Discriminative Clustering with Cardinality Constraints

The 2018 International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Calgary, Canada

Anh T. Pham (PhD student), Raviv Raich, and Xiaoli Z. Fern
School of EECS, Oregon State University, Corvallis, OR 97331-5501, USA
{phaman,raich,xfern}@eecs.oregonstate.edu
Clustering

• Clustering is one of the most important tasks in machine learning [Jain’PRL10]: e.g., displaying news and search engines.

• **Goal:** grouping similar objects in the same cluster

Clustering results
Constrained Clustering
Constrained Clustering

Instance-level constraints

Clustering with pair-wise constraints

Must-link constraint
Constrained Clustering

Instance-level constraints

Clustering with pair-wise constraints

Must-link constraint

Cannot link constraint
Constrained Clustering

Instance-level constraints

Clustering with pair-wise constraints

- Must-link constraint
- Cannot link constraint

• Well covered in literature [Basu’SDM04, Bilenko’ICML04, Wagstaff’ICML01]
Constrained Clustering

Instance-level constraints

- Clustering with pair-wise constraints

- Group-level constraints

- Clustering with cardinality constraints

Must-link constraint

Cannot link constraint

- Well covered in literature [Basu’SDM04, Bilenko’ICML04, Wagstaff’ICML01]
Constrained Clustering

Instance-level constraints

Clustering with pair-wise constraints

Cluster 1

Cluster 2

Must-link constraint
Cannot link constraint

• Well covered in literature [Basu’SDM04, Bilenko’ICML04, Wagstaff’ICML01]

Group-level constraints

Clustering with cardinality constraints

• E.g., 7 images in cluster 1 and 3 images in cluster 2
Constrained Clustering

- E.g., 7 images in cluster 1 and 3 images in cluster 2
- Limited coverage in literature

This work focuses on group-level constraints
Applications

• Political election: [Quadrianto’JMLR09]

  *E.g., Clinton vs. Trump electoral map*

<table>
<thead>
<tr>
<th>State</th>
<th>Date</th>
<th>Clinton</th>
<th>Trump</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alaska</td>
<td>1/24</td>
<td>44</td>
<td>49</td>
</tr>
<tr>
<td>Arizona</td>
<td>4/26</td>
<td>42</td>
<td>35</td>
</tr>
<tr>
<td>California</td>
<td>5/2</td>
<td>56</td>
<td>34</td>
</tr>
<tr>
<td>Connecticut</td>
<td>4/12</td>
<td>48</td>
<td>40</td>
</tr>
</tbody>
</table>

**Task:** Cluster individuals by political affiliation
Applications

• Political election: [Quadrianto’JMLR09]

  *E.g.*, *Clinton vs. Trump electoral map*

<table>
<thead>
<tr>
<th>State</th>
<th>Date</th>
<th>Clinton</th>
<th>Trump</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alaska</td>
<td>1/24</td>
<td>44</td>
<td>49</td>
</tr>
<tr>
<td>Arizona</td>
<td>4/26</td>
<td>42</td>
<td>35</td>
</tr>
<tr>
<td>California</td>
<td>5/2</td>
<td>56</td>
<td>34</td>
</tr>
<tr>
<td>Connecticut</td>
<td>4/12</td>
<td>48</td>
<td>40</td>
</tr>
</tbody>
</table>

  **Task:** Cluster individual by political affiliation

• Health-care data: [Yu’14]

  *E.g.*, *Proportions of 2 types of diabetes*

<table>
<thead>
<tr>
<th></th>
<th>Type 1</th>
<th>Type 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20.0</td>
<td>80.0</td>
</tr>
</tbody>
</table>

  **Task:** Cluster type 1 versus type 2 diabetes (e.g., for drug recommendation)
Problem formulation

• **Observed data:**
  - $X = [x_1, x_2, ..., x_n]$, where $x_i \in \mathbb{R}^d$ denotes the $i^{th}$ data point.
  - $N = [N_1, N_2, ..., N_C]$, where $N_c$ indicates the number of samples in class $c$.

• **Hidden data:**
  - $Y = [y_1, y_2, ..., y_n]$ denotes the hidden label for each sample, $y_i \in \{1, 2, ..., C\}$.
Problem formulation

• **Observed data:**
  - $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the $i^{th}$ data point.
  - $\mathbf{N} = [N_1, N_2, \ldots, N_C]$, where $N_c$ indicates the number of samples in class $c$.

