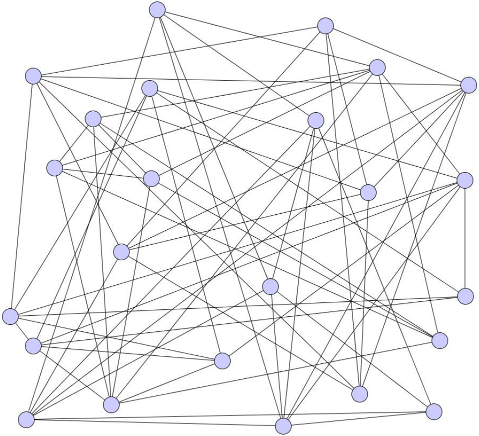
Clustering and Comparing Clusterings

Robert Görke   Dorothea Wagner   Silke Wagner

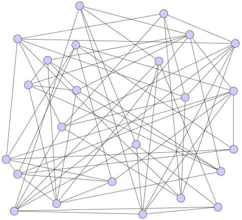
University of Karlsruhe
Faculty of Computer Science
Department of Theoretical Computer Science

Why Cluster?



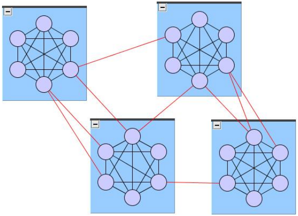
Why Cluster?

- Need for structural information about a network



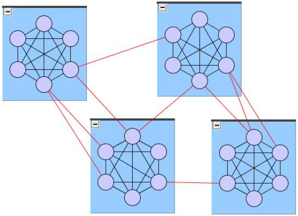
Why Cluster?

- Need for structural information about a network



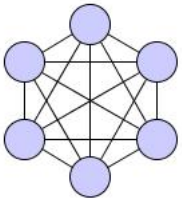
Why Cluster?

- Need for structural information about a network
- Most applications on large networks fall within two cases



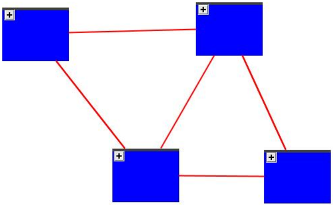
Why Cluster?

- Need for structural information about a network
- Most applications on large networks fall within two cases
  - Interested in small section (e.g. for queries,...)
    $\longrightarrow$ reduction



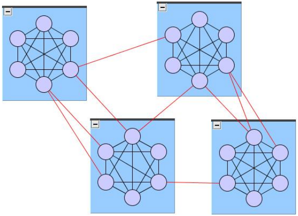
Why Cluster?

- Need for structural information about a network
- Most applications on large networks fall within two cases
  - Interested in small section (e.g. for queries,...)
    $\longrightarrow$ reduction
  - Interested in coarse structure (e.g. for visualization)
    $\longrightarrow$ abstraction



Why Cluster?

- Need for structural information about a network
- Most applications on large networks fall within two cases
  - Interested in small section (e.g. for queries,...)
    $\longrightarrow$ reduction
  - Interested in coarse structure (e.g. for visualization)
    $\longrightarrow$ abstraction
- Detect groups/clusters as basic structural units



How to Cluster

Abstract Idea

Given: (un)weighted, (un)directed graph $G = (V, E)$
Find: partition of V into clusters $C_1, \ldots, C_k$ such that

How to Cluster

Abstract Idea

Given: (un)weighted, (un)directed graph $G = (V, E)$

Find: partition of V into clusters $C_1, \ldots, C_k$ such that

(1) intra-cluster density is maximized

How to Cluster
Abstract Idea

Given: (un)weighted, (un)directed graph $G = (V, E)$
Find: partition of V into clusters $C_1, \dots, C_k$ such that

(1) intra-cluster density is maximized

(2) inter-cluster sparsity is maximized

How to Cluster
Abstract Idea

Given: (un)weighted, (un)directed graph $G = (V, E)$
Find: partition of V into clusters $C_1, \ldots, C_k$ such that

