Auto-encoder Based Data Clustering

Chunfeng Song¹, Feng Liu², Yongzhen Huang¹, Liang Wang¹ and Tieniu Tan¹

1. NLPR, Chinese Academy of Sciences
2. Southeast University
Data clustering

Previous clustering method

- K-means
- Spectral clustering
- N-cut

Perform bad for bad distributed data.
Data clustering

Previous clustering method

- K-means
- Spectral clustering
- N-cut

Linear mapping

Perform bad for bad distributed data.

Can not deal with this similar images

Original Data Space

3 8 5
8 3 5
8 3
Data clustering

Previous clustering method

- ✓ K-means
- ✓ Spectral clustering
- ✓ N-cut

Linear mapping

Perform bad for bad distributed data.

✔ Auto-encoder based clustering can provide non-linear mapping.

Original Data Space

Feature Space

Non-linear Mapping
Auto-encoder

Basic single-layer auto-encoder

Is a kind of BP-NN

Encoder function

$$h_i = f(x_i) = \frac{1}{1 + \exp(-(W_1 x_i + b_1))}$$

Decoder function

$$x'_i = g(h_i) = \frac{1}{1 + \exp(-(W_2 h_i + b_2))}$$

Obj. function

$$\min \frac{1}{N} \sum_{i=1}^{N} \|x_i - x'_i\|^2$$

Basic BP-NN for classify

supervised
**Auto-encoder**

Basic single-layer auto-encoder

Is a kind of BP-NN

---

**Encoder function**

Sigmoid-type

\[
    h_i = f(x_i) = \frac{1}{1 + \exp(-(W_1 x_i + b_1))}
\]

**Decoder function**

\[
    x'_i = g(h_i) = \frac{1}{1 + \exp(-(W_2 h_i + b_2))}
\]

**Obj. function**

\[
    \min \frac{1}{N} \sum_{i=1}^{N} \|x_i - x'_i\|^2
\]
Auto-encoder

Basic single-layer auto-encoder

Encoder function

\[ h_i = f(x_i) = \frac{1}{1 + \exp(-(W_1 x_i + b_1))} \]

Decoder function

\[ x'_i = g(h_i) = \frac{1}{1 + \exp(-(W_2 h_i + b_2))} \]

Obj. function

\[ \min \frac{1}{N} \sum_{i=1}^{N} \| x_i - x'_i \|^2 \]
Auto-encoder

Basic single-layer auto-encoder

Is a kind of BP-NN

Different images may have **different** feature,

Similar images have **similar** feature!

**Encoder function**

\[ f(x) = \exp(-W_2h + b_2) \]

**Decoder function**

\[ x' = g(h) = \frac{1}{1 + \exp(-(W_2h + b_2))} \]

**Obj. function**

\[ \min \frac{1}{N} \sum_{i=1}^{N} \| x_i - x'_i \|^2 \]
Auto-encoder

Encoder function

Decoder function

Obj. function

Feature

Image

Reconstructed image

Different images may have different feature,

Similar images have similar feature!

Be fit for clustering

unsupervised

Basic single-layer auto-encoder

Is a kind of BP-NN

Basic single-layer auto-encoder

Is a kind of BP-NN

Sigmoid type

no-label

Different images may have different feature,

Similar images have similar feature!

Be fit for clustering

unsupervised

Basic single-layer auto-encoder

Is a kind of BP-NN

Sigmoid type

no-label
Auto-encoder

Basic single-layer auto-encoder

- $x'$: Reconstructed image
- $g(h)$: Feature
- $f(x)$: Image
- $W_1$: Unsupervised
- $W_2$: No-label
**Auto-encoder**

**Basic single-layer auto-encoder**

- $f(x) \rightarrow W_1$
- $h$ (feature) no-label
- $g(h) \rightarrow W_2$
- $x'$ (Reconstructed image)

**multi-layers auto-encoders**

**Encoder**

- $f(x)$
- $H$ 10 code layer
- $W_8$
- $W_9$
- $W_{10}$
- $1000$
- $250$

**Decoder**

- $g(H)$
- $W_1$
- $W_2$
- $W_3$
- $1000$
- $3$ $5$ $8$ ...

**unsupervised**
Auto-encoder

Basic single-layer auto-encoder

multi-layers auto-encoders

Why so deep?
Why so deep?

✓ Deep makes more accurate results.

✓ Deep networks can learn better.

✓ Deep networks can provide better non-linear mapping.
Why so deep?

- Deep makes more accurate results
- Deep networks can learn better
- Deep networks can provide better non-linear mapping

Is auto-encoder perfect for clustering?
Why so deep?

✓ Deep makes more accurate results
✓ Deep networks can learn better
✓ Deep networks can provide better non-linear mapping

Is auto-encoder perfect for clustering?

