Comparing two clusterings using matchings between clusters of clusters

F. Cazals, D. Mazauric, R. Tetley, and R. Watrigant
Preprint 2017
Comparing clusterings

Motivation

Previous work

D-family matching: problem

Hardness

Algorithms

Experiments
Clustering algorithms

- Many algorithms: which one?
- Many parameters: which ones?
- Many clustering: are they consistent? A problem of scale...
Comparing clusterings: at which scale do clusters merge?

What is the right number of clusters?

▷ Example:
  ▷ Using k-means++ to cluster 5000 samples from five Gaussian blobs
  ▷ Using D-family matching to infer the right/natural # of clusters

(A) k-means++, $k = 20$

(B) k-means++, $k = 50$

(C) $D = 3$, 17 meta clusters, $\Phi = (4)068$

(D) $D = 4$, 4 meta clusters, $\Phi = (5)000$
Problem formalization in terms of intersection graph and **metaclusters**

▷ **Cluster editing:** can this be inferred?

![Diagram](image)

(a) $F_1$ $F_2$ $F$ $F'$

(b) $D = 1$

![Diagram](image)

(c) $D = 2$

![Diagram](image)

▷ **Rationale:** aggregating clusters into $k$ metaclusters
Comparing clusterings

Motivation

Previous work

D-family matching: problem

Hardness

Algorithms

Experiments
Comparing clusterings: matchings

- 1-1 mapping of clusters: equivalent to the problem of computing a maximum weighted matching in weighted bipartite graph.
- Solution: solved in $O(n^2 \log n + nm)$
- Particular case of the $D$-family-matching problem for $D = 1$ – see later
Comparing clusterings: the Variation of Information

- A set $Z$ of $t$ items
- A clustering $F$ of size $r$ for $Z$: $F = \{F_1, \ldots, F_r\}$; $n_k = |F_k|; p_k = n_k/t$.
- A clustering $F'$ of size $r'$ for $Z$: $F' = \{F_1, \ldots, F_r\}$; $n'_k = |F'_k|; p_k = n'_k/t$.
- Overlap between two clusters: $p(k, k') = |F_k \cap F'_k|/t$.
- Entropy of clustering: $H(F) = -\sum_{k=1}^r p(k) \ln p(k)$

- Mutual information between $F$ and $F'$:
  \[ I(F, F') = \sum_k \sum_{k'} p(k, k') \ln \frac{p(k, k')}{p(k)p(k')} \]

- Variation of information (VI):
  \[ VI(F, F') = H(F) + H(F') - 2I(F, F') \]

- Main properties:
  - $VI$ is a metric
  - $VI(F, F') \leq \ln t$

Comparing clusterings

Motivation

Previous work

D-family matching: problem

Hardness

Algorithms

Experiments
Intersection graph

▷ Notations:

- Data: \( Z = \{z_1, \ldots, z_t\} \)
- Clustering \( F \) of size \( r \): \( F = \{F_1, \ldots, F_r\} \)
  \[ F_i \subseteq Z, F_i \neq \emptyset \text{ and } F_i \cap F_j = \emptyset \text{ for every } i, j \in \{1, \ldots, r\}, i \neq j. \]
- Clustering \( F' \) of size \( r' \): \( F' = \{F'_1, \ldots, F'_{r'}\} \)
  \[ F'_i \subseteq Z, F'_i \neq \emptyset, \text{ and } F'_i \cap F'_j = \emptyset \text{ for every } i, j \in \{1, \ldots, r'\}, i \neq j. \]

NB: a clustering may not contain all \( t \) items

Definition 1 (Intersection graph \( G = (U, U', E, w) \) for \( F \) and \( F' \)).

The set \( U = \{u_1, \ldots, u_r\} \): vertices of \( F \)
The set \( U' = \{u'_1, \ldots, u'_{r'}\} \): vertices of \( F' \)
Edges \( E = \{\{u_i, u'_j\} \mid F_i \cap F'_j \neq \emptyset, 1 \leq i \leq r, 1 \leq j \leq r'\} \).
Edge weight of edge \( e = \{u_i, u'_j\} \in E \) is \( w_e = |F_i \cap F'_j| \).
D-family matching

Let $D \in \mathbb{N}^+$: a constraint on the diameter of certain subgraph of the intersection graph

Definition 2. [D-family-matching for an intersection graph]
A family $S = \{S_1, \ldots, S_k\}$, $k \geq 1$, such that

- for every $i, j \in \{1, \ldots, k\}$, if $i \neq j$, then: $S_i \subseteq V$, $S_i \neq \emptyset$, $S_i \cap S_j = \emptyset$,
- and the graph $G[S_i]$ induced by the set of nodes $S_i$ has diameter at most $D$.