(1) intra-cluster density is maximized

(2) inter-cluster sparsity is maximized

Typically, a clustering algorithm tries to maximize a quality function that captures (1) and/or (2)

How to Cluster
Quality Functions

- Coverage:

$$c(\mathcal{C}) = \frac{\text{\# intra-cluster edges}}{\text{\# edges}}$$

How to Cluster
Quality Functions

- Coverage:

$$c(\mathcal{C}) = \frac{\text{\# intra-cluster edges}}{\text{\# edges}}$$

- Performance:

$$c(\mathcal{P}) = \frac{\text{\# intra-cluster edges} + \text{\# absent inter-cluster edges}}{\text{\# point pairs}}$$

How to Cluster
Quality Functions

- Coverage:

$$c(\mathcal{C}) = \frac{\#\ \text{intra-cluster edges}}{\#\ \text{edges}}$$

- Performance:

$$c(\mathcal{P}) = \frac{\#\ \text{intra-cluster edges} + \#\ \text{absent inter-cluster edges}}{\#\ \text{point pairs}}$$

- Conductance: measure for sparse cuts (bottlenecks)

How to Cluster
Quality Functions

- Coverage:

$$c(\mathcal{C}) = \frac{\text{\# intra-cluster edges}}{\text{\# edges}}$$

- Performance:

$$c(\mathcal{P}) = \frac{\text{\# intra-cluster edges} + \text{\# absent inter-cluster edges}}{\text{\# point pairs}}$$

- Conductance: measure for sparse cuts (bottlenecks)
- ...

How to Cluster

Quality Functions

- Coverage:

$$c(\mathcal{C}) = \frac{\text{\# intra-cluster edges}}{\text{\# edges}}$$

- Performance:

$$c(\mathcal{P}) = \frac{\text{\# intra-cluster edges} + \text{\# absent inter-cluster edges}}{\text{\# point pairs}}$$

- Conductance: measure for sparse cuts (bottlenecks)
- ...
- $\overline{\mathrm{QF}}_1 = \mathrm{QF} - \mathrm{E}[\mathrm{QF}] \qquad \overline{\mathrm{QF}}_2 = \frac{\mathrm{QF}}{\mathrm{E}[\mathrm{QF}]}$

How to Cluster

Quality Functions

Finding global optimum of quality function is (in general)

NP-hard

How to Cluster

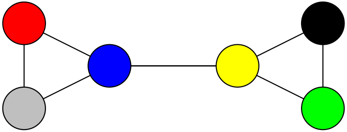
Quality Functions

Finding global optimum of quality function is (in general)
NP-hard

$\Longrightarrow$ Approximate with greedy algorithms

How to Cluster

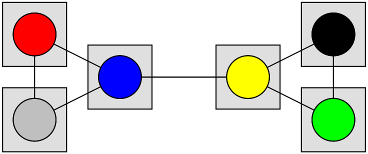
Methodologies



How to Cluster

Methodologies

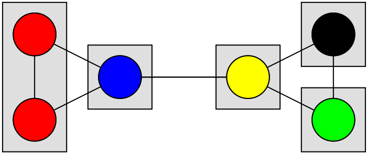
- Bottom-up: Start with *singletons*



How to Cluster

Methodologies

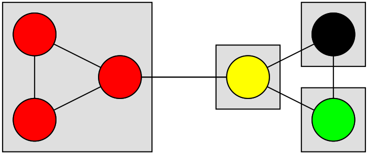
- Bottom-up: Start with *singletons* ⇒ merge clusters



