_X, \ Y \sim Q_Y, \ X, \ Y \in \mathcal{R}^d \]

\[ D_P(Q_X, Q_Y) = \inf_{\mathcal{P}_X, \mathcal{P}_Y: \mathcal{P}_X = Q_X, \mathcal{P}_Y = Q_Y} (E \left\| X - Y \right\|^p)^{\frac{1}{p}} \]

- Gaussian distributions:
  \[ D^2(\phi(\mu_1, \Sigma_1), \phi(\mu_2, \Sigma_2)) = \left\| \mu_1 - \mu_2 \right\|^2 + tr(\Sigma_1) + tr(\Sigma_2) - 2tr[(\sqrt{\Sigma_1} \Sigma_2 \sqrt{\Sigma_1})^{1/2}] \].

- Discrete \( Q_X, \ Q_Y \): linear programming.

Wasserstein Distance for Discrete Distributions

\[ \mathcal{P}^{(k)} = \{(x_1^{(k)}, w_1^{(k)}), (x_2^{(k)}, w_2^{(k)}), \ldots, (x_{m^{(k)}}^{(k)}, w_{m^{(k)}}^{(k)})\}. \]

The Wasserstein distance \( D(\mathcal{P}^{(1)}, \mathcal{P}^{(2)}) \) is

\[
D^2(\mathcal{P}^{(1)}, \mathcal{P}^{(2)}) = \min_{\{\pi_{i,j}\}} \sum_{i=1}^{m^{(1)}} \sum_{j=1}^{m^{(2)}} \pi_{i,j} \| x_i^{(1)} - x_j^{(2)} \|^2
\]

s.t. \[
\sum_{j=1}^{m^{(2)}} \pi_{i,j} = w_i^{(1)}, \ i = 1, \ldots, m^{(1)},
\]

\[
\sum_{i=1}^{m^{(1)}} \pi_{i,j} = w_j^{(2)}, \ j = 1, \ldots, m^{(2)},
\]

\[ \pi_{i,j} \geq 0, \ i = 1, \ldots, m^{(1)}, j = 1, \ldots, m^{(2)}. \]

Linear programming
<table>
<thead>
<tr>
<th>Euclidean distance/cosine similarity</th>
<th>Wasserstein distance (<em>Kantorovich, 1942</em>)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Fixed bins, orthogonal axes</td>
<td>• Different supports</td>
</tr>
<tr>
<td>• Pros:</td>
<td>• Pros:</td>
</tr>
<tr>
<td>• Simple calculation</td>
<td>• Directly applies to discrete distributions</td>
</tr>
<tr>
<td>• Abundant methodologies</td>
<td>• Good to measure sparse and high dimensional data</td>
</tr>
<tr>
<td>• Cons:</td>
<td>• Cons:</td>
</tr>
<tr>
<td>• Inefficient in high dimensions</td>
<td>• Complex computation</td>
</tr>
<tr>
<td>• Sensitive to vector quantization</td>
<td></td>
</tr>
</tbody>
</table>
D2-Clustering

\[ B = \{ \mathcal{P}^{(i)} : \mathcal{P}^{(i)} \in \Omega, i = 1, ..., n \} \]

Optimize: \( A = \{ Q^{(i)} : Q^{(i)} \in \Omega, i = 1, ..., \tilde{n} \} \), and cluster assignment \( c(i) \in \{1, 2, ..., \tilde{n}\}, i = 1, ..., n \).

Optimization Criterion

\[ L(B, A^*, c^*) = \min_{A} \min_{c} \sum_{i=1}^{n} D^2(\mathcal{P}^{(i)}, Q^{(c(i))}) \]

K-means \( \rightarrow \) D2-Clustering

(vectors) \( \rightarrow \) (bags of weighted vectors)
D2-Clustering Algorithm

1. For every instance $i$, set

$$c(i) = \arg\min_{j=1,\ldots,\bar{n}} D^2(P^{(i)}, Q^{(j)}).$$

2. For each cluster $j$, $C_j = \{i : c(i) = j\}$, $j = 1, \ldots, \bar{n}$, solve

$$Q^{(j)} = \arg\min_{Q \in \Omega} \sum_{i \in C_j} D^2(P^{(i)}, Q) \quad \leftarrow \text{challenging}$$
Wasserstein Centroid Problem

Given a set of distributions \( \{ \mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(N)} \} \), solve the centroid \( Q = \{(w_1, x_1), \ldots, (w_m, x_m)\} \), such that

\[
\min_Q \frac{1}{N} \sum_{k=1}^N D^2(Q, \mathcal{P}^{(k)}) = \min_{x, w} \sum_{k=1}^N \min_{\Pi^{(k)}} \sum_{i \in \mathcal{I}', j \in \mathcal{I}_k} \pi_{i,j}^{(k)} \|x_i - x_j^{(k)}\|^2
\]

