clustering uncertain data

Presented by Mohammad Hadi Salari

Simon Fraser University

msalari@sfu.ca

November 20, 2019
Overview

1. Motivations
   - Some Definitions
   - Applications

2. Problem Formulation

3. Related Works
   - Related Works Drawbacks

4. Our Solution
   - The First Try
   - The Second Try

5. Baseline

6. Data sets

7. Experiments
   - Evaluation
   - Experiment Result

8. Discussion and Conclusion

9. References
Some Definitions

- **Uncertain Data**: When each data point has a probability distribution over some space instead of being one certain point in the space.

  - (a) Amazon users ranking for one item
  - (b) The distribution of a person’s location

- **Clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters)[1]
Motivation

Applications of clustering uncertain points:
- Recommendation system
- Dimensionality reduction
- Summarization
- ...

Figure: clusters of items of an online shop
Problem Formulation

- We have $N$ points, each comes from a distribution: $p_i \sim P_i$
- We want to find $K$ clusters, whose centers come from a distribution: $q_c \sim Q_c$
- Point $i$ belongs to center $c$ with the probability of $\gamma_{ic}$
  $\forall p_i \sum_{q_c} \gamma_{ic} = 1.0$
- We define the dissimilarity of point $i$ to cluster $c$ by Kullback–Leibler divergence of their distributions.
  Other options: $\frac{1}{2}(KL(P_i \| Q_c) + KL(Q_c \| P_i))$, Jensen–Shannon divergence

Objective Function

$$\text{argmin} \sum_{i=1}^{N} \sum_{c=1}^{K} \gamma_{ic} KL(P_i \| Q_c)$$
Problem Formulation Example

- $p_1, p_2, p_3 \sim \mathcal{N}(0.0, 1.0)$ and $p_4, p_5 \sim \mathcal{N}(1.0, 1.0)$
- $q_1 \sim \mathcal{N}(0.0, 1.0)$ and $q_2 \sim \mathcal{N}(1.0, 1.0)$
- $\gamma_{11} = \gamma_{21} = \gamma_{31} = 1.0$ and $\gamma_{42} = \gamma_{52} = 1.0$
- Objective function =
  \[ 3KL(\mathcal{N}(0.0, 1.0) \mid \mid \mathcal{N}(0.0, 1.0)) + 2KL(\mathcal{N}(1.0, 1.0) \mid \mid \mathcal{N}(1.0, 1.0)) = 0 \]
Current works can be separated to three groups:

- **density-based algorithms**: Put the dense region of point to a cluster. e.g. FDBSCAN[2]

- **possible world-based algorithms**: A set of possible worlds is sampled from an uncertain data. Aggregating the result of the clusters of the possible worlds. e.g. [3]

- **partition-based algorithms**: Try to minimize the expected distance of points to their cluster centers. e.g. [4]
Related Works Drawbacks

- density-based algorithms: They assume that pairwise distances between uncertain objects are mutually independent which may not be a reasonable assumption.

**Independent Distance Assumption**

\[
P(A \leftrightarrow_\epsilon B, B \leftrightarrow_\epsilon C) = P(A \leftrightarrow_\epsilon B)P(B \leftrightarrow_\epsilon C)
\]

\((\leftrightarrow_\epsilon\) means that the distance is lower than an \(\epsilon\))

**Figure:** Distances of \(U\) to \(A\) and \(B\) are not independent
Related Works Drawbacks

- possible world-based algorithms: A sampled possible world does not consider the distribution of a data object. So the most probable clusters calculated using possible worlds may still carry a very low probability.

- partition-based algorithms: Current works assume hard clusters which means each data point either belongs to a cluster completely or not. Some of them also restrict cluster centers to be exactly one of the data points.

![Hard clustering vs Soft clustering](image)
We just have some samples from each $P_i$. In our method we use non-parametric Kernel density estimation with Gaussian kernels to estimate $P_i$s.

given the centers of our clusters we consider $\gamma_{ic}$ related to the KL-divergence of $P_i$ and $Q_c$.

$$\gamma_{ic} = \text{softmax}(-KL(P_i \| Q_c)) = \frac{e^{-KL(P_i \| Q_c)}}{\sum_{c'} e^{-KL(P_i \| Q_{c'})}}$$
Our Solution

Use an EM scheme to solve the problem.

