Agglomerative clustering with relational constraints of large symbolic data sets

Vladimir Batagelj
with Nataša Kejžar and Simona Korenjak-Černe

IMFM Ljubljana, IAM UP Koper and University of Ljubljana

SDA workshop: Data Science: New Data, New Paradigms
Université Paris-Dauphine, 22-23. January 2018
Outline

1. Clustering with relational constraint
2. Agglomerative method
3. Adjusting relation
4. Adjusting dissimilarity
5. Clustering symbolic data
6. Software support
7. Examples
8. Conclusions
9. References

Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si

Last version of slides (January 23, 2018, 16:36): CluReC.pdf
Suppose that the units are described by attribute data \( a : \mathcal{U} \to [\mathcal{U}] \) and related by a binary relation \( R \subseteq \mathcal{U} \times \mathcal{U} \) that determine the relational data or network \((\mathcal{U}, R, a)\).

The set \( R(X) = \{ Y : X R Y \} \) is a set of successors of unit \( X \in \mathcal{U} \) and for a cluster \( C \subseteq \mathcal{U} \)

\[
R(C) = \bigcup_{X \in C} R(X)
\]

We want to cluster the units according to a (dis)similarity of their descriptions, but also considering the relation \( R \) – it imposes constraints on the set of feasible clusterings, usually in the following form:

\[
\Phi(R) = \{ C \in \mathcal{P}(\mathcal{U}) : \text{each cluster } C \in \mathcal{C} \text{ induces a subgraph } (C, R \cap C \times C) \text{ in the graph } (\mathcal{U}, R) \text{ of the required type of connectedness} \}
\]
Clustering with relational constraints

\[
P(C) = \sum_{C \in \mathcal{C}} p(C), \quad p(C) = \sum_{X \in C} d(X, T_C)
\]

We can define different types of sets of feasible clusterings for the same relation \( R \). Some examples of types of relational constraint \( \Phi^i(R) \) are

<table>
<thead>
<tr>
<th>clusterings</th>
<th>type of connectedness</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Phi^1(R) )</td>
<td>weakly connected units</td>
</tr>
<tr>
<td>( \Phi^2(R) )</td>
<td>weakly connected units that contain at most one center</td>
</tr>
<tr>
<td>( \Phi^3(R) )</td>
<td>strongly connected units</td>
</tr>
<tr>
<td>( \Phi^4(R) )</td>
<td>clique</td>
</tr>
<tr>
<td>( \Phi^5(R) )</td>
<td>the existence of a trail containing all the units of the cluster</td>
</tr>
</tbody>
</table>

Trail – all arcs are distinct.
A set of units \( L \subseteq C \) is a center of cluster \( C \) in the clustering of type \( \Phi^2(R) \) iff the subgraph induced by \( L \) is strongly connected and \( R(L) \cap (C \setminus L) = \emptyset \).
We can use both hierarchical and local optimization methods for solving some types of problems with relational constraint (Ferligoj and Batagelj, 1983).

1. $k := n; \mathbf{C}(k) := \{\{X\} : X \in \mathcal{U}\}$;
2. while $\exists C_i, C_j \in \mathbf{C}(k): (i \neq j \land \psi(C_i, C_j))$ repeat
   2.1. $(C_p, C_q) := \text{argmin}\{D(C_i, C_j): i \neq j \land \psi(C_i, C_j)\}$;
   2.2. $\mathbf{C} := C_p \cup C_q; \ k := k - 1$;
   2.3. $\mathbf{C}(k) := \mathbf{C}(k + 1) \setminus \{C_p, C_q\} \cup \{C\}$;
   2.4. determine $D(C, C_s)$ for all $C_s \in \mathbf{C}(k)$
   2.5. adjust the relation $R$ as required by the clustering type
3. $m := k$

The fusibility condition $\psi(C_i, C_j)$ is equivalent to $C_iRC_j$ for tolerant, leader and strict method; and to $C_iRC_j \land C_jRC_i$ for two-way method.

For large data sets we assume that the relation is sparse.
Adjusting relation after joining

\( \phi^1 \) – tolerant
\( \phi^2 \) – leader
\( \phi^4 \) – two-way
\( \phi^5 \) – strict
Adjusting relation after joining

**tolerant**

\[
R(C_r) = \begin{cases} 
\{C_r\} \cup R(C_p) \cup R(C_q) \setminus \{C_p, C_q\} & \text{for } s \neq r \land \{C_p, C_q\} \cap R(C_s) \neq \emptyset \\
\{C_r\} \cup R(C_s) \setminus \{C_p, C_q\} & \text{otherwise}
\end{cases}
\]