1. \( \mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{R}^{d \times m} \), \( \mathbf{w} = (w_1, \ldots, w_m) \in \mathbb{R}^+_m \).
2. \( \mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_m^{(k)}) \in \mathbb{R}^{d \times m^{(k)}} \), \( k = 1, \ldots, N \).
3. \( \Pi^{(k)} = (\pi_{i,j}^{(k)}) \in \mathbb{R}^+_{m \times m^{(k)}} \), \( k = 1, \ldots, N \).
4. \( \mathbf{X} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}) \in \mathbb{R}^{d \times n} \), where \( n = \sum_{k=1}^N m^{(k)} \).
5. \( \Pi = (\Pi^{(1)}, \ldots, \Pi^{(N)}) \in \mathbb{R}^{m \times n} \).
6. Index set \( \mathcal{I}^c = \{1, \ldots, N\} \), \( \mathcal{I}_k = \{1, \ldots, m^{(k)}\} \), for \( k \in \mathcal{I}^c \), and \( \mathcal{I}' = \{1, \ldots, m\} \).
With $\mathbf{w}$ and $\Pi$ fixed,

$$x_i := \frac{1}{Nw_i} \sum_{k=1}^{N} \sum_{j=1}^{m^{(k)}} \pi_{i,j}^{(k)} x_j^{(k)}, \quad i \in \mathcal{I}' ,$$

(1)

With fixed $\mathbf{x}$, updating $\mathbf{w}$ and $\Pi$ is a large-scale LP:

$$\min_{\Pi, \mathbf{w}} \quad \sum_{k=1}^{N} \sum_{i \in \mathcal{I}'. j \in \mathcal{I}_k} \pi_{i,j}^{(k)} \| x_i - x_j^{(k)} \|_2^2 ,$$

$$\text{s.t.} \quad \sum_{l=1}^{m^{(k)}} \pi_{i,l}^{(k)} = w_i , \quad \forall k \in \mathcal{I}_c, \quad i \in \mathcal{I}' ,$$

$$\sum_{l=1}^{m} \pi_{l,j}^{(k)} = w_j^{(k)} , \quad \forall k \in \mathcal{I}_c, \quad j \in \mathcal{I}_k ,$$

$$\sum_{l=1}^{m} w_l = 1 , \quad w_i \geq 0 ,$$

$$\pi_{i,j}^{(k)} \geq 0 , \quad \forall k \in \mathcal{I}_c, \quad i \in \mathcal{I}', \quad j \in \mathcal{I}_k.$$
Pairwise distances between symbols are accounted for.
Existing Clustering Approaches in Big Data

- **D2-clustering**
- **Spectral**
- **PCA + K-means**
- **DBSCAN**
- **Hierarchical**
- **K-spherical**
- **Mean-shift**
- **K-means**

Diagram showing the relationship between **Complexity**, **Discrete distribution**, **High-dimensional vector**, **Low-dimensional vector**, and **Volume** in the context of Big Data.
Modal EM for Clustering: HMAC

• Finite mixture model

\[ f(x) = \sum_{k=1}^{K} a_k f_k(x) = \sum_{k=1}^{K} a_k \phi(x | \mu_k, \Sigma_k) \]

Cloud Map Synthesis

• Different models generate a set of cloud maps.
• How to synthesize the cloud maps?

Set 1

Set 2
Modal clustering
- Clustering based on mode association: different kernel bandwidth determines the number of clusters
- Data: pixel locations
- Weight: intensity of cloud

Mixture representation
- Summarize each cluster by fitting one Gaussian distribution
- Location: Gaussian mean, Shape & spreadness: covariance matrix

Wasserstein barycenter
- Discrete distribution over the mean locations of the cloud segments
- Support size of the barycenter: average support size of the instances

Convert to GMM based on optimal matching weights
- Find optimal matching weights between the support points of the barycenter and those of any cloud map under the Wasserstein distance.
- Compute weighted average covariance matrix for the components in the barycenter.
Summary based on HMAC: relatively aggressive merging
Summary based on HMAC: less merging
Barycenter over 41 images: #support points 3, 5, 8, 15, 45

Average image

D2-clustering into 3 clusters, barycenters #support points: 8
Barycenter over 41 images: #support points 3, 5, 9, 15, 50

Average image

D2-clustering into 3 clusters
3 Barycenters, #support points: 9
References


Acknowledgment

• NSF DMS-0705210, NSF CCF-0936948