How should we (correctly) compare n networks?

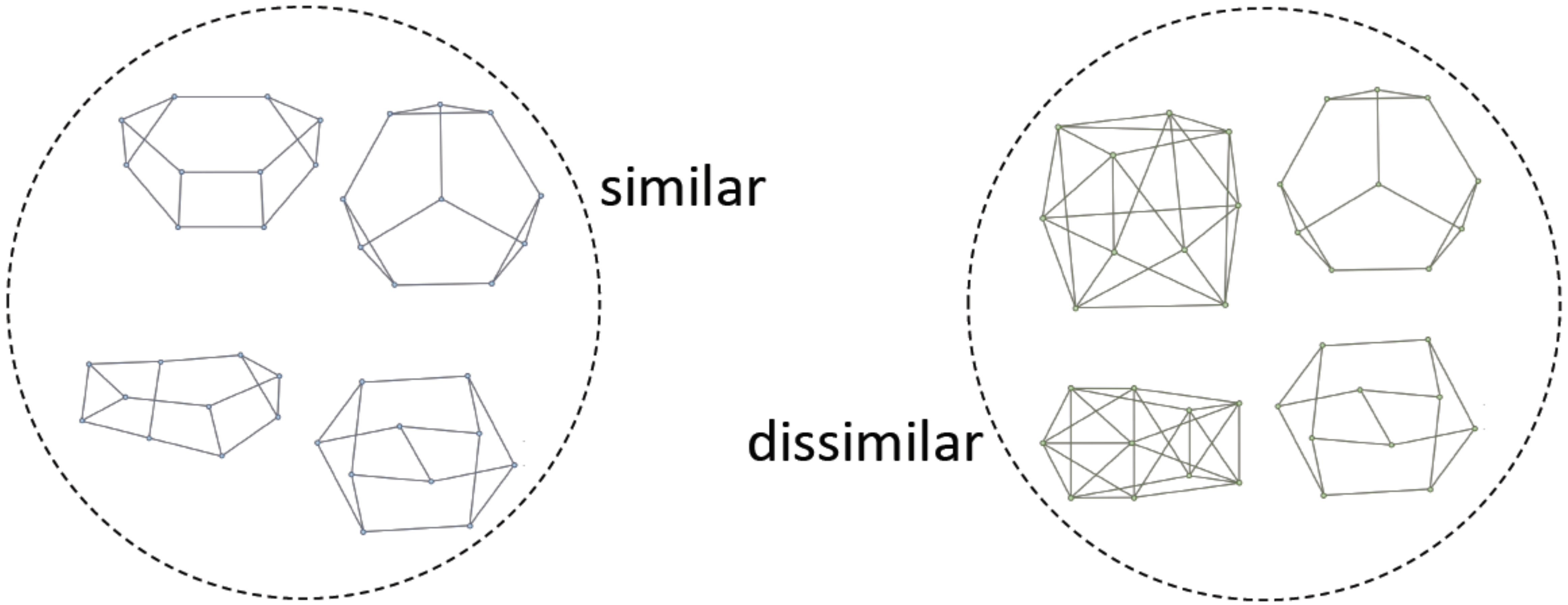
Open Data Science Conference
Boston, May 2019

Sam Safavi & José Bento

Problem

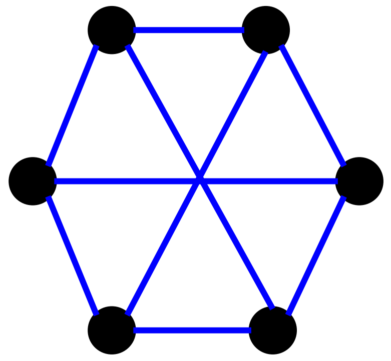
Given n graphs, $\{G_1, \cdots, G_n\}$, we want to find a notion of similarity, $d(G_1, \cdots, G_n)$, that gives a small value when the graphs are similar, and a large value when the graphs are not similar.

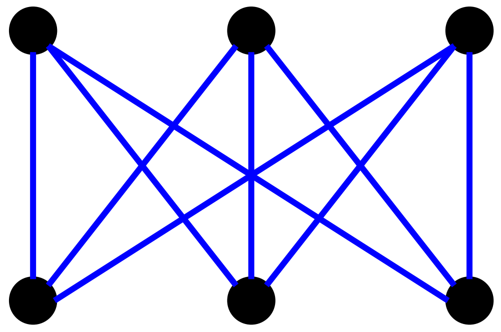
Problem



Problem

Note that graphs that do not look the same, might actually be the same (or closely related).





Problem

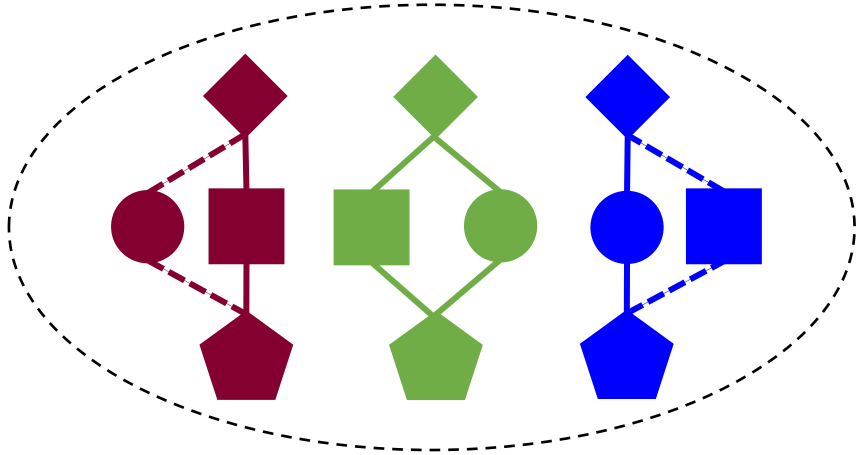
There are also many different kinds of graphs, and comparing them might require using information about:

1. Size (# nodes / edges)
2. Topology
3. Labels
4. Weights
5. Edge direction

Why is this important?

In biology, for example, the topology of a network of interacting proteins (a protein complex) might give some clues about its function, [Dohrmann et al. 15].

Having access to d, allows us to answer the question: "do these complexes have the same function?"