• **Hidden data:**
  - $\mathbf{Y} = [y_1, y_2, \ldots, y_n]$ denotes the hidden label for each sample, $y_i \in \{1, 2, \ldots, C\}$.

• **Goal:**
  - Learn a mapping for each feature vector in $\mathbb{R}^d$ to a label in $\{1, 2, \ldots, C\}$. 
Discriminative model with cardinality constraints

• Suppose label $y_i$ is **known** for $x_i$, for all $i$

$$p(y_i = c | x_i, w) = \frac{e^{w_c^T x_i}}{\sum_{c=1}^{C} e^{w_c^T x_i}}$$
Discriminative model with cardinality constraints (cont.)

- However, $y_i$ is **unknown**

$$p(y_i = c | x_i, w) = \frac{e^{w_c^T x_i}}{\sum_{c=1}^{C} e^{w_c^T x_i}}$$

*Logistic regression*
Discriminative model with cardinality constraints (cont.)

Logistic regression

$\mathbf{W}$

$\mathbf{x}_i$

$\mathbf{y}_i$

$\mathbf{N}$

Cardinality constraints

$p(\mathbf{N}|\mathbf{y}) = \prod_{c=1}^{C} \mathbb{1}[N_c = \sum_{i=1}^{n} \mathbb{1}[y_i = c]]$
Discriminative model with cardinality constraints (cont.)

Logistic regression

Cardinality constraints

\[
p(N|y) = \prod_{c=1}^{C} \mathbb{I}[N_c = \sum_{i=1}^{n} \mathbb{I}[y_i = c]]
\]

Challenge: Too many ways to partition given \( N \) (e.g., \( N = [7, 3] \))

Crispness on the boundary may help
Model: Cluster crispness

- Generate $s$ labels for each sample
Model: Cluster crispness

- Generate $s$ labels for each sample
- Test if $s$ labels disagree using $d_i$ ($d_i \in \{0, 1\}$)
- Higher crispness, smaller no. of disagreements over the data

\[ d_i = 1 - \mathbb{I}[u_i^{(1)} = u_i^{(2)} = \cdots = u_i^{(s)}] \]
Model: Cluster crispness

- Generate $s$ labels for each sample
- Test if $s$ labels disagree using $d_i$ ($d_i \in \{0, 1\}$)
- Higher crispness, smaller no. of disagreements over the data
- $m$ controls total crispness in all data points

$$d_i = 1 - \mathbb{I}[u_i^{(1)} = u_i^{(2)} = \cdots = u_i^{(s)}]$$

$$p(I = 1|d) = \mathbb{I}[\sum_{i=1}^{n} d_i \leq m] \quad (m \text{ is a hyper-parameter})$$
Cluster crispness vs. Entropy

Crispness vs. entropy

(two class)
Model

Logistic regression

Hidden variables: y and u (marginalize d)

Cardinality constraints

Cluster crispness

Model 22

Logistic regression

Cardinality constraints

Cluster crispness
Inference

Complete log-likelihood

\[ \mathbf{L}_c(\mathbf{w}) = \log p(I, \mathbf{N}, \mathbf{y}, \mathbf{u}|\mathbf{X}, \mathbf{w}) \]

Auxiliary function

\[ Q(\mathbf{w}, \mathbf{w}') = E_{\mathbf{y}, \mathbf{u}|I, \mathbf{N}, \mathbf{w}'}[\mathbf{L}_c(\mathbf{w})] \]

\[ = \zeta + \sum_{i=1}^{n} \left[ \sum_{c=1}^{C} p(y_i = c|\mathbf{N}, \mathbf{X}, \mathbf{w}') \mathbf{w}_c^T \mathbf{x}_i - \log \left( \sum_{c=1}^{C} e^{\mathbf{w}_c^T \mathbf{x}_i} \right) \right] \]

\[ + s \times \left[ \sum_{c=1}^{C} p(u_i = c|I, \mathbf{X}, \mathbf{w}') \mathbf{w}_c^T \mathbf{x}_i - \log \left( \sum_{c=1}^{C} e^{\mathbf{w}_c^T \mathbf{x}_i} \right) \right] \]