The steps of the EM scheme

- step 1: given the probability of each point belonging to each cluster ($\gamma_{ic}$) we find the $Q_c$s that minimize the objective function.
- step 2: given the centers of the clusters ($Q_c$), we find the probability of each point belonging to each cluster ($\gamma_{ic}$).
We assume the centers are mixtures of Gaussians and use gradient descent to find their parameters that minimize the objective function.

\[ Q_c \sim \lambda_1 \mathcal{N}(\mu_1, \sigma_1) + \lambda_2 \mathcal{N}(\mu_2, \sigma_2) + \ldots + \lambda_k \mathcal{N}(\mu_k, \sigma_k) \]

Gradient of objective function of the centers parameters:

\[
\nabla (L) \nabla (\mu_c) = - \sum P_i \int P_i(x) \frac{\nabla (Q_c(x))}{Q_c(x)} \frac{\nabla (\mu_c)}{Q_c(x)} \, dx
\]

\[
\nabla (L) \nabla (\sigma_c) = - \sum P_i \int P_i(x) \frac{\nabla (Q_c(x))}{Q_c(x)} \frac{\nabla (\sigma_c)}{Q_c(x)} \, dx
\]

\[
\nabla (L) \nabla (\lambda) = - \sum P_i \int P_i(x) \frac{\nabla (Q_c(x))}{Q_c(x)} \frac{\nabla (\lambda)}{Q_c(x)} \, dx
\]
Gradient of $Q_c(X)$ of its parameters

\[
\nabla(Q_c(x)) \over \nabla(\mu_c) = \lambda_c \frac{1}{\sqrt{2\pi\sigma_c}} e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}} \frac{x-\mu_c}{\sigma_c^2}
\]

\[
\nabla(Q_c(x)) \over \nabla(\sigma_c) = \lambda_c \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}} \left( -1 + \frac{(x-\mu_c)^2}{\sigma_c^2} \right)
\]

\[
\nabla(Q_c(x)) \over \nabla(\lambda_c) = \frac{1}{\sqrt{2\pi\sigma_c}} e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}}
\]

We estimate those integrals with Monte Carlo integration.
Why it doesn’t work?!

- In each iteration, the gradient descent takes a lot of time.
- To have a low variance in the Monte Carlo estimation, we need to choose a large sample which makes the previous problem worse.
- We do not know how much complex the centers should be. So we to run the algorithm several times with different K and that makes the previous problems even worse!
We tried to find a close form for the objective function minimization.

**Minimizing objective function analytically**

\[
\begin{align*}
\text{argmin} & \sum_{i=1}^{N} \sum_{c=1}^{K} \gamma_{ic} KL(P_i\|Q_c) = \\
\text{argmin} & \sum_{c=1}^{K} \sum_{i=1}^{N} \gamma_{ic} KL(P_i\|Q_c) = \\
\text{argmin} & \sum_{c=1}^{K} \sum_{i=1}^{N} \gamma_{ic} \int P_i(x) \log \frac{P_i(x)}{Q_c(x)} \, dx = \\
\text{argmax} & \sum_{c=1}^{K} \sum_{i=1}^{N} \gamma_{ic} \int P_i(x) \log Q_c(x) \, dx = \\
\text{argmax} & \sum_{c=1}^{K} \int \sum_{i=1}^{N} \gamma_{ic} P_i(x) \log Q_c(x) \, dx = \\
\text{argmax} & \sum_{c=1}^{K} \int P'_c(x) \log Q_c(x) \, dx = \\
\text{argmin} & \sum_{c=1}^{K} \int P'_c(x) \log \frac{P'_c(x)}{Q_c(x)} \, dx = \\
\text{argmin} & \sum_{c=1}^{K} KL(P'_c\|Q_c) \Rightarrow \\
Q_c & \sim P'_c, \quad P'_c \sim \sum_{i=1}^{N} \frac{\gamma_{ic}}{\sum_{j=1}^{N} \gamma_{jc}} P_i
\end{align*}
\]
When should we terminate?