How to Cluster

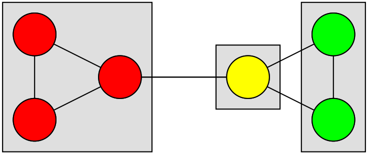
Methodologies

- Bottom-up: Start with *singletons* $\Rightarrow$ merge clusters



How to Cluster
Methodologies

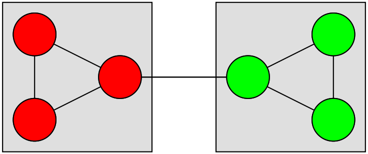
- Bottom-up: Start with *singletons* $\Rightarrow$ merge clusters



How to Cluster

Methodologies

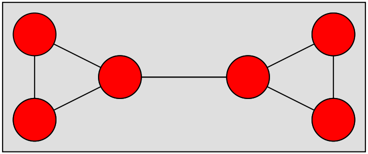
- Bottom-up: Start with *singletons* $\Rightarrow$ merge clusters



How to Cluster

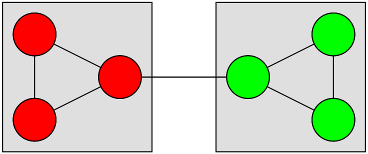
Methodologies

- Bottom-up: Start with *singletons* ⇒ merge clusters
- Top-down: Start with the *one-cluster*



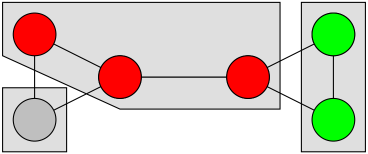
How to Cluster
Methodologies

- Bottom-up: Start with *singletons* $\Rightarrow$ merge clusters
- Top-down: Start with the *one-cluster* $\Rightarrow$ split clusters



How to Cluster
Methodologies

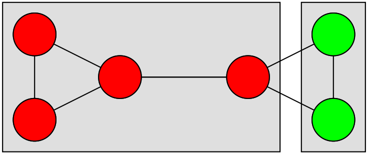
- Bottom-up: Start with *singletons* ⇒ merge clusters
- Top-down: Start with the *one-cluster* ⇒ split clusters
- Morphing: Start with random clustering



How to Cluster

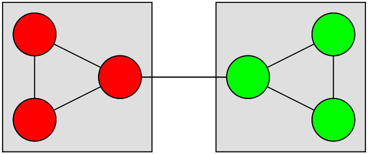
Methodologies

- Bottom-up: Start with *singletons* $\Rightarrow$ merge clusters
- Top-down: Start with the *one-cluster* $\Rightarrow$ split clusters
- Morphing: Start with random clustering $\Rightarrow$ migrate nodes



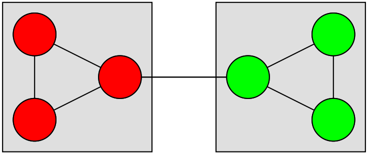
How to Cluster
Methodologies

- Bottom-up: Start with *singletons* $\Rightarrow$ merge clusters
- Top-down: Start with the *one-cluster* $\Rightarrow$ split clusters
- Morphing: Start with random clustering $\Rightarrow$ migrate nodes



How to Cluster
Methodologies

- Bottom-up: Start with *singletons* ⇒ merge clusters
- Top-down: Start with the *one-cluster* ⇒ split clusters
- Morphing: Start with random clustering ⇒ migrate nodes



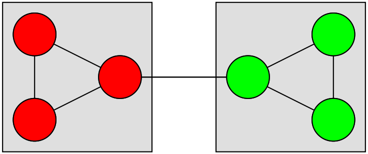
Other techniques

- Spectral clustering (eigendecomposition of adjacency matrix)

How to Cluster

Methodologies

- Bottom-up: Start with *singletons* $\Rightarrow$ merge clusters
- Top-down: Start with the *one-cluster* $\Rightarrow$ split clusters
- Morphing: Start with random clustering $\Rightarrow$ migrate nodes



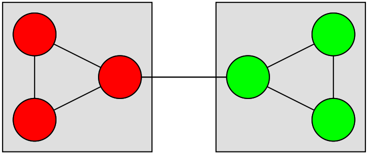
Other techniques

- Spectral clustering (eigendecomposition of adjacency matrix)
- Identifying structures directly (Cliques, Coresets,. . .)