E-step:

\[ p(y_i = c|\mathbf{N}, \mathbf{X}, \mathbf{w}') = \frac{p(y_i = c, \mathbf{N}|\mathbf{X}, \mathbf{w}')}{\sum_{l=1}^{C} p(y_i = l, \mathbf{N}|\mathbf{X}, \mathbf{w}')} \]

where \( \mathbf{w}' = \mathbf{w}^{(h)} \)

Similarly for \( P(u_i = c | I, \mathbf{X}, \mathbf{w}') \)

M-step:

\[ \mathbf{w}^{(h+1)} = \mathbf{w}^{(h)} + \eta \frac{\partial Q(\mathbf{w}, \mathbf{w}^{(h)})}{\partial \mathbf{w}} \bigg|_{\mathbf{w} = \mathbf{w}^{(h)}} \]
Dynamic programming for E-step

- \( N^i_c = \sum_{j \neq i} I[y_j = c] \),  
  \( p(y_i = c, N = v|X, w') = p(y_i = c|x_i, w')p(N^i = v - e_c|X, w') \)

- Compute \( p(N^i | X, w') \)?
Dynamic programming for E-step

- \( N_c^i = \sum_{j \neq i} I[y_j = c] \), \( p(y_i = c, N = v|X, w') = p(y_i = c|x_i, w')p(N^i = v - e_c|X, w') \)

- Compute \( p(N^i | X, w') \)

\( \sum_{y_i} p(N^i | X, w') = \sum_{y_i} \sum_{v} p(N^i = v - e_c|X, w')p(y_i = c|x_i, w') \)

- Infeasible for large \( C \)

\[ O(C^n) \]

\[ O(n^C) \]
Gaussian approximation for E-step

- \( y_i \sim p(y_i = c|x_i, w) \) and \( y_1, y_2, ..., y_n \) are independent given \( X \)
- \( N^c_i = \sum_{j=1,\neq i}^{n} I[y_i = c], \forall c \)
- \( N^{\setminus i} \) follows central limit theorem when \( n \) is sufficiently large (true in real-world application)
- \( N^{\setminus i} \) is multivariate normal with mean \( \mu^{\setminus i} = \sum_{j=1,\neq i}^{n} \mu_i \) and variance \( \Sigma^{\setminus i} = \sum_{j=1,\neq i}^{n} \Sigma_i \)
Experiments on MNIST

- **Datasets:** MNIST with pairs of digits: uniform among two classes.
- **Baseline:** K-means, Maximum-margin clustering (MMC) [Xu’NIPS04], Regularized Information Maximization (RIM) [Krause’NIPS10] (*RIM uses cardinality constraints*).
- **Evaluation metric:** Normalized mutual information (NMI) [Jain’PRL10], averaged 10 times
- **Setting:**
  - MNIST is reasonably well separated, $m = 0$, $s = 2$
  - Consider both dynamic programming implementation and Gaussian approximation
Experiments on MNIST

- **Datasets**: MNIST with pairs of digits: uniform among two classes.

- **Baseline**: K-means, Maximum-margin clustering (MMC) [Xu’NIPS04], Regularized Information Maximization (RIM) [Krause’NIPS10] (RIM uses cardinality constraints).

- **Evaluation metric**: Normalized mutual information (NMI) [Jain’PRL10], averaged 10 times

- **Setting**:
  - MNIST is reasonably well separated, \( m = 0, s = 2 \)
  - Consider both dynamic programming implementation and Gaussian approximation

<table>
<thead>
<tr>
<th>Datasets</th>
<th>1vs.2</th>
<th>3vs.4</th>
<th>5vs.6</th>
<th>7vs.8</th>
<th>9vs.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCCC-D</td>
<td>0.70</td>
<td>0.93</td>
<td>0.72</td>
<td>0.89</td>
<td>0.93</td>
</tr>
<tr>
<td>DCCC-G</td>
<td>0.70</td>
<td>0.93</td>
<td>0.72</td>
<td>0.89</td>
<td>0.93</td>
</tr>
</tbody>
</table>
Experiments on MNIST

- **Datasets:** MNIST with pairs of digits: uniform among two classes.