Comments:
- $D = 1$: matching
- $D = 2$: clusters as stars

Notations:
- Set of all D-family matchings of a graph $G$: $S_D(G)$
D-family matching problem

Score $\Phi(S)$ of a $D$-family-matching $S$:

$$\Phi(S) = \sum_{i=1}^{k} \sum_{e \in E(G[S_i])} w_e.$$  \hspace{1cm} (1)

Remarks:

- The sum runs over all edges of a connected component. (Later: see algorithms based on spanning trees.)
- We wish to compute a $D$-family-matching which minimizes the inconsistencies.

Definition 3 ($D$-family-matching problem). Let $D \in \mathbb{N}^+$. Given an intersection graph $G$, the $D$-family-matching problem consists in computing

$$\Phi_D(G) = \max_{S \in S_D(G)} \Phi(S).$$  \hspace{1cm} (2)

NB: Score with the diameter $D$ stressed: $\Phi(S^{D=d})$
Figure: Simple instance of the D-family-matching problem and solutions: panels (c,d,e,f) represent optimal solutions for different values of D. (a) Simple instance of the D-family-matching problem with $t = 12$, $r = 5$, $r' = 4$, and so $n = 9$. The family $F$ contains five sets and the family $F'$ contains four sets. (b) Intersection graph $G$. (c) Optimal solution $S$ for $D \geq 7$ with $\Phi(S) = \Phi_D(G) = 12$. (d) Optimal solution $S$ for $D = 3$ with $\Phi(S) = \Phi_3(G) = 11$. (e) Optimal solution $S$ for $D = 2$ with $\Phi(S) = \Phi_2(G) = 9$. (f) Optimal solution $S$ for $D = 1$ with $\Phi(S) = \Phi_1(G) = 8$. 
<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z = {z_1, \ldots, z_t}$</td>
<td>Set of $t \geq 1$ elements</td>
</tr>
<tr>
<td>$F = {F_1, \ldots, F_r}$</td>
<td>Family of $r \geq 1$ disjoint subsets of $Z$</td>
</tr>
<tr>
<td>$F' = {F'<em>1, \ldots, F'</em>{r'}}$</td>
<td>Family of $r' \geq 1$ disjoint subsets of $Z$</td>
</tr>
<tr>
<td>$G = (V, E, w)$</td>
<td>Intersection graph of $n \geq 1$ nodes and $m \geq 1$ edges</td>
</tr>
<tr>
<td>$N_G(v) = {v' \mid {v, v'} \in E}$</td>
<td>Set of neighbors of node $v \in V$</td>
</tr>
<tr>
<td>$\Delta = \max_{v \in V}</td>
<td>N_G(v)</td>
</tr>
<tr>
<td>$cc(G)$</td>
<td>Set of maximal connected components of $G$</td>
</tr>
<tr>
<td>$S = {S_1, \ldots, S_k}$</td>
<td>$D$-family-matching</td>
</tr>
<tr>
<td>$\Phi(S) = \sum_{i=1}^{k} \sum_{e \in E(G[S_i])} w_e$</td>
<td>Score of a $D$-family-matching $S$</td>
</tr>
<tr>
<td>$S_D(G)$</td>
<td>Set of all $D$-family-matching for $G$</td>
</tr>
<tr>
<td>$\Phi_D(G) = \max_{S \in S_D(G)} \Phi(S)$</td>
<td>Optimal score for the $D$-family-matching problem</td>
</tr>
<tr>
<td>$S_D(G, T_r)$</td>
<td>Set of all $D$-family-matching constrained by $T_r$</td>
</tr>
<tr>
<td>$\Phi_D(G, T_r) = \max_{S \in S_D(G, T_r)} \Phi(S)$</td>
<td>Optimal score for the $D$-family-matching problem constrained by $T_r$</td>
</tr>
</tbody>
</table>
Comparing clusterings

Motivation

Previous work

D-family matching: problem

Hardness

Algorithms

Experiments
Main result

Theorem 4. Let $D \geq 2$ be any integer. The decision version of the $D$-family-matching problem is NP-complete for:

- bipartite graphs of maximum degree 3;
- bipartite graphs of maximum degree 4 even if the maximum weight is constant.