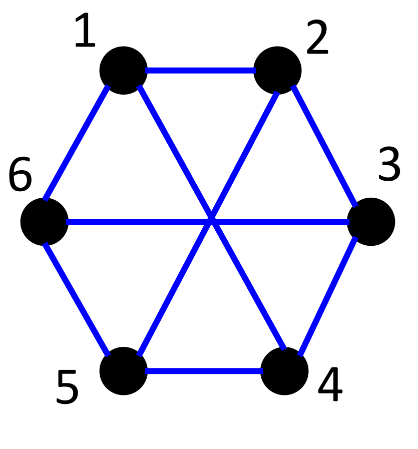


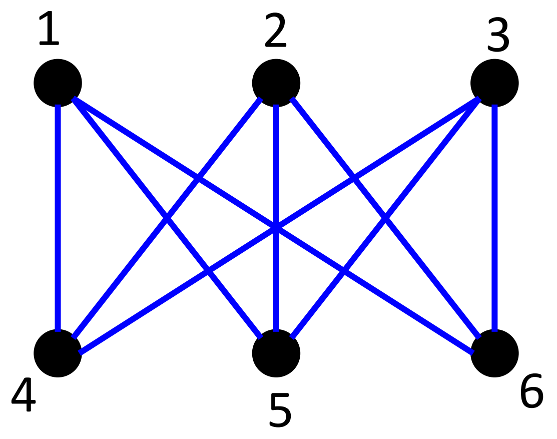
are these 3 nets.
similar, as a
group?

Some additional goals

In addition to looking for a measure of closeness between n graphs, we might want

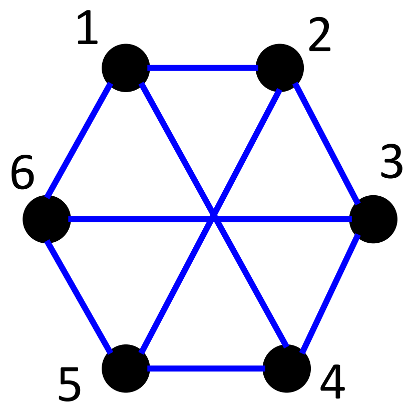
1. to find an association between the nodes of the graphs, such that it becomes clear why the graphs are similar or dissimilar.

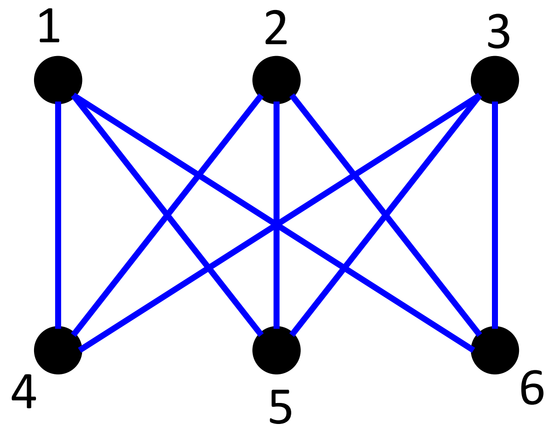




Some additional goals

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$





$$B = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

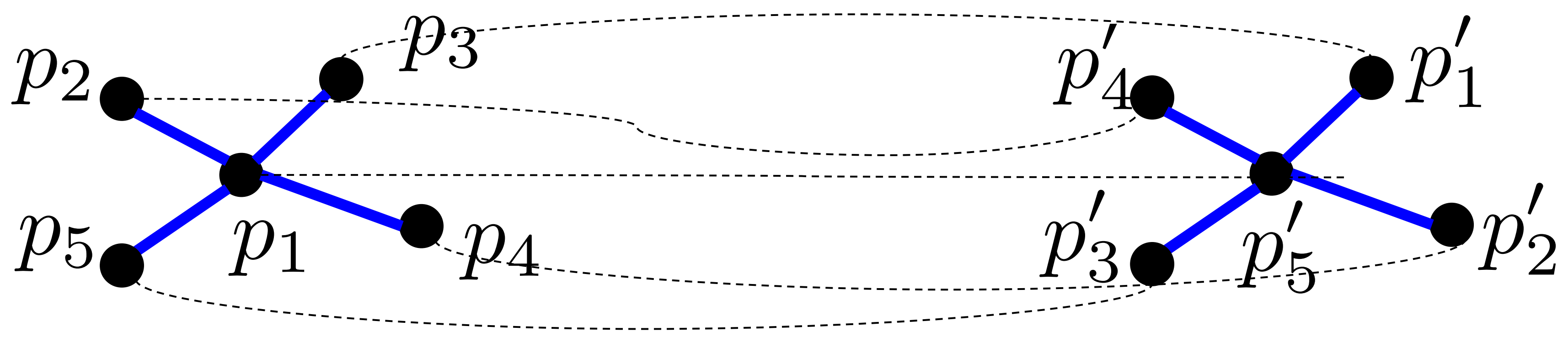
$$\|A - PBP^\top\|$$

$$\|AP - PB\|$$

$$= \text{small}$$

Some additional goals

This allows, e.g., knowledge transfer in PPI nets.

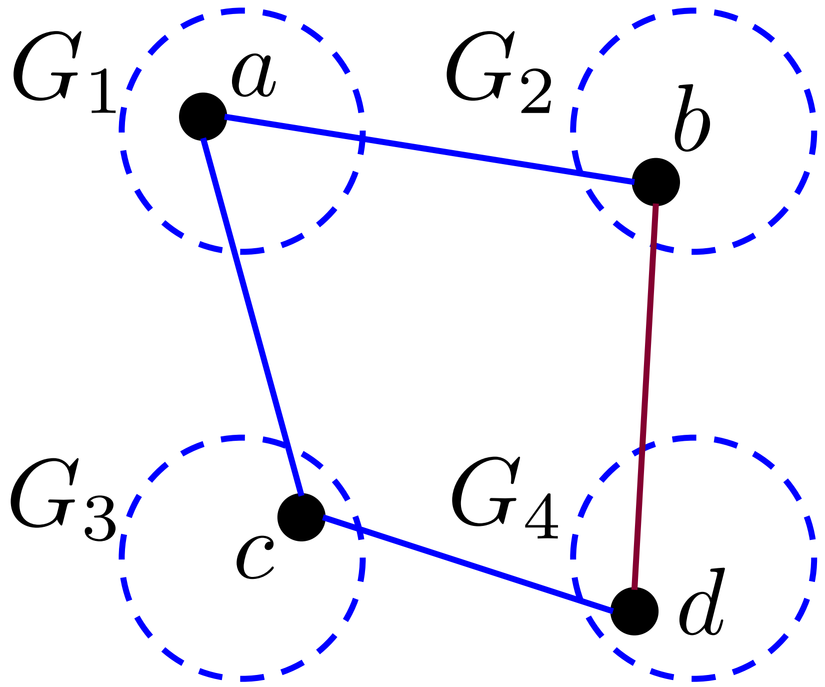


known function for p_1 inferred function for p'_5

Some additional goals

In addition to looking for a measure of closeness between n graphs, we might want

2. the association between multiple graphs to be consistent.