How to Cluster

Methodologies

- Bottom-up: Start with *singletons* $\Rightarrow$ merge clusters
- Top-down: Start with the *one-cluster* $\Rightarrow$ split clusters
- Morphing: Start with random clustering $\Rightarrow$ migrate nodes



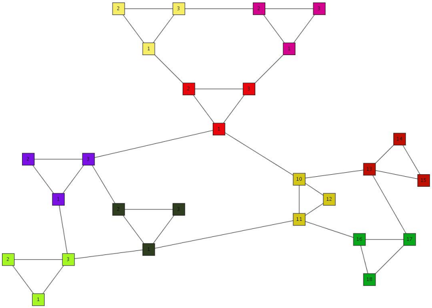
Other techniques

- Spectral clustering (eigendecomposition of adjacency matrix)
- Identifying structures directly (Cliques, Coresets,. . .)
- . . .

Granularity

Which solution is desired?

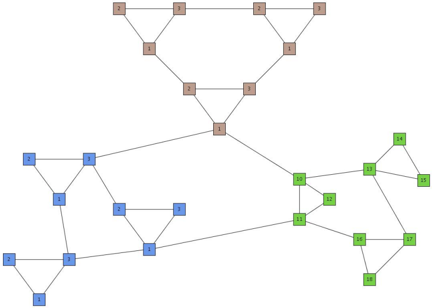
fine



Granularity

Which solution is desired?

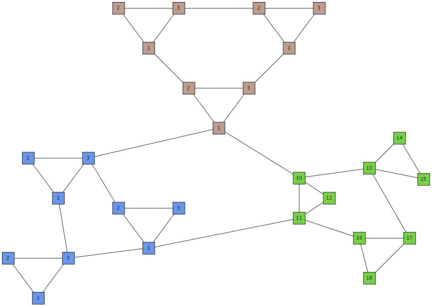
coarse



Granularity

Which solution is desired?

coarse



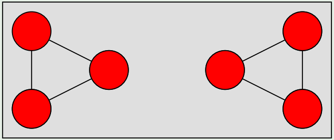
algorithmic optimum $\Leftrightarrow$ desired clustercount

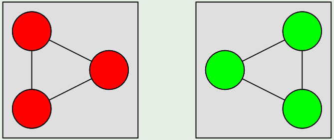
Cheating a Criterion

Any single optimization criterion can be fooled

Example (Coverage (very simple))

The following two clusterings have the same coverage value



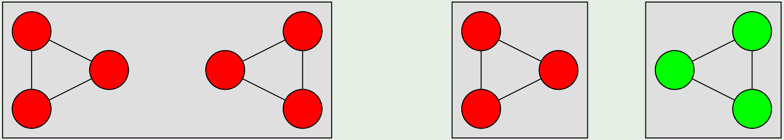


Cheating a Criterion

Any single optimization criterion can be fooled

Example (Coverage (very simple))

The following two clusterings have the same coverage value



Similar (more sophisticated) examples exist for any criterion

Problems and Questions
of Static Clustering

- Which criterion works well for which kind of graph?

Problems and Questions

of Static Clustering

- Which criterion works well for which kind of graph?
- Best method/algorithm for optimizing a certain criterion?

Problems and Questions
of Static Clustering

- Which criterion works well for which kind of graph?
- Best method/algorithm for optimizing a certain criterion?

- Comparability of clusterings/algorithms
  - How similar are two clustering results?
  - How close is a result to optimal solution (if known)?

Problems and Questions

of Static Clustering

- Which criterion works well for which kind of graph?
- Best method/algorithm for optimizing a certain criterion?