**strict**

\[
R(C_r) = \begin{cases} 
\{C_r\} \cup R(C_p) \cup R(C_q) \setminus \{C_p, C_q\}, & \text{for } C_q R C_p \\
\{C_r\} \cup R(C_s) \setminus \{C_p, C_q\}, & \text{otherwise}
\end{cases}
\]

\[
R(C_s) = \begin{cases} 
\{C_s\} \cup R(C_s) \setminus \{C_p, C_q\}, & \text{for } s \neq r \land (C_p \in R(C_s) \lor C_q \in R(C_s) \land C_q R C_p) \\
R(C_s) \setminus \{C_p, C_q\}, & \text{otherwise for } s \neq r
\end{cases}
\]
Adjusting relation after joining

**leader**

\[
R(C_r) = \begin{cases} 
\{ C_r \} \cup R(C_p) \cup R(C_q) \setminus \{ C_p, C_q \}, & \text{for } C_q \sim C_p \\
\{ C_r \} \cup R(C_s) \setminus \{ C_p, C_q \}, & \text{otherwise}
\end{cases}
\]

\[
R(C_s) = \begin{cases} 
\{ C_s \} \cup R(C_s) \setminus \{ C_p, C_q \}, & \text{for } s \neq r \land \{ C_p. C_q \} \cap R(C_s) \neq \emptyset \\
R(C_s) \setminus \{ C_p, C_q \}, & \text{otherwise for } s \neq r
\end{cases}
\]

**two-way**

\[
R(C_r) = \{ C_r \} \cup (R(C_p) \cap R(C_q)) \setminus \{ C_p, C_q \}
\]

\[
R(C_s) = \begin{cases} 
\{ C_s \} \cup R(C_s) \setminus \{ C_p, C_q \}, & \text{for } s \neq r \land \{ C_p. C_q \} \subseteq R(C_s) \\
R(C_s) \setminus \{ C_p, C_q \}, & \text{otherwise for } s \neq r
\end{cases}
\]
Updating dissimilarities

In the original approach a complete dissimilarity matrix is needed. To obtain fast algorithms that can be applied to large data sets we propose to consider only the dissimilarities between linked units.

The step 2.4. “determine $D(C, C_s)$ for all $C_s \in \mathbf{C}(k)$” in the agglomerative procedure requires the adjustment of dissimilarities – computing the dissimilarities between new cluster $C$ and other remaining clusters. In the case of relational constraints we can limit the computation only to clusters that are related/linked to $C$.

This can be done efficiently in the following two cases:

- **first approach**: we define a dissimilarity $D(S, T)$ between clusters $S$ and $T$ that allows quick updates (as in Lance-Williams formula)

- **second approach**: to each cluster we assign a representative and we can efficiently compute a representative of merged clusters and a dissimilarity between clusters in terms of their representatives (Batagelj, 1988).
First approach: Dissimilarities between clusters

Let \((\mathcal{U}, R), R \subseteq \mathcal{U} \times \mathcal{U}\) be a graph and \(\emptyset \subset S, T \subset \mathcal{U}\) and \(S \cap T = \emptyset\). We call a block of relation \(R\) for \(S\) and \(T\) its part 
\[
R(S, T) = R \cap S \times T.
\]

The symmetric closure of relation \(R\) we denote with \(\hat{R} = R \cup R^{-1}\). It holds: \(\hat{R}(S, T) = \hat{R}(T, S)\).

For all dissimilarities between clusters \(D(S, T)\) we set:
\[
D(\{s\}, \{t\}) = \begin{cases} 
    d(s, t) & s \hat{R} t \\
    \infty & \text{otherwise}
\end{cases}
\]

where \(d\) is a selected dissimilarity between units.
... Dissimilarities between clusters

**Minimum**

\[ D_{\text{min}}(S, T) = \min_{(s, t) \in \hat{R}(S, T)} d(s, t) \]

\[ D_{\text{min}}(S, T_1 \cup T_2) = \min(D_{\text{min}}(S, T_1), D_{\text{min}}(S, T_2)) \]

**Maximum**

\[ D_{\text{max}}(S, T) = \max_{(s, t) \in \hat{R}(S, T)} d(s, t) \]

\[ D_{\text{max}}(S, T_1 \cup T_2) = \max(D_{\text{max}}(S, T_1), D_{\text{max}}(S, T_2)) \]
Dissimilarities between clusters

**Average**

$w : V \rightarrow \mathbb{R}$ – is a weight on units; for example $w(v) = 1$, for all $v \in U$.

$$D_a(S, T) = \frac{1}{w(\hat{R}(S, T))} \sum_{(s, t) \in \hat{R}(S, T)} d(s, t)$$

$$w(\hat{R}(S, T_1 \cup T_2)) = w(\hat{R}(S, T_1)) + w(\hat{R}(S, T_2))$$

$$D_a(S, T_1 \cup T_2) = \frac{w(\hat{R}(S, T_1))}{w(\hat{R}(S, T_1 \cup T_2))} D_a(S, T_1) + \frac{w(\hat{R}(S, T_2))}{w(\hat{R}(S, T_1 \cup T_2))} D_a(S, T_2)$$
Reducibility