Moreover, the 2-family-matching problem is NP-complete for bipartite graphs of maximum degree 3 with unary weights.

Is the $D$-family-matching problem APX-hard or not? That is, does it admits a polynomial-time algorithm achieving a constant approximation ratio?
Lemma 5. For any integer $n \geq 1$, then there exists an intersection graph $G = (V, E, w)$ composed of $n$ nodes such that $\Phi_2(G)/\Phi_1(G) \geq n - 1$.

One has:

- $\Phi(S^{D=1}) = 1$ (one edge)
- $\Phi(S^{D=2}) = t = n - 1$ (all edges)
Comparing clusterings

Motivation

Previous work

D-family matching: problem

Hardness

Algorithms

Experiments
Theorem 6 (Computation of $\Phi_D(G)$ for trees). Let $D \in \mathbb{N}^+$. Consider any intersection tree $T = (V, E, w)$ of maximum degree $\Delta \geq 0$. Then, there exists an $O(D^2\Delta^2 n)$-time complexity algorithm for the $D$-family-matching problem for $T$.

**Proof.**
See black board.

Theorem 7. For any $D \in \mathbb{N}^+$, the $D$-family-matching problem can be solved:

- in $O(Dn)$ time if $G$ is a path;
- in $O(D^2 n)$ time if $G$ is a cycle(s) or a graph of maximum degree 2.

**Proof.**
See paper.
Generic approach on spanning trees

Def 2 for $D$-family-matching: a family of sets $S = \{S_1, \ldots, S_k\}$, $k \geq 1$, such that each $S_i$ induces a graph with diameter $\leq D$

Definition 8 ($D$-family-matching constrained by a tree). Let $G = (V, E, w)$ be an intersection graph and $T$ be a spanning tree of $G$. A $D$-family-matching for $G$ constrained by $T$ is a $D$-family-matching $S$ for $G$ such that all $S_i \in S$ induces a connected subtree in $T$. The set of all $D$-family-matching constrained by $T$ is denoted $S_D(G, T)$.

With this Def., we obtain the following sub-problem of $D$-family-matching:

Definition 9 ($D$-family-matching problem constrained by a tree). The $D$-family-matching problem consists in computing

$$\Phi_D(G, T) = \max_{S \in S_D(G, T)} \Phi(S)$$ (3)
Generic algorithm for the $D$-family-matching problem

Three ingredients:

- A property $\Pi(M)$, depending on the set $M$ of already computed $D$-family-matchings, represents the halting condition of the algorithm.
- A spanning tree generator $R(G, \lambda)$ computes the rooted spanning tree $T^\lambda$ of $G$ that is used at step $\lambda \geq 1$ by Algorithm $\mathcal{A}$.
- An algorithm $\mathcal{A}(G, T^\lambda, D)$ computes a $D$-family-matching $S^\lambda$ constrained by $T^\lambda$.

Generic algorithm for the $D$-family-matching problem:

Require: An intersection graph $G = (V, E, w)$, an integer $D \geq 1$, a property $\Pi$, a spanning tree generator $R$, and an algorithm $\mathcal{A}$.

1: $M := \emptyset$, $\lambda := 0$
2: while $\neg \Pi(M)$ do
3: $\lambda := \lambda + 1$; Compute the spanning tree $T^\lambda := R(G, \lambda)$
4: Compute $S^\lambda$ by using Algorithm $\mathcal{A}(G, T^\lambda, D)$; $M := M \cup S^\lambda$
5: return $S \in M$ of maximum score
Results on spanning trees

**Lemma 10.** Let $D \in \mathbb{N}^+$. Let $G$ be any intersection graph. Then, there exists a rooted spanning tree $T$ of $G$ such that $\Phi_D(G) = \Phi_D(G, T)$.

**Proof.**
See black board. □

**Lemma 11 (Computation of $\Phi_D(G, T)$).** Let $D \in \mathbb{N}^+$. Let $G = (V, E, w)$ be any intersection graph and $T$ be any spanning tree of $G$. Then, there exists a $O(2^{D\Delta \log_2(\Delta)} n)$-time algorithm for the $D$-family-matching problem for $G$ constrained by $T$.