We define a confidence metric that shows how much confident are we for the resulting clusters in each iteration.

Confidence Metric

\[
\text{confidence} = \min_{p_i} \text{the difference between the first and second largest } \gamma_{ic}
\]

So we terminate our iterations when the confidence metric is higher than a threshold or the iteration number is higher than Max Iteration.
Centers Initialization

- First, we pick the point that minimize the objective function if we assign all points to this cluster.
- Then, to find initial center $t+1$, for each remaining point $(N-t)$ we first, compute $\gamma_{ic}$ if we add that point to our centers and then find the one that minimize our objective function.
We compare our algorithm on the Movement data set with the state-of-the-art clustering algorithms for uncertain data:

- UK-means (UKM) [5]
- CK-means (CKM) [6]
- UK-medoids (UKMD) [7]
- MMVar (MMV) [8]
- UCPC [9]
- FDBSCAN (FDB) [2]
- FOPTICS (FOP) [10]
- RPC [3]
Data sets

- **Synthetic data**: We generated each cluster as a mixture of Gaussians with different number of Gaussians. We generated 10 points for each cluster.

- **Real data**: Indoor User Movement Prediction from RSS data Data Set, 13197 radio signal records about 314 temporal sequences from a wireless sensor network. According to user movement path, the data set is divided into six classes.
If we consider G as ground truth clustering and C as the clustering obtained by our method.

- TP is the set of common pairs of objects in both C and G
- FP is the set of pairs of objects in C but not G
- FN is the set of pairs of objects in G but not C

**Precision and Recall**

\[
\text{Precision}(C) = \frac{|TP|}{|TP| + |FP|} \\
\text{Recall}(C) = \frac{|TP|}{|TP| + |FN|}
\]
**Experiment Result**

- With only 5 iteration we got the precision and recall of 1.0 on the synthetic data.
- With 14 iterations we got 0.36 precision and 0.34 recall on the Movement data set.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>UKM</th>
<th>CKM</th>
<th>UKMD</th>
<th>MMV</th>
<th>UCPC</th>
<th>FDB</th>
<th>FOP</th>
<th>PDB</th>
<th>SC</th>
<th>RPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movement</td>
<td>ACC</td>
<td>0.3490</td>
<td>0.3341</td>
<td>0.3478</td>
<td>0.3427</td>
<td>0.3494</td>
<td>0.2834</td>
<td>0.2643</td>
<td>0.3121</td>
<td>0.2548</td>
<td>0.4315</td>
</tr>
</tbody>
</table>

**Figure:** The precision of some state-of-the-art algorithms [3]
Discussion and Conclusion

- We proposed an EM algorithm to find clusters for uncertain data using KL-divergence distance.
- We used a non-parametric kernel density estimation to estimate the density function of data points, but it works not very well when the dimension of our space is larger than 2. We can use generative models such as Normalizing Flows to generate new samples and measure the probability of samples that work better for large dimensions.
- The time complexity of our algorithm is $O(N^2 \times \text{(sample size)} + \text{(iteration number)} \times N \times K)$. 

Presented by Mohammad Hadi Salari
clustering uncertain data
November 20, 2019 22 / 25

Hans-Peter Kriegel and Martin Pfeifle *Density-based clustering of uncertain data*. KDD 2005

Hongmei Liu and Xian-Chao Zhang and Xiaotong Zhang and Qimai Li and Xiao-ming Wu *Clustering Uncertain Data via Representative Possible Worlds with Consistency Learning*. ArXiv 2019

Bin Jiang and Jian Pei and Yufei Tao and Xuemin Lin-ming Wu *Clustering Uncertain Data Based on Probability Distribution Similarity*. IEEE Transactions on Knowledge and Data Engineering 2013


Thank You!