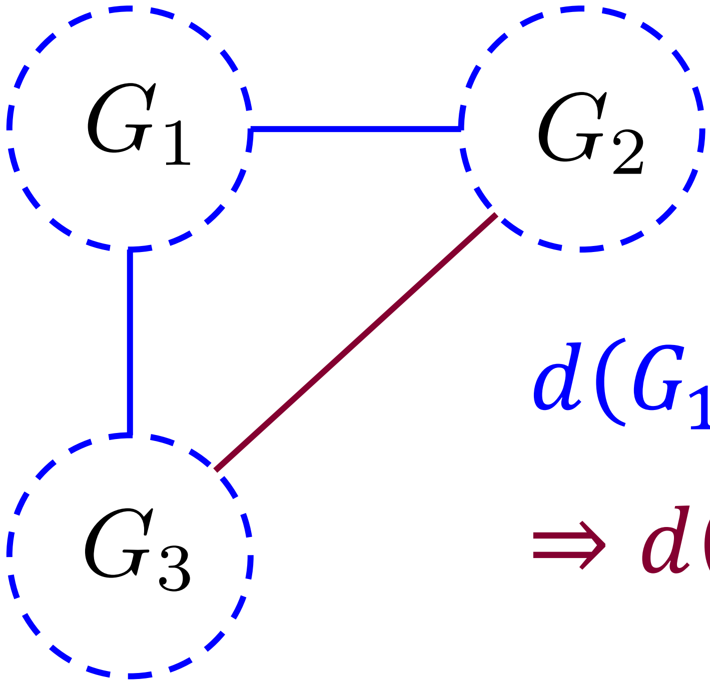
$a \sim b, a \sim c, c \sim d$

$\Rightarrow b \sim d$

Some additional goals

In addition to looking for a measure of closeness between n graphs, we might want

3. the distance function to satisfy intuitive properties of *metrics*.



$d(G_1, G_2), d(G_1, G_3)$ small

$\Rightarrow d(G_2, G_3)$ small

Some additional goals

For two graphs, d is a metric (2-metric) if the following conditions are satisfied:

$$
\begin{aligned}
&d(G_1, G_2) \geq 0, \\
&d(G_1, G_2) = 0, \text{ iff } G_1, G_2 \text{ are not distinct (isomorphic)}, \\
&d(G_1, G_2) = d(G_2, G_1), \\
&d(G_1, G_3) \leq d(G_1, G_2) + d(G_2, G_3).
\end{aligned}
$$

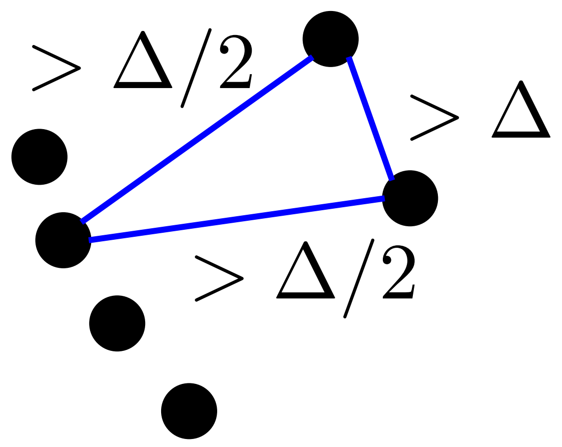
What about for *n* graphs? (more on this later)

Metrics and computational advantages

$$\max_{G_1, G_2 \in S} d(G_1, G_2)$$

$\mathcal{O}(|S|^2)$ v.s.

$\mathcal{O}(|S|)$ (1/2-approx. in expectation)



Related work

1. There are many different graph metrics, for two graphs, most of which are not easy to compute [Deza & Deza 2009]:

   a. Chemical distance

   b. Edit distance

   c. Maximum common subgraph distance

Related work

Chemical Distance

a mapping between the two graphs that minimizes their edge discrepancies:

$$\min_{P \in \Pi} \|A_1 P - P A_2\|_F$$

C.D. is zero if and only if two graphs are isomorphic.

Related work

Chemical Distance

For small graphs the C.D. is easy to compute. The for-loop can be trivially parallelized. Using a GPU, we can compute the C.D. for graphs of size $n = 13$ in $< 1\text{h}$.

```
best = inf;
all_perms = perms(1:n);
parfor i = 1:size(all_perms, 1)
    P = all_perms(i,:);
    if (norm(A1*P-P*A2)<best)
        best = norm(A1*P-P*A2);
    end
end
```

CPU (Malab) | GPU (CUDA C)

```
__global__ void kernel_to_compute_optimal_match(int
chunck_per_cycle, int num_perm_per_thread, lint nfact, int n, float
*A, float *B, float (*metric)(int , float* , float *, int* ), float
* obj_vals, lint * obj_perms ){

    int baseix = blockIdx.x*blockDim.x + threadIdx.x;;
    lint ix = baseix;
    extern __shared__ float AB_shared_mem[];
    float * shared_A = AB_shared_mem;
    float * shared_B = &AB_shared_mem[n*n];
    if (threadIdx.x == 0){
        for (int i = 0; i < n*n ; i++){
            shared_A[i] = A[i]; shared_B[i] = B[i];
        }
    }
    __syncthreads();
    float best_val = FLT_MAX;
    lint best_perm_ix;
    for (int i = 0; i < num_perm_per_thread ; i++){
        ix = baseix + chunck_per_cycle*i;
        if (ix < nfact){
            int perm[MAX_N_PERM]; int scrap[MAX_N_PERM];
            index_to_perm( ix , n, perm, scrap);
            float val = (*metric)( n,  shared_A ,  shared_B,
perm);
            if (val < best_val){
                best_val = val; best_perm_ix = ix;
            }
        }
    }
    obj_vals[baseix] = best_val; obj_perms[baseix] = best_perm_ix;
}
```

Related work

We can relax the constraint $P \in \Pi$ and obtain tractable metrics. For example, if Π is the set of doubly stochastic matrices, or the set of orthogonal matrices, then $\min_{P \in \Pi} \|A_1 P - P A_2\|_F$ is easy to compute, and is a metric [Bento & Ioannidis 2018].

Orthogonal matrices

$(P^T P = I \,;\, A_1, A_2 \; sym.)$

```
norm(sort(eigs(A1)) - sort(eigs(A2))
```

(Malab)

Doubly stochastic matrices

$(P \geq 0,\, 1P = 1,\, P^T = 1)$

(Malab - CVX)

```
cvx_begin
   variable P(n,n)
   minimize   (    norm(B*P - P*A)   )
      subject to
         P >= 0; sum(P ,1) == 1; sum(P ,2) == 1
cvx_end
```

Related work

Once a non-permutation P is obtained, we can project this to the permutations by solving a simple LP:

```
cvx_begin
   variable M(n,n)
   minimize   -trace( M*P' )
      subject to
         M >= 0; sum(M) == 1; sum(M') == 1;
cvx_end
```

(Malab - CVX)

This projection can destroy optimality/metric property.

Related work

Edit distance

$$\min_{\{e_i\}_{i=1}^{k} \in \mathcal{O}^k : G_1 = (e_k \circ \cdots \circ e_1) \circ G_2} \sum_{i=1}^{k} c(e_k)$$

$$\mathcal{O} = \{\text{vertex/edge/label} \\ \text{insertion/deletion/substitution}\}$$

For trees, we can solve this via dynamic prog. [Benjamin 2018].