- Comparability of clusterings/algorithms
  - How similar are two clustering results?
  - How close is a result to optimal solution (if known)?
  - $\Rightarrow$ need for similarity/distance measures for clusterings

Dynamic Situation

Given: Graph $G = (V, E)$; clustering algorithm A; update operation $\Delta : G \mapsto G' = (V', E')$

Dynamic Situation

Given: Graph $G = (V, E)$; clustering algorithm A; update operation $\Delta : G \mapsto G' = (V', E')$
Possible updates:

Dynamic Situation

Given: Graph $G = (V, E)$; clustering algorithm A; update operation $\Delta : G \mapsto G' = (V', E')$

Possible updates:

- insertion of an edge
- deletion of an edge

Dynamic Situation

Given: Graph $G = (V, E)$; clustering algorithm A; update operation $\Delta : G \mapsto G' = (V', E')$
Possible updates:

- insertion of an edge
- deletion of an edge
- insertion of a node (and its incident edges)
- deletion of a node (and its incident edges)

Dynamic Situation

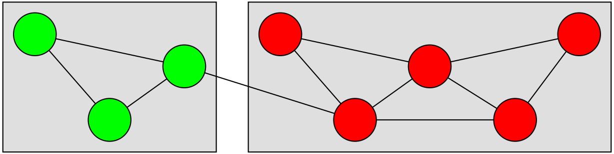
Given: Graph $G = (V, E)$; clustering algorithm A; update operation $\Delta : G \mapsto G' = (V', E')$
Possible updates:

- insertion of an edge
- deletion of an edge
- insertion of a node (and its incident edges)
- deletion of a node (and its incident edges)

Find: efficient method for calculating $A(\Delta G)$ from $A(G)$

Typical Clustering Dynamics

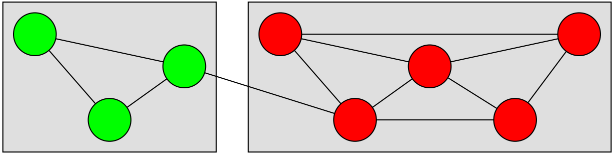
Consistent with intuition:



Typical Clustering Dynamics

Consistent with intuition:

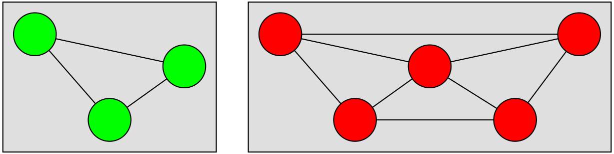
- insertion of intra-cluster edge strengthens cluster



Typical Clustering Dynamics

Consistent with intuition:

- insertion of intra-cluster edge strengthens cluster
- deletion of inter-cluster edge strengthens disjunction

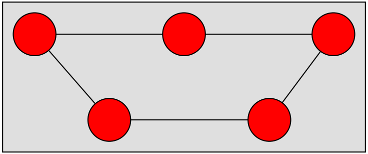


Typical Clustering Dynamics

Consistent with intuition:

- insertion of intra-cluster edge strengthens cluster
- deletion of inter-cluster edge strengthens disjunction

Contrary to intuition:



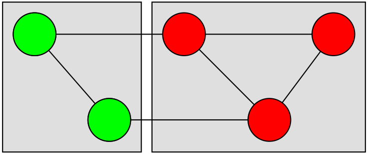
Typical Clustering Dynamics

Consistent with intuition:

- insertion of intra-cluster edge strengthens cluster
- deletion of inter-cluster edge strengthens disjunction

Contrary to intuition:

- insertion of intra-cluster edge can cause splitting of cluster



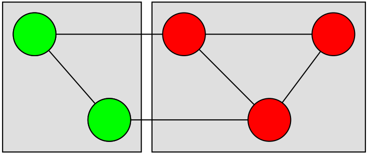
Typical Clustering Dynamics

Consistent with intuition:

- insertion of intra-cluster edge strengthens cluster
- deletion of inter-cluster edge strengthens disjunction

Contrary to intuition:

- insertion of intra-cluster edge can cause splitting of cluster
- deletion of inter-cluster edge can cause merge of clusters



Clustering Issues

- All problems inherited from static clustering

Clustering Issues

- All problems inherited from static clustering
- New problems due to dynamics

Clustering Issues

- All problems inherited from static clustering
- New problems due to dynamics
  - Can we calculate the exact update?

