
Main Contributions
- Non-Exhaustive, Overlapping Co-Clustering Problem:
  - Simultaneously identify a clustering of the rows as well as the columns of a two dimensional data matrix.
  - Both of the row and column clusters can overlap with each other.
- Outliers are not assigned to any cluster.
- An intuitive objective function is proposed to formulate this problem.
- NEO-CC: an efficient iterative algorithm that optimizes the non-exhaustive, overlapping co-clustering objective function.
- Experimental results show that the NEO-CC algorithm effectively captures the underlying co-clustering structure of real-world data.

An Example on a User-Movie Rating Matrix
- Result on a user-movie rating dataset where each row represents a user and each column represents a movie.
- The NEO-CC method detects one outlier from the rows, which corresponds to a user who randomly gives ratings.

The NEO-CC Algorithm

Input: \( X \in \mathbb{R}^{n \times m} \), \( k, l, \alpha_r, \alpha_c, \beta_r, \beta_c \)
Output: Row clustering \( U \in \{0,1\}^{r \times k} \), Column clustering \( V \in \{0,1\}^{c \times l} \)

1: Initialize \( U, V \), and \( r, l = 0 \).
2: while not converged do
3: Update row clustering by computing the distance between a data point \( x_p \in X \) for \( p = 1, \ldots, n \) and a row cluster \( C_i \) for \( i = 1, \ldots, k \).
4: Update column clustering by computing the distance between a data point \( x_q \in X \) for \( p = 1, \ldots, m \) and a column cluster \( C_j \) for \( q = 1, \ldots, l \).
5: end while

Example: the distance between a data point and a row cluster.

\[
U = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}
\]

\[
\text{dist}(x_1, C_1) = \|x_1 - m_{11}\|^2 + \|x_1 - m_{12}\|^2
\]

Experimental Results

Datasets: user-movie rating matrices, an yeast gene expression dataset, and a social network with node attributes.
- Compare the clustering performance of the NEO-CC method with other state-of-the-art co-clustering and one-way clustering methods.
- The NEO-CC algorithm achieves the highest \( F_1 \) scores.
- Co-clustering enables us to perform an implicit dimensionality reduction – performing an implicit regularized clustering.

Table: \( F_1 \) scores (% of the real-world datasets.)

<table>
<thead>
<tr>
<th></th>
<th>IPM</th>
<th>ROCC</th>
<th>MSSR1</th>
<th>MSSR2</th>
<th>NEO-iter</th>
<th>NEO-lrsdp</th>
<th>NEO-CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML1</td>
<td>22.4</td>
<td>55.7</td>
<td>43.8</td>
<td>44.2</td>
<td>58.3</td>
<td>56.4</td>
<td>58.1</td>
</tr>
<tr>
<td>worst</td>
<td>N/A</td>
<td>15.0</td>
<td>17.4</td>
<td>19.3</td>
<td>36.6</td>
<td>39.1</td>
<td>40.7</td>
</tr>
<tr>
<td>best</td>
<td>N/A</td>
<td>27.0</td>
<td>30.6</td>
<td>31.8</td>
<td>34.7</td>
<td>37.6</td>
<td>37.7</td>
</tr>
<tr>
<td>Yeast</td>
<td>24.0</td>
<td>28.7</td>
<td>27.3</td>
<td>33.3</td>
<td>36.4</td>
<td>39.0</td>
<td>36.2</td>
</tr>
<tr>
<td>worst</td>
<td>N/A</td>
<td>14.3</td>
<td>16.9</td>
<td>18.5</td>
<td>36.0</td>
<td>39.1</td>
<td>40.0</td>
</tr>
<tr>
<td>best</td>
<td>N/A</td>
<td>25.2</td>
<td>29.7</td>
<td>31.9</td>
<td>33.9</td>
<td>35.9</td>
<td>37.3</td>
</tr>
</tbody>
</table>

Conclusions & Future Work
- The NEO-CC method provides a principled way to capture the underlying co-clustering structure of real-world data.
- We plan to investigate a low-rank semidefinite programming for the NEO-CC method to develop a more sophisticated algorithm.