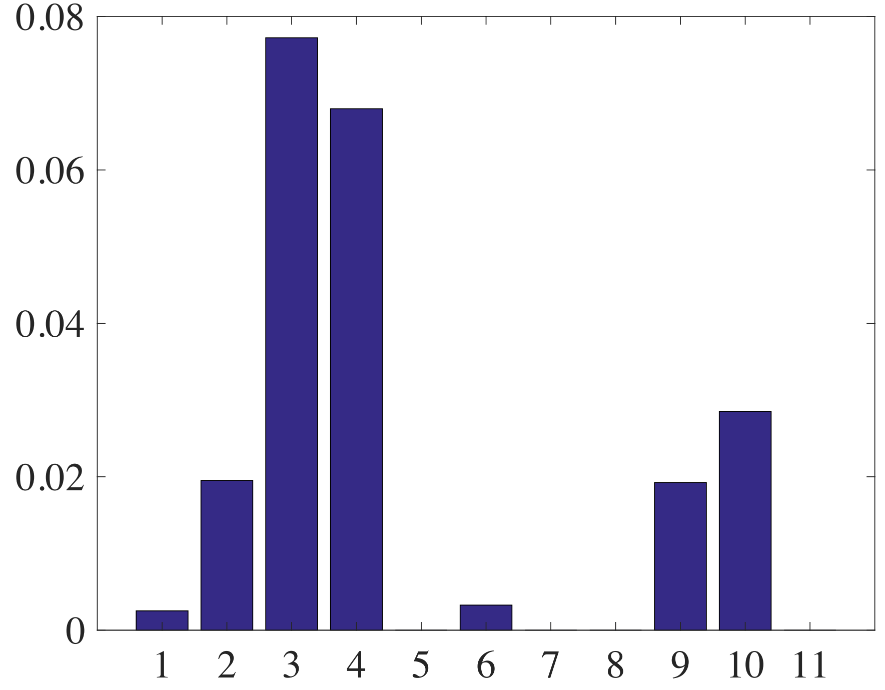
Max. common subgraph distance

$$\max\{|V_1|, |V_2|\} - n(G_1, G_2)$$

Related work

2. There are many different scalable methods to generate global alignments between two graphs, but many do not result in metrics [Bento & Ioannidis 2018].

Fraction on TIV among triples of 7 by 7 node graphs



1 - **InnerDSL2**
2 - **NetAlignBP**
3 - **IsoRank**
4 - **SparseIsoRank**
5 - **NetAlignMR**
6 - **Natalie**
7 - **DSL1**
8 - **DSL2**
9 - **InnerPerm**
10 - **InnerDSL1**

$\left.\begin{array}{l}1\text{–}5\end{array}\right\}$ Bayatti et a. 2009

$\left.\begin{array}{l}6\end{array}\right\}$ El-Kebir et a. 2015

$\left.\begin{array}{l}9\text{–}10\end{array}\right\}$ Lyzinski et a. 2014

Related work

3. There are many different scalable methods to generate global alignments between n graphs, which have not been tested for their “metric’’ properties. Most algorithms come from computer vision.
    i. [Guibas et al. 2013, 2018]
    ii. [Daniilidis et al. 2015]
    iii. [Stephen et al. 2015]
    iv. [Tong et al. 2015]
    v. [Huang et al. 2014]
    vi. [Singh et al. 2013]

Mathematical background: n-metrics

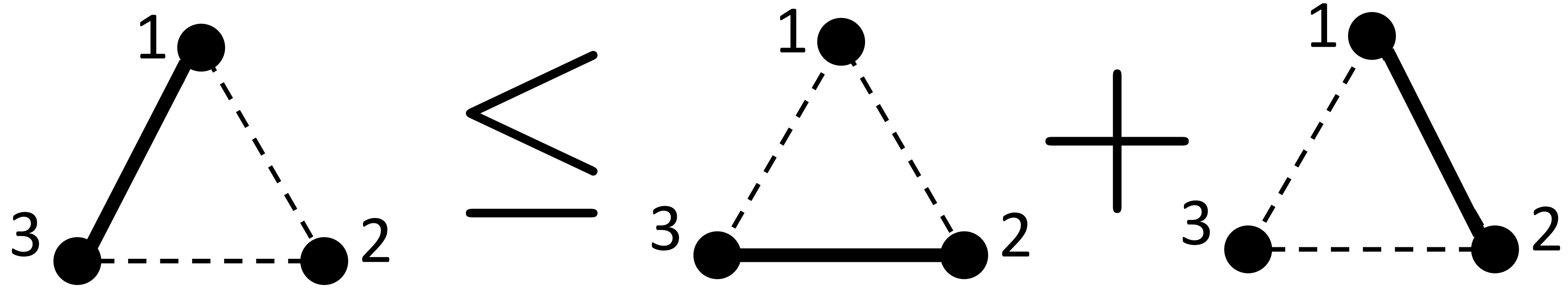
How do we generalize the metric property from 2 graphs to n graphs?

$$d(G_1, G_2) \geq 0,$$
$$d(G_1, G_2) = 0, \text{ iff } G_1, G_2 \text{ are not distinct (isomorphic)},$$
$$d(G_1, G_2) = d(G_2, G_1),$$
$$d(G_1, G_3) \leq d(G_1, G_2) + d(G_2, G_3).$$

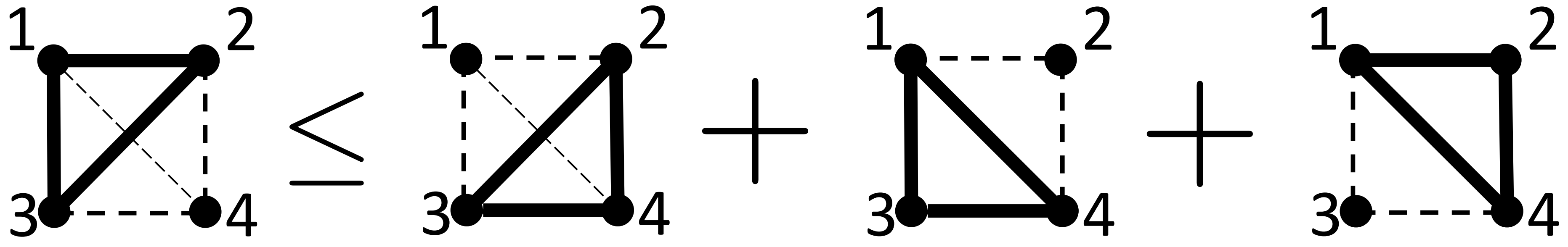
$$d(G_1, ..., G_n) \geq 0,$$
$$d(G_1, ..., G_n) = 0, \text{ iff } G_i \sim G_j \forall i, j$$
$$d(G_1, ..., G_n) = d(\text{permute}(G_1, ..., G_n)),$$
$$d(G_1, ..., G_n) \leq \sum_{i=1}^{n} d(G_1, ..., G_{i-1}, G_{i+1}, ..., G_{n+1}).$$

Mathematical background: n-metrics

$$d(G_1, G_3) \leq d(G_1, G_2) + d(G_2, G_3)$$



$$d(G_1, G_2, G_3) \leq d(G_2, G_3, G_4) + d(G_1, G_3, G_4) + d(G_1, G_2, G_4)$$



Defining an n-metrics

Given a metric for two graphs, why can we not simply define

$$d(G_1, \ldots, G_n) = \sum_{(i,j)} d(G_i, G_j) \ \ ?$$

This does define an n-metric. However, if we look e.g. at what would happen if we used the Chemical distance, we quickly notice we cannot guarantee consistent alignments.