Clustering Issues

- All problems inherited from static clustering
- New problems due to dynamics
  - Can we calculate the exact update?
  - Complexity?

Clustering Issues

- All problems inherited from static clustering
- New problems due to dynamics
  - Can we calculate the exact update?
  - Complexity?
  - Are there good approximations?

Clustering Issues

- All problems inherited from static clustering
- New problems due to dynamics
  - Can we calculate the exact update?
  - Complexity?
  - Are there good approximations?
  - *Distance*: approximation $\leftrightarrow$ reclustering?

Update of a Clustering

Running Time

Example (Simple Algorithm)

Clustering algorithm A: connected components $\widehat{=}$ clusters

Update of a Clustering

Running Time

Example (Simple Algorithm)

Clustering algorithm A: connected components $\widehat{=}$ clusters

Full run: $O(m+n)$

Update of a Clustering
Running Time

Example (Simple Algorithm)

Clustering algorithm A: connected components $\widehat{=}$ clusters
Full run: $O(m+n)$
Complexity of updates:

Update of a Clustering

Running Time

Example (Simple Algorithm)

Clustering algorithm A: connected components $\hat{=}$ clusters
Full run: $O(m+n)$
Complexity of updates:

- Edge deletion: $O(\sqrt{n})$

Update of a Clustering

Running Time

Example (Simple Algorithm)

Clustering algorithm A: connected components $\widehat{=}$ clusters

Full run: $O(m+n)$

Complexity of updates:

- Edge deletion: $O(\sqrt{n})$
- Edge insertion: $O(\sqrt{n})$

Update of a Clustering

Running Time

Example (Simple Algorithm)

Clustering algorithm A: connected components $\widehat{=}$ clusters

Full run: $O(m+n)$

Complexity of updates:

- Edge deletion: $O(\sqrt{n})$
- Edge insertion: $O(\sqrt{n})$

Most clustering criterions are higly non-trivial!

Similarity measures for clusterings

Existing similarity / distance measures can be divided into 3 groups:

1. measures based on **counting pairs**
2. measures based on **set cardinality**
3. measures based on **mutual information**

Counting Pairs

- Count the number of node pairs that are grouped in the same way by both clusterings

Counting Pairs

- Count the number of node pairs that are grouped in the same way by both clusterings
- Example: Rand's index (Rand, 1971)

$$\mathcal{R}(\mathcal{C}, \mathcal{C}') = \frac{2(n_{11} + n_{00})}{n(n-1)}$$

where n_{11}= # pairs in the same cluster under both, $\mathcal{C}$ and $\mathcal{C}'$
n_{00}= # pairs in different clusters under $\mathcal{C}$ and $\mathcal{C}'$

Counting Pairs

- Count the number of node pairs that are grouped in the same way by both clusterings
- Example: Rand's index (Rand, 1971)

$$\mathcal{R}(\mathcal{C}, \mathcal{C}') = \frac{2(n_{11} + n_{00})}{n(n-1)}$$

where n_{11}= # pairs in the same cluster under both, $\mathcal{C}$ and $\mathcal{C}'$
n_{00}= # pairs in different clusters under $\mathcal{C}$ and $\mathcal{C}'$

- Problem: $\mathcal{R}(\mathcal{C}, \mathcal{C}') \rightarrow 1$ for $k \rightarrow n$

Set Cardinality

- Find a ”best match” for each cluster and add up the contributions of the matches