Not enough.
Clustering Based on Auto-encoder

Basic auto-encoder

\[ \min_{W,b} \frac{1}{N} \sum_{i=1}^{N} \| x_i - x'_i \|^2 \]
**Clustering Based on Auto-encoder**

Basic auto-encoder

\[
\min_{W,b} \frac{1}{N} \sum_{i=1}^{N} \| x_i - x'_i \|^2 - \lambda \cdot \sum_{i=1}^{N} \| f^t(x_i) - c^*_i \|^2
\]

- **Proposed**

- **Restrains added** to achieve compact distribution in **feature layer**

---

Objective function

- **Encoder**
- Basic Auto-encoder

- **Decoder**

**Iteration**

- Iteration = 1
- Iteration = t_1
- Iteration = t_2
Algorithm

$$\min_{W,b} \frac{1}{N} \sum_{i=1}^{N} \|x_i - x'_i\|^2 - \lambda \cdot \sum_{i=1}^{N} \|f^t(x_i) - c^*_i\|^2 \quad (4)$$

$$c^*_i = \arg \min_{c^t_{j-1}} \|f^t(x_i) - c^t_{j-1}\|^2, \quad (5)$$

$$c^t_j = \frac{\sum_{x_i \in C^t_{j-1}} f^t(x_i)}{|C^t_{j-1}|}, \quad (6)$$

**Algorithm 1** Auto-encoder based data clustering algorithm

1: **Input:** Dataset $X$, the number of clusters $K$, hyper-parameter $\lambda$, the maximum number of iterations $T$.
2: **Initialize** sample assignment $C^0$ randomly.
3: **Set** $t$ to 1.
4: **repeat**
5: Update the mapping network by minimizing Eqn. (4) with stochastic gradient descent for one epoch.
6: Update cluster center $c^t$ via Eqn. (6).
7: Partition $X$ into $K$ clusters and update the sample assignment $C^t$ via Eqn. (5).
8: $t = t + 1$.
9: **until** $t > T$
10: **Output:** Final sample assignment $C$. 
## Iteration

ACC: the cluster accuracy. **Distance**: the sum of distances between 10 clusters in feature layer.

<table>
<thead>
<tr>
<th>epoch</th>
<th>ACC</th>
<th>Distance</th>
<th>Visualization of 10 cluster centers (the reconstruction of feature)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.30</td>
<td>0.003</td>
<td>8 8 8 8 8 8 8 8 8 8</td>
</tr>
<tr>
<td>2</td>
<td>0.46</td>
<td>0.296</td>
<td>9 9 0 6 3 7 9 0 3 0</td>
</tr>
<tr>
<td>3</td>
<td>0.53</td>
<td>0.432</td>
<td>9 9 0 6 3 7 9 0 1 0</td>
</tr>
<tr>
<td>4</td>
<td>0.56</td>
<td>0.493</td>
<td>9 9 0 6 3 7 9 0 1 0</td>
</tr>
<tr>
<td>5</td>
<td>0.59</td>
<td>0.515</td>
<td>9 9 0 6 3 7 9 0 1 8</td>
</tr>
<tr>
<td>6</td>
<td>0.61</td>
<td>0.526</td>
<td>9 9 5 6 3 7 2 0 1 8</td>
</tr>
<tr>
<td>7</td>
<td>0.63</td>
<td>0.534</td>
<td>9 9 5 6 3 7 2 0 1 8</td>
</tr>
<tr>
<td>8</td>
<td>0.65</td>
<td>0.537</td>
<td>9 9 5 6 3 7 2 0 1 8</td>
</tr>
<tr>
<td>9</td>
<td>0.67</td>
<td>0.538</td>
<td>9 9 5 6 3 7 2 0 1 8</td>
</tr>
<tr>
<td>10</td>
<td>0.68</td>
<td>0.539</td>
<td>9 9 5 6 3 7 2 0 1 8</td>
</tr>
</tbody>
</table>

Test on MNIST datasets (including 60000 images with 28*28 resolution)
Experiments

- Influence of the iteration number on three databases

![MNIST](image1)
![USPS](image2)
![YaleB](image3)

- Performance comparison of clustering algorithms on three databases

<table>
<thead>
<tr>
<th>Datasets</th>
<th>MNIST</th>
<th>USPS</th>
<th>YaleB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criterion</td>
<td>NMI</td>
<td>ACC</td>
<td>NMI</td>
</tr>
<tr>
<td>K-means</td>
<td>0.494</td>
<td>0.535</td>
<td>0.615</td>
</tr>
<tr>
<td>Spectral</td>
<td>0.482</td>
<td>0.556</td>
<td><strong>0.662</strong></td>
</tr>
<tr>
<td>N-cut</td>
<td>0.507</td>
<td>0.543</td>
<td>0.657</td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>0.669</strong></td>
<td><strong>0.760</strong></td>
<td>0.651</td>
</tr>
</tbody>
</table>
Experiments

Distribution of data over ten clusters and the visualized images of cluster centers on MNIST

- **Auto-encoder Based Data Clustering**

- **K-means**
Experiments

- Performance comparison in three different spaces with \textit{k-means}.

<table>
<thead>
<tr>
<th></th>
<th>NMI</th>
<th>ACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>0.53</td>
<td>0.49</td>
</tr>
<tr>
<td>Auto-encoder</td>
<td>0.66</td>
<td>0.63</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.77</td>
<td>0.69</td>
</tr>
</tbody>
</table>

*Original means the images (pixel) space. Auto-encoder means the feature space trained by auto-encoder nets. Proposed means the feature space trained by restraints added auto-encoder nets.*
Conclusions

✓ Auto-encoder can provide good non-linear mapping.

✓ Auto-encoder nets can provide data-stable network.

✓ Restrains added can ensure compact.
Conclusions

✔ Auto-encoder can provide good non-linear mapping.

✔ Auto-encoder nets can provide data-stable network.

✔ Restrains added can ensure compact.

Is good for clustering.
Thank you!

Any questions?