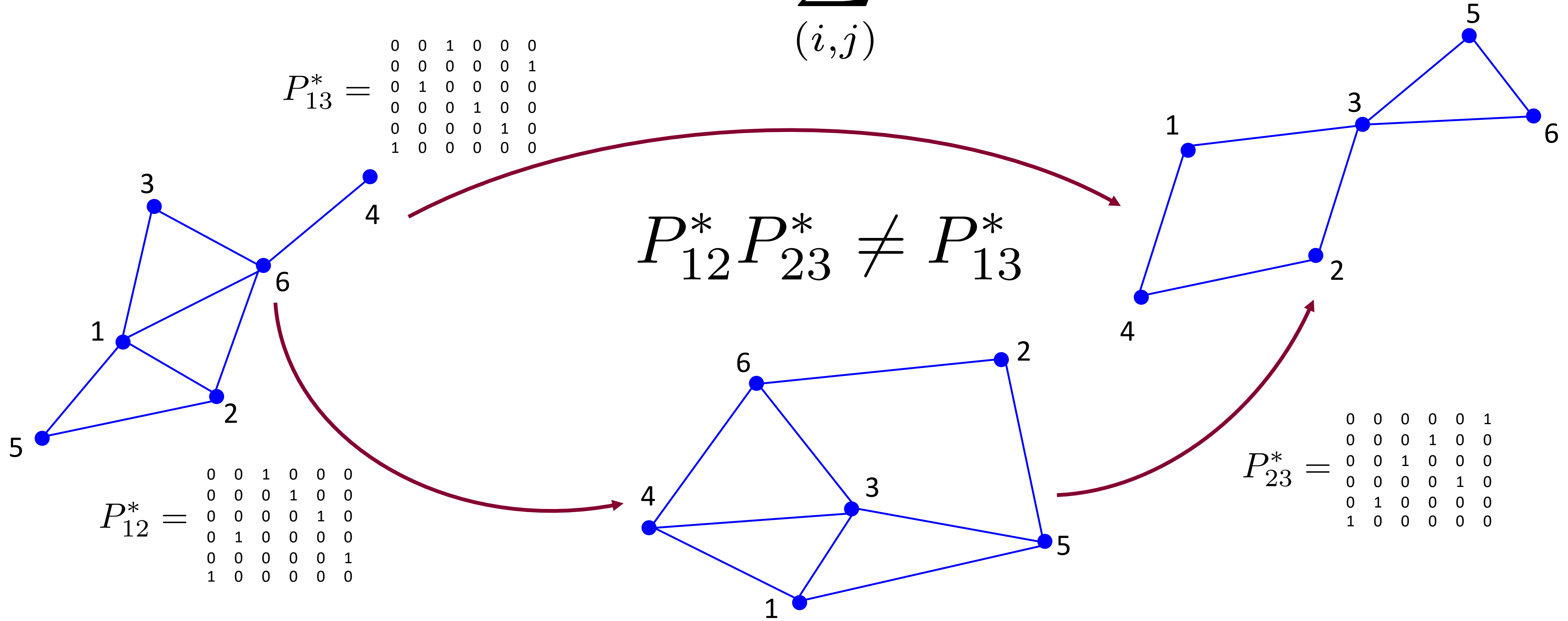
$$d(G_1, \ldots, G_n) = \min_{P_{i,j} \in \Pi \forall i,j} \sum_{(i,j)} \|A_i P_{i,j} - P_{i,j} A_j\|_F$$

$$P^*_{i,j} ? = ? P^*_{i,k} P^*_{k,j}$$

Defining an n-metrics

Given a metric for two graphs, why can we not simply define

$$d(G_1,\ldots,G_n)=\sum_{(i,j)} d(G_i,G_j) \quad ?$$



$$P^*_{13}=\begin{matrix}0&0&1&0&0&0\\0&0&0&0&0&1\\0&1&0&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\1&0&0&0&0&0\end{matrix}$$

$$P^*_{12}P^*_{23}\neq P^*_{13}$$

$$P^*_{12}=\begin{matrix}0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&1&0&0&0&0\\0&0&0&0&0&1\\1&0&0&0&0&0\end{matrix}$$

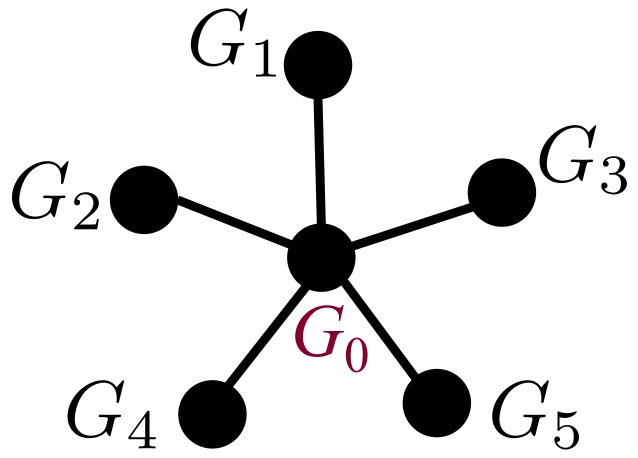
$$P^*_{23}=\begin{matrix}0&0&0&0&0&1\\0&0&0&1&0&0\\0&0&1&0&0&0\\0&0&0&0&1&0\\0&1&0&0&0&0\\1&0&0&0&0&0\end{matrix}$$

Defining an n-metrics

Let $d(A,B)$ be a metric for two graphs. An easy way to obtain an n-metric is to define

$$d(G_1, \ldots, G_n) = \min_{G_0} \sum_{i=1}^{n} d(G_i, G_0)$$

This is called the Fermat distance associated with d.



We want G_0 to be close to all of the G_i ‘s. If we can find such a G_0, then the graphs are similar.