- **Baseline:** K-means, Maximum-margin clustering (MMC) \([\text{Xu'NIPS04}]\), Regularized Information Maximization (RIM) \([\text{Krause'NIPS10}]\) (RIM uses cardinality constraints).

- **Evaluation metric:** Normalized mutual information (NMI) \([\text{Jain'PRL10}]\), averaged 10 times

- **Setting:**
  - MNIST is reasonably well separated, \(m = 0, s = 2\)
  - Consider both dynamic programming implementation and Gaussian approximation

<table>
<thead>
<tr>
<th>Datasets</th>
<th>1vs.2</th>
<th>3vs.4</th>
<th>5vs.6</th>
<th>7vs.8</th>
<th>9vs.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCCC-D</td>
<td>0.70</td>
<td><strong>0.93</strong></td>
<td>0.72</td>
<td>0.89</td>
<td>0.93</td>
</tr>
<tr>
<td>DCCC-G</td>
<td>0.70</td>
<td><strong>0.93</strong></td>
<td>0.72</td>
<td>0.89</td>
<td>0.93</td>
</tr>
<tr>
<td>RIM</td>
<td><strong>0.73</strong></td>
<td>0.89</td>
<td>0.69</td>
<td>0.88</td>
<td><strong>0.93</strong></td>
</tr>
</tbody>
</table>
Experiments on MNIST

- **Datasets:** MNIST with pairs of digits: uniform among two classes.
- **Baseline:** K-means, Maximum-margin clustering (MMC) [Xu’NIPS04], Regularized Information Maximization (RIM) [Krause’NIPS10] (*RIM uses cardinality constraints*).
- **Evaluation metric:** Normalized mutual information (NMI) [Jain’PRL10], averaged 10 times
- **Setting:**
  - MNIST is reasonably well separated, $m = 0, s = 2$
  - Consider both dynamic programming implementation and Gaussian approximation

<table>
<thead>
<tr>
<th>Datasets</th>
<th>1vs.2</th>
<th>3vs.4</th>
<th>5vs.6</th>
<th>7vs.8</th>
<th>9vs.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCCC-D</td>
<td>0.70</td>
<td><strong>0.93</strong></td>
<td>0.72</td>
<td>0.89</td>
<td>0.93</td>
</tr>
<tr>
<td>DCCC-G</td>
<td>0.70</td>
<td><strong>0.93</strong></td>
<td>0.72</td>
<td>0.89</td>
<td>0.93</td>
</tr>
<tr>
<td>RIM</td>
<td><strong>0.73</strong></td>
<td>0.89</td>
<td>0.69</td>
<td>0.88</td>
<td><strong>0.93</strong></td>
</tr>
<tr>
<td>MMC</td>
<td>0.64</td>
<td>0.81</td>
<td>0.71</td>
<td>0.76</td>
<td>0.90</td>
</tr>
<tr>
<td>Kmeans</td>
<td>0.46</td>
<td>0.81</td>
<td>0.56</td>
<td>0.79</td>
<td>0.81</td>
</tr>
</tbody>
</table>
Experiments on real datasets

• Datasets:
  • HJA bird-song dataset (13 classes): each syllable is a sample
  • MSCV2 (19 classes) + Voc12 are image annotation (20 classes) datasets: each segment is a sample

• Baseline:
  • Consider Gaussian approximation $O(nC)$ only due to the high complexity of dynamic programming $O(n^C)$
  • Skip MMC since MMC is not applicable for multi-class

• Setting:
  • $s \in \{2,3\}, m \in \{10,20, ..., 50\}$. Tuning based on likelihood on validation set wrt. N.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>HJA bird song</th>
<th>MSCV2</th>
<th>Voc12</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCCC-G</td>
<td>0.40</td>
<td>0.31</td>
<td>0.12</td>
</tr>
<tr>
<td>RIM</td>
<td>0.39</td>
<td>0.25</td>
<td>0.11</td>
</tr>
<tr>
<td>K-means</td>
<td>0.06</td>
<td>0.13</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Conclusions

• We proposed a discriminative framework for clustering with cardinality constraints and high crispness.

• We proposed both exact and approximate inference.

• We verified the effectiveness of our method on synthetic and real world datasets.
References