**Proof.**
See paper. □
Comparing clusterings

Motivation

Previous work

D-family matching: problem

Hardness

Algorithms

Experiments
Generic code and instantiation for experiments

- Implementation in the SBL:

- Implementation $STS(G, D)$ has the following ingredients:
  
  - (i) the spanning tree generator $R$ returns a maximum spanning tree, or a random spanning tree;
  
  - (ii) the property $\Pi(M)$ returns true once we have computed a solution on the maximum spanning tree, as well as a solution on $n_i = (10,000)$ distinct random spanning trees (for a given $n_i$);
  
  - (iii) $A$: algorithm as in Theorem 6 with an additional step: edges for which both extremities belong to the same meta-cluster are added to the said meta-cluster. (In general, the intersection graph is indeed not a tree, so that such edges were unaccounted for.)
  
  - The solution returned for a given graph $G$ and a diameter $D$ is the best yielded by the aforementioned $1 + n_i$ spanning trees.
Randomly edited clusterings: setup

- **Initial random clusterings:**
  - \((t = 1\,000, r = 20)\) and \((t = 3\,000, r = 50)\).
  - Generated with the Boltzmann sampler from Flajolet - Duchon et al
  - Due to the randomness, the process is repeated \(N_r = 10\) times for each pair \((t, r)\).

- **Edited clusterings:** a copy \(F'\) of a clustering \(F\) is edited in two steps
  - Union operations: \(e\) unions reduce the number of clusters to \(r - e\)
  - Jittering: for each cluster, a fraction \(\tau\) of its items are distributed amongst the remaining \(k - 1\) clusters uniformly at random.

- **Values:**
  - \(e \in \{0, \lfloor r/4 \rfloor, \lfloor r/2 \rfloor\}\) and \(\tau \in \{0.05, 0.1, 0.2\}\). (NB: for \(e = 0\), \(F'\) is a jittered version of \(F\) (i.e. the numbers of clusters are identical.))
  - yields \(N_r \times \#(t, r) \times \#e \times \#\tau = 180\) comparisons, which are ascribed to 9 scenarii (3 values for \(e \times 3\) values for \(\tau\) ) denoted \(EeJy\), where \(y = 100\tau\).

- **Comparison against VI:** comparison of normalized scores \(\in [0, 1]\):
  \[
  s_\Phi = 1 - \Phi_D(\cdot)/t \text{ versus } s_{VI} = \text{VI}/\log t.
  \]
Randomly edited clusterings: results for \( (t = 1000, r = 20) \) (I)

\[
\begin{array}{c}
\text{D = 1} \\
\text{D = 2} \\
\text{D = 3} \\
\text{D = 4}
\end{array}
\]

Figure: Algorithm \( STS(G, D) \) for clusterings with \( (t = 1 \, 000, r = 20) \).

(Left) Best value for \( k \) as a function of the 9 scenarii. (Right) Scores \( s_\Phi \) as a function of the 9 scenarii.

- \( D \leq 2 \): algo. finds the right number of clusters \( \forall e \) (resp: 20, 15, 10)
- For \( D = 2 \): score \( \Phi_D(\cdot) \) is almost perfect \( (\geq 800, \text{wrt } t = 1000) \)
- Across scenarii: scores hardly depend on the jitter level
- For \( D = 3 \): scores \( \Phi_D(\cdot) \) varies significantly–but medians ok
- For \( D = 4 \): the algorithms output a full graph
On the separability of clusters and $D$: setup

- Datasets from five random samples each drawn from a 2D Gaussian distribution:
  - Relative position of the Gaussians: determined by a distance parameter $d$ controlling the separability of the four clusters
  - Values: $d = 5, 20, 50$.
- Clustering with k-means++ on each aforementioned dataset: varying $k^{++}$ in the range 5, 10, 20, 50, 100, 200

Fom Strehl et al (JMLR 2002): “In fact, the right number of clusters in a dataset often depends on the scale at which the dataset is inspected”.