The dissimilarity $D$ has the *reducibility* property (Bruynooghe, 1977) iff

$$D(C_p, C_q) \leq \min(D(C_p, C_s), D(C_q, C_s)) \Rightarrow$$

$$\min(D(C_p, C_s), D(C_q, C_s)) \leq D(C_p \cup C_q, C_s)$$

or equivalently

$$D(C_p, C_q) \leq t, \ D(C_p, C_s) \geq t, \ D(C_q, C_s) \geq t \Rightarrow \ D(C_p \cup C_q, C_s) \geq t$$

**Theorem**

*If a dissimilarity $D$ has the reducibility property then $h_D$ is a level function.*

All three disimilarities have the reducibility property. In this case also the *nearest neighbors network* for a given network is preserved after joining the nearest clusters. This allows us to develop a very fast agglomerative hierarchical clustering procedure (Murtagh, 1985). It is available in program Pajek.
To apply the first proposed approach to symbolic data we have to transform the constraining relation over a given set of symbolic objects $U$ into a network $N = (U, R, w)$ with weights for each $(X, Y) \in R \Rightarrow w(X, Y) = d(X, Y)$, where $d$ is a selected dissimilarity between symbolic objects.

For the second approach we need the representatives of clusters and a dissimilarity between clusters that can be expressed in terms of representatives. For symbolic objects described by discrete distributions (histograms, barcharts) there exist some possibilities (Batagelj et al., 2015).
Second approach: Some dissimilarities between discrete distributions

We introduce a dissimilarity measure between SOs and $T$ with

$$d(X, T) = \sum_i \alpha_i d_i(X, T), \quad \alpha_i \geq 0, \quad \sum_i \alpha_i = 1,$$

where $\alpha_i$ are weights for variables.

$$d_i(X, T) = \sum_{j=1}^{k_i} w_{xij} \delta(p_{xij}, t_{ij}), \quad w_{xij} \geq 0$$

where $w_{xij}$ are weights for each variable’s component.

$$w_C = \sum_{X \in C} w_X$$

$$P_C = \sum_{X \in C} w_X p_X$$

$$H_C = \sum_{X \in C} \frac{w_X}{p_X}$$

$$Q_C = \sum_{X \in C} w_X p_X^2$$

$$G_C = \sum_{X \in C} \frac{w_X}{p_X^2}$$

$$D(C_u, C_v) = p(C_u \cup C_v) - p(C_u) - p(C_v)$$
Representatives of clusters

The basic dissimilarities and the corresponding cluster leader, the leader of the merged clusters and dissimilarity between merged clusters. Indices $i$ and $j$ are omitted.

$$
\begin{array}{llll}
\delta(x, t) & t^*_c & z & D(C_u, C_v) \\
\delta_1 & (p_x - t)^2 & \frac{P_u}{w_c} + \frac{w_u}{w_v} & \frac{w_u \cdot w_v}{w_u + w_v} (u - v)^2 \\
\delta_2 & \left( \frac{p_x - t}{t} \right)^2 & \frac{Q_u}{P_c} & \frac{P_u}{P_c} + \frac{P_v}{P_c} \\
\delta_3 & \left( \frac{p_x - t}{t} \right)^2 & \sqrt{\frac{Q_u}{w_c}} & \sqrt{\frac{w_u}{w_c} + \frac{w_v}{w_c}} \\
\delta_4 & \left( \frac{p_x - t}{p_x} \right)^2 & \frac{H_u}{G_c} & \frac{H_u}{G_c} + \frac{H_v}{G_c} \\
\delta_5 & \left( \frac{p_x - t}{p_x} \right)^2 & \frac{w_c}{H_c} & \frac{w_u}{H_c} + \frac{w_v}{H_c} \\
\delta_6 & \left( \frac{p_x - t}{p_x} \right)^2 & \sqrt{\frac{P_c}{H_c}} & \sqrt{\frac{P_u}{P_c} + \frac{P_v}{P_c}} \\
\end{array}
$$
The first approach is implemented for weighted networks (weight is a dissimilarity) in Pajek – a program for analysis and visualization of large networks. Therefore we only need a program that transforms symbolic data set and constraining relation into weighted network (symData, Clamix). We intend to implement all parts in a single procedure in R.

An implementation in R of the second approach is still a work in progress.
Agglomerative clustering

V. Batagelj

Clustering with relational constraint
Agglomerative method
Adjusting relation
Adjusting dissimilarity
Clustering symbolic data
Software support
Examples
Conclusions
References
Conclusions

• (efficiently) implement the proposed methods in R;

• when applying the proposed methods to large data sets we assume that the constraining relation is sparse. Analyze the growth of the size of neighborhoods of joined units (clusters); the role of the structure of the relation (planar graph, large diameter, etc.);

• reevaluate the role of inversions in dendrograms in case of clustering with constraints.
References I


References II


This work was supported in part by the Slovenian Research Agency research program P1-0294 and research projects J7-8279 and J1-6720.