Defining an n-metrics

Let us look at the Fermat distance associated with the Chemical distance: $d(G_1, G_2) = \min_{P \in \Pi} \|A_1 P - P A_2\|$.

$$d(G_1, \ldots, G_n) = \min_{B, P_i \in \Pi \forall i} \sum_{i=1}^{n} \|A_i P_i - P_i B\|$$

Let $P^*_{i,j} = P^*_i {P^*_j}^\top$ then,

$$P^*_{i,j} = P^*_i ({P^*_k}^\top P^*_k) {P^*_j}^\top = P^*_i {P^*_j}^\top = P^*_{i,k} P^*_{k,j}$$

Consistency is easy to achieve! The difficulty is that, even if we relax Π to be a convex set, the problem is still non-convex, and hence not easy to solve exactly.

Defining an n-metrics

Instead we define

$$d(G_1, \ldots, G_n) = \min_{P \in S} \frac{1}{2} \sum_{i,j=1}^{n} \|A_i P_{i,j} - P_{i,j} A_j\|$$

$$S = \{P_{i,j} \in \Pi : P_{i,j} P_{j,k} = P_{i,k}, P_{i,i} = I\}$$

Theorem [Safavi & Bento 2018]: d is an n-metric.

Defining an n-metrics

The set S can be defined in several equivalent ways. Let all the graphs have m nodes, and let

$$\mathbf{P} = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & \cdots \\ P_{2,1} & P_{2,2} & P_{2,3} & \cdots \\ P_{3,1} & P_{3,2} & P_{3,3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

then

$$S = \{P_{i,j} \in \Pi : \mathrm{rank}(\mathbf{P}) = m, P_{i,i} = I\}$$
$$S = \{P_{i,j} \in \Pi : \mathbf{P} \succeq 0, P_{i,i} = I\}$$
$$S = \{P_{i,j} \in \Pi : P_{i,j}P_{j,k} = P_{i,k}, P_{i,i} = I\}$$

Defining an n-metrics

These different representations automatically lead to different relaxations of the original n-metric (related relaxations have been proposed before).

$$d(G_1, \ldots, G_n) = \min_{\substack{P_{i,j} \in \mathcal{C} \\ P_{i,i} = I \\ \mathbf{P} \succeq 0}} \frac{1}{2} \sum_{i,j \in [n]} \|A_i P_{i,j} - P_{i,j} A_j\|$$

$$d(G_1, \ldots, G_n) = \min_{\substack{P_{i,j} \in \mathcal{C} \\ P_{i,i} = I \\ \|\mathbf{P}\|_* \leq mn}} \frac{1}{2} \sum_{i,j \in [n]} \|A_i P_{i,j} - P_{i,j} A_j\|$$

$$\mathcal{C} = \text{some convex set of matrices}$$

Defining an n-metrics

Note that in

$$d(G_1, ..., G_n) = \min_{\substack{P_{i,j} \in \mathcal{C} \\ P_{i,i} = I \\ \mathbf{P} \succeq 0}} \frac{1}{2} \sum_{i,j \in [n]} \|A_i P_{i,j} - P_{i,j} A_j\|$$

we require that $P_{i,j} = P_{j,i}^{\top}$ but not in

$$d(G_1, ..., G_n) = \min_{\substack{P_{i,j} \in \mathcal{C} \\ P_{i,i} = I \\ \|\mathbf{P}\|_* \leq mn}} \frac{1}{2} \sum_{i,j \in [n]} \|A_i P_{i,j} - P_{i,j} A_j\|$$

Defining an n-metrics

A typical choice for $\mathcal{C}$ is, for example, the set of doubly stochastic matrices:

$$\mathcal{C} = \{P \in \mathbb{R}^{m \times m} : P\mathbf{1} = \mathbf{1}, P^\top \mathbf{1} = \mathbf{1}, P \geq 0\}$$

Theorem [Safavi & Bento 2018]: For this choice of $\mathcal{C}$, both maps are n-metrics.

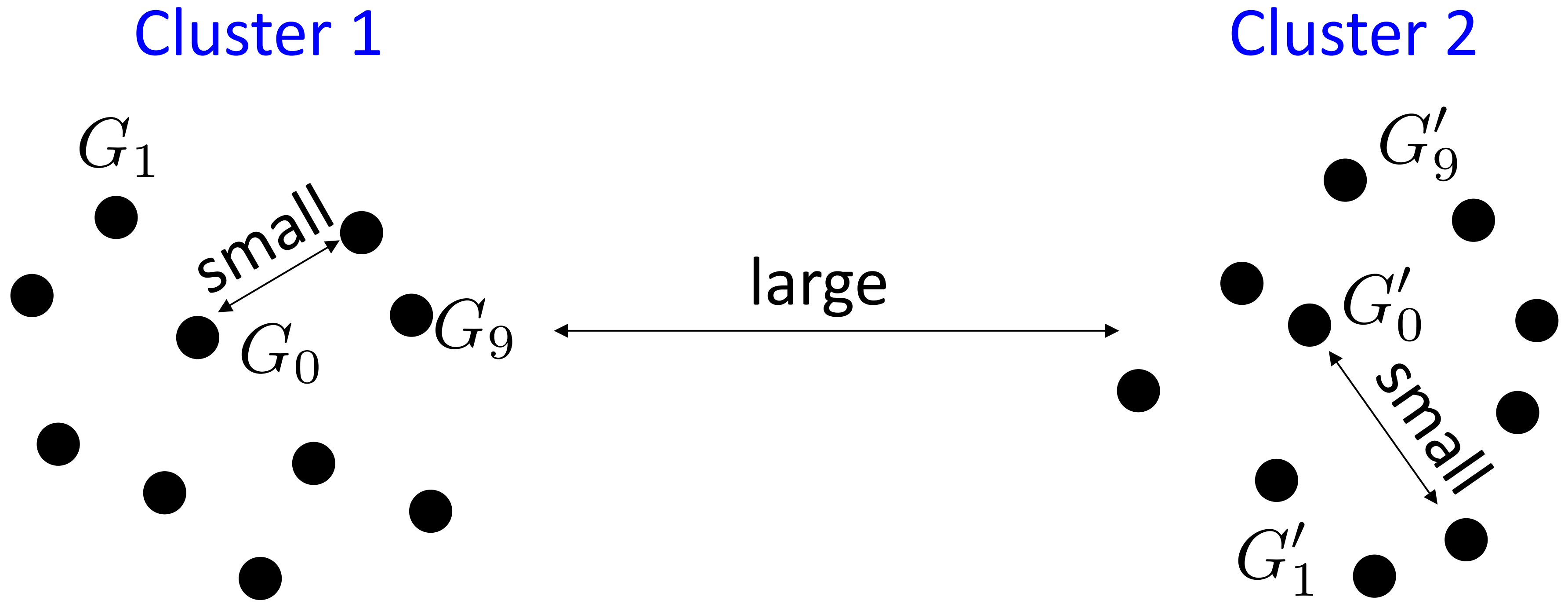
Defining an n-metrics

It is easy to compute $d(G_1, \ldots, G_n)$ when we relax the consistency constraint and also relax the set S to $\mathcal{C}$.

```
cvx_begin
   variable P(n*k,n*k)
   s = 0;
   for i = 1:k
      for j = 1:k
         s=s+norm(A(:,:,i)*P([1:n]+n*(i-1) , [1:n]+n*(j-1) )-P([1:n]+n*(i-1) , [1:n] +  n*(j-1) )*A(:,:,j));
      end
   end
   minimize (0.5*s   )
   subject to
               P == semidefinite(n*k); diag(P) == 1;
         for i = 1:k
            for j = 1:k
               P([1:n] +  n*(i-1) , [1:n] +  n*(j-1) ) >= 0;
               sum(P([1:n]+n*(i-1) , [1:n]+n*(j-1) ))==1;  sum(P([1:n]+n*(i-1) , [1:n] +  n*(j-1) )') == 1;
            end
         end
cvx_end
```