![Diagram of Gaussian distributions with varying $d$ values and clustering results with k-means++ for different $k^{++}$ values]
On the separability of clusters and $D$: illustration

Figure: The diameter parameter $D$ allows recovering the correct number of clusters. (A, B) The two input clusterings of the random sample of Figure 27 for $k_1 = 20$ and $k_2 = 50$. (C) For $D = 3$: 17 meta clusters, $\Phi_3(\cdot) = 4068$. (D) For $D = 4$: 4 meta clusters, $\Phi_4(\cdot) = 5000$. 
Clusterings yielded by $k$-means++ and the influence of $D$

The influence of $D$ on $s_\Phi$ and $K$. Clusterings of the datasets from Fig. 27, obtained for various values of $k^{++}$ passed to $k$-means++. (Inset) An individual heatmap illustrates the $s_\Phi$; since we vary $d \in \{5, 20, 50\}$ and $D \in \{2, 3, 4, 5, 6\}$, the heat map has $3 \times 5 = 15$ entries. Each rectangle is labeled with the $K$ value returned by the best computed $D$-Family matching. (Main panel) All pairwise comparisons while varying the parameter $k^{++}$ of $k$-means++ in the set $\{5, 10, 20, 50, 100, 200\}$.
On the separability of clusters and $D$: outliers and pruning of the intersection graph

Take home messages:

▶ Whatever the values of $k_1$, $k_2$, there is a threshold of $D$ above which the score incurs a significant jump. Concomitantly the number of meta-clusters $k$ decreases. When the clusters are linearly separable (as is the case for $d = 50$), this corresponds to the “correct” number of clusters, that is four in our case.

▶ When the number of clusters is overestimated, large values of $D$ require to get the right number of clusters. For example, for $k_1, k_2 = 200$ and $d = 5$, $k = 37$ meta-clusters are obtained even at $D = 6$.

▶ Increasing solely $k_1$ (or only $k_2$) has no significant effect on the score or the number of meta-clusters. For example, one and four meta-clusters are always obtained when $k_1$ (or $k_2$) is less than 10. This is due to the topology of the intersection graph which is star-shaped, so that a small $D$ suffices to recover one or four clusters.
Pruning the intersection graph: outliers → erroneous edges → over-clustering

Figure: Pruning the intersection graph reduces the sensitivity of meta-clusters to outliers. (A, B) The clusterings of the dataset with $d = 20$ returned by $k$-means++ for $k_1 = 5$ (A) and $k_2 = 20$ (B). (C, D) The meta-clusters obtained for the clusters of (A, B) displayed on the intersection graph. No pruning (C). Edges whose weights are below 10 are pruned (D). Note that the two graphs respectively contain one and three connected components. (E, F) Meta-clusters defined from the graphs (C,D) displayed on the point clouds.
Comparison with the Variation of Information: results

- **Method**: scatter plot of $s_\Phi = 1 - \Phi_D(\cdot)/t$ versus $s_{VI} = VI/\log t$
  
  NB: 1 symbol per scenario; copy number of a symbol: number of repeats.

**Figure**: Normalized score $s_{VI}$ versus normalized score $s_\Phi$ of algorithm $STS(G, D)$. Each marker is a different union scenario and each color represents a different jitter scenario following the legend on the upper right. We plot the $y = x$ function for reference.
Comparison with the Variation of Information: results

- \( D = 2 \): \( s_\Phi \) corresponds to a matching.
- \( D = 2 \), two key differences with VI: \( s_\Phi \leq s_{VI} \); \( s_\Phi \) constant against union operations. Both \( s_{VI} \) and \( s_\Phi \) are affected by jittering.
- \( D = 3 \): higher variability in \( s_\Phi \); dependence on jittering and \# union operations.
- For \( D = 4 \): \( s_\Phi = 0 \) ie the full intersection graph reported.
Final words on the choice of $D$

Fom Strehl et al (JMLR 2002): “In fact, the right number of clusters in a dataset often depends on the scale at which the dataset is inspected”.

▶ Parameter $D$ acts as a scale parameter providing information of the structure of the intersection graph.

▶ When this graph is dense or has a specific topology (star-shaped), trivial values of $\Phi$ are obtained for small values of $D$, and a unit change of $D$ may trigger an abrupt change of $\Phi$. However, in more complex situations, large values of $D$ may be required.

▶ As a general strategy to choose $D$, we suggest identifying drops in $\Phi$ when decreasing $D$. Indeed, for any range of $D$ corresponding to a plateau for $\Phi$, the most significant value for $D$ is the smallest one.