Set Cardinality

- Find a ”best match” for each cluster and add up the contributions of the matches
- Example: Van Dongen (2000):

$$\mathcal{D}(\mathcal{C},\mathcal{C}') = 2n - \sum_{i=1}^{k} \max_{j} n_{ij} - \sum_{j=1}^{k'} \max_{i} n_{ij}$$

where $n_{ij} = |C_i \cap C'_j|, i = 1, \dots, k, j = 1, \dots, k'$

Set Cardinality

- Find a ”best match” for each cluster and add up the contributions of the matches
- Example: Van Dongen (2000):

$$\mathcal{D}(\mathcal{C},\mathcal{C}') = 2n - \sum_{i=1}^{k} \max_{j} \, n_{ij} - \sum_{j=1}^{k'} \max_{i} \, n_{ij}$$

where $n_{ij} = |C_i \cap C'_j|, i = 1, \ldots, k, j = 1, \ldots, k'$

- Drawbacks:
    - Depending on n
    - Ignores what happens in unmatched part of the clusters

Mutual Information

- Derived from information theory

Mutual Information

- Derived from information theory
- Entropy of of a clustering:

$$\mathcal{H}(\mathcal{C}) = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Pick node randomly, **uncertainty** *which cluster it is in?*

Mutual Information

- Derived from information theory
- Entropy of of a clustering:

$$\mathcal{H}(\mathcal{C}) = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Pick node randomly, **uncertainty** *which cluster it is in?*

- Mutual information $I(C, C')$: *Knowing cluster C_i of node in clustering $\mathcal{C}$,* **reduction of uncertainty** *about cluster in $\mathcal{C}'$.*

Mutual Information

- Derived from information theory
- Entropy of of a clustering:

$$\mathcal{H}(\mathcal{C}) = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Pick node randomly, **uncertainty** *which cluster it is in?*

- Mutual information $I(C, C')$: *Knowing cluster C_i of node in clustering $\mathcal{C}$,* **reduction of uncertainty** *about cluster in $\mathcal{C}'$.*
- Variation of Information (Meila, 2002):
  $\mathcal{VI}(\mathcal{C}, \mathcal{C}') = \mathcal{H}(\mathcal{C}) + \mathcal{H}(\mathcal{C}') - 2\mathcal{I}(\mathcal{C}, \mathcal{C}')$
  *Information we lose, going from $\mathcal{C}$ to $\mathcal{C}'$ plus extra information we have to gain (***geometric difference***)*

Research Areas

- $\overline{\mathrm{QF}}_1 = \mathrm{QF} - \mathrm{E}[\mathrm{QF}]$ $\qquad$ $\overline{\mathrm{QF}}_2 = \frac{\mathrm{QF}}{\mathrm{E}[\mathrm{QF}]}$

Research Areas

- $\overline{\mathrm{QF}}_1 = \mathrm{QF} - \mathrm{E}[\mathrm{QF}] \qquad \overline{\mathrm{QF}}_2 = \frac{\mathrm{QF}}{\mathrm{E}[\mathrm{QF}]}$
- Formalizing clustering

Research Areas

- $\overline{\mathrm{QF}}_1 = \mathrm{QF} - \mathrm{E}[\mathrm{QF}] \qquad \overline{\mathrm{QF}}_2 = \frac{\mathrm{QF}}{\mathrm{E}[\mathrm{QF}]}$
- Formalizing clustering
- Quality $\Longleftrightarrow$ Silimarity

Research Areas

- $\overline{\mathrm{QF}}_1 = \mathrm{QF} - \mathrm{E}[\mathrm{QF}] \qquad \overline{\mathrm{QF}}_2 = \frac{\mathrm{QF}}{\mathrm{E}[\mathrm{QF}]}$
- Formalizing clustering
- Quality $\Longleftrightarrow$ Silimarity
- Map classes of graphs to suitable clustering techniques
  - Find global optimum in polynomial time
  - Find criterion that *fits* certain classes

End

Thank you!

Thank you!

Questions?