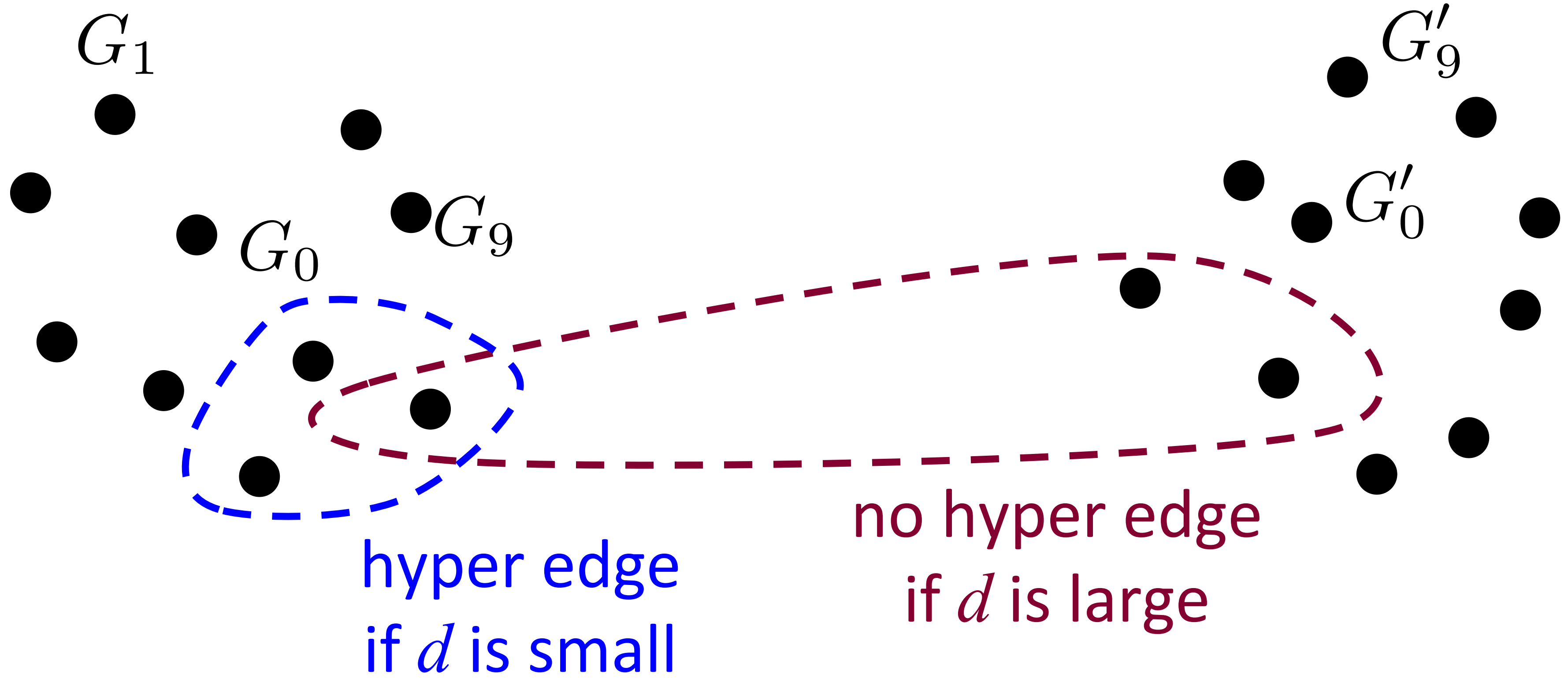
Numerical experiment: clustering graphs

Ioannidis & Bento 2018 show that metrics can cluster graphs better than non-metrics. Here we test if this is also the case for n-metrics, $n > 2$.



Numerical experiment: clustering graphs

Ioannidis & Bento 2018 show that metrics can cluster graphs better than non-metrics. Here we test if this is also the case for n-metrics, $n > 2$.

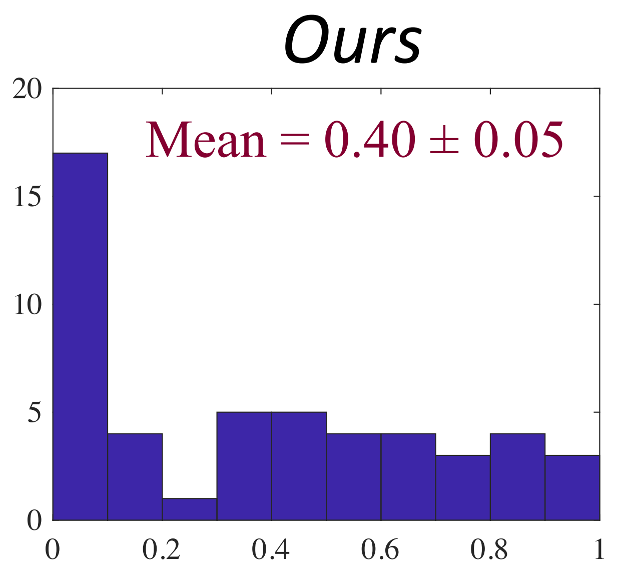


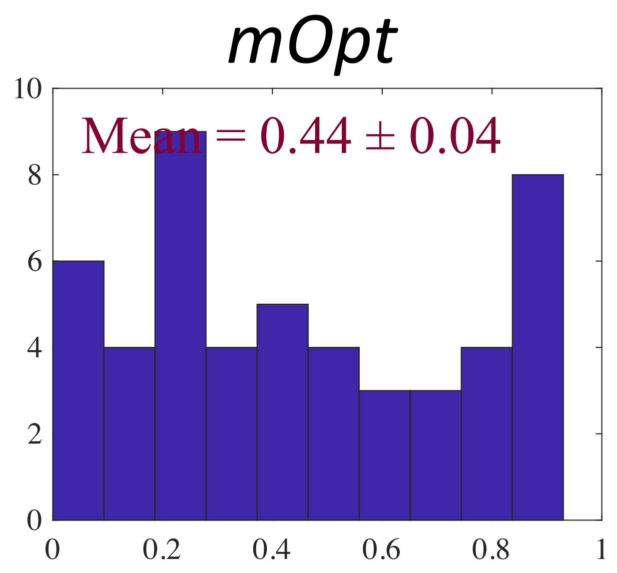
Numerical experiment: clustering graphs

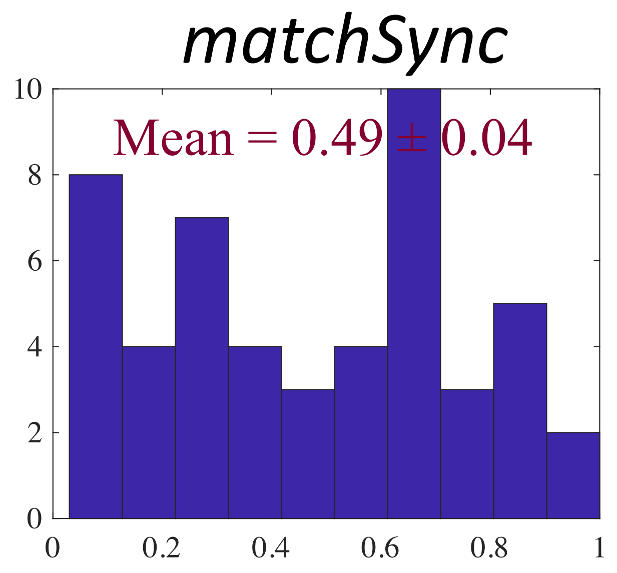
Ioannidis & Bento 2018 show that metrics can cluster graphs better than non-metrics. Here we test if this is also the case for n-metrics, $n > 2$.

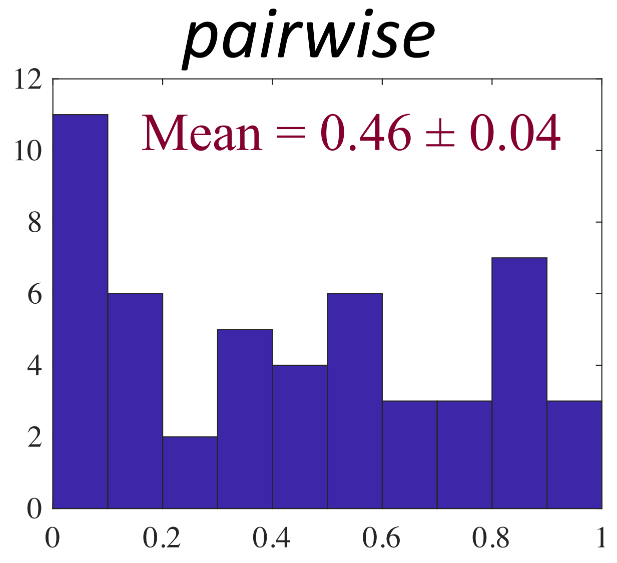
We compute the hyper-edges using our n-metric and 3 other distances (not all proven to be n-metrics): *matchSync* [Pachauriet al., 2013], *mOpt* [Yan et al., 2015], *pairwise*. We partition the hyper graph into as many equal-sized parts as possible using a min-hypergraph-cut algorithm by Vazquez 2009.

Numerical experiment: clustering graphs









Improving the generalized Δ-inequality

In fact, our theorems, for both permutations and relaxations over doubly-stochastic matrices, hold for a more stringent notion of (n, r)-metric.

$$
\begin{aligned}
&d(G_1, \ldots, G_n) \geq 0, \\
&d(G_1, \ldots, G_n) = 0, \text{ iff } G_i \sim G_j \forall i, j \\
&d(G_1, \ldots, G_n) = d(\text{permute}(G_1, \ldots, G_n)), \\
&r d(G_1, \ldots, G_n) \leq \sum_{i=1}^{n} d(G_1, \ldots, G_{i-1}, G_{i+1}, \ldots, G_{n+1})
\end{aligned}
$$

Improving the generalized Δ-inequality

Theorem [Safavi & Bento 2018]: The following three maps, are $(n, n/4)$-metrics, for n large enough, and the set of doubly stochastic matrices.

$$d(G_1, \ldots, G_n) = \min_{P \in S} \frac{1}{2} \sum_{i,j=1}^{n} \|A_i P_{i,j} - P_{i,j} A_j\|$$

$$S = \{P_{i,j} \in \Pi : P_{i,j} P_{j,k} = P_{i,k}, P_{i,i} = I\}$$

$$d(G_1, \ldots, G_n) = \min_{\substack{P_{i,j} \in \mathcal{C} \\ P_{i,i} = I \\ \mathbf{P} \succeq 0}} \frac{1}{2} \sum_{i,j \in [n]} \|A_i P_{i,j} - P_{i,j} A_j\|$$

$$d(G_1, \ldots, G_n) = \min_{\substack{P_{i,j} \in \mathcal{C} \\ P_{i,i} = I \\ \|\mathbf{P}\|_* \leq mn}} \frac{1}{2} \sum_{i,j \in [n]} \|A_i P_{i,j} - P_{i,j} A_j\|$$

Relation with existing work and future work

Several authors, e.g. [Daniilidis et al. 2015] and [Guibas et al. 2013], formulate multi-graph matching in a way that is related to ours.

$$d(G_1, ..., G_n) = \min_{\substack{P_{i,j} \in \mathcal{C} \\ P_{i,i} = I}} \lambda \|\mathbf{P}\|_* + \frac{1}{2} \sum_{i,j \in [n]} \|A_i P_{i,j} - P_{i,j} A_j\|$$

$$d(G_1, ..., G_n) = \min_{\substack{P_{i,j} \in \mathcal{C} \\ P_{i,i} = I}} \lambda \|\mathbf{P}\|_* - \frac{1}{2} \sum_{i,j \in [n]} \langle A_i P_{i,j}, P_{i,j} A_j \rangle$$

Are these n-metrics?

Please cite this tutorial by citing

```
@article{safavi2018admmtutorial, title={How should we (correctly)
compare n networks?}, note={Open Data Science Conference}, author={Safavi, Sam and Bento, Jos{\'e}},
year={2019} }

@article{safavi2018admmtutorial, title={Graph metric spaces}, note={SDM Tutorials}, author={Bento,
Jos{\'e} and Eliassi-Rad, Tina  and Ioannidis, Stratis and Torres, Leo}, year={2019} }

@inproceedings{bento2018family,
  title={A family of tractable graph distances},
  author={Bento, Jose and Ioannidis, Stratis},
  booktitle={Proceedings of the 2018 SIAM International Conference on Data Mining},
  pages={333--341},
  year={2018},
  organization={SIAM}
}

@inproceedings{safavi2019tractable,
  title={Tractable n-Metrics for Multiple Graphs},
  author={Safavi, Sam and Bento, Jose},
  booktitle={International Conference on Machine Learning},
  pages={5568--5578},
  year={2019}
}
```

Thank you !