Community evolution

Community evaluation

with ground truth

ground truth

When ground truth is available, the evaluator has prior knowledge of what a community should be

- That is, we know the correct community assignments.
- How do we get networks with ground truth communities?
  - Explicit communities

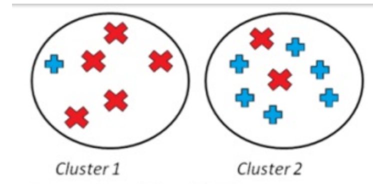
Precision and recall

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

		Actual: Positive	Actual: Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

- **True Positive (TP) :**
  - when **similar points** are assigned to the **same communities**
  - This is considered a correct decision.
- **True Negative (TN) :**
  - when **dissimilar points** are assigned to **different communities**
  - This is considered a correct decision
- **False Negative (FN) :**
  - when **similar points** are assigned to **different communities**
  - This is considered an incorrect decision
- **False Positive (FP)** :
  - when **dissimilar points** are assigned to the **same communities**
  - This is considered an incorrect decision

Example:



$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

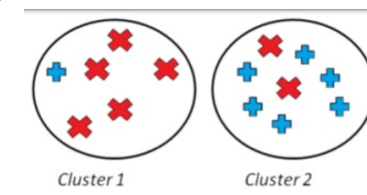
True Positive (TP) :
when similar points are assigned to the same communities
This is considered a correct decision.

For each community, count the pairs of similar points

For cluster 1
- Red points: 5 red points, can form $\binom{5}{2} = \frac{5\times4}{2\times1} = 10$ pairs
- Blue points: 1 blue points, can form 0 pair

$$TP = \binom{5}{2} + \binom{6}{2} + \binom{2}{2} = 26$$

For cluster 2
- Red points: 2 red points, can form $\binom{2}{2} = \frac{2\times1}{2\times1} = 1$ pair
- Blue points: 6 blue points, can form $\binom{6}{2} = \frac{6\times5}{2\times1}$ =15 pairs



$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

False Positive (FP) :
when dissimilar points are assigned to the same communities
This is considered an incorrect decision

For each community, count the pairs of dissimilar points

For cluster 1
- 5 red points and 1 blue point can form $5 \times 1 = 5$ dissimilar pairs

$$FP = (5 \times 1) + (6 \times 2) = 17$$

For cluster 2
- 2 red points and 6 blue points, can form $6 \times 2 = 12$ dissimilar pairs

False Negative (FN) :
when similar points are assigned to different communities
This is considered an incorrect decision

For two communities, count the pairs of similar points

For cluster 1 and cluster 2
- 5 red points in cluster 1 and 2 red points in cluster 2
  - can form $5 \times 2 = 10$ pairs
- 1 blue point in cluster 1 and 6 blue points in cluster 2
  - can form $1 \times 6 = 6$ pairs

$$FN = (5 \times 2) + (6 \times 1) = 16$$

True Negative (TN) :
when dissimilar points are assigned to different communities
This is considered a correct decision

For two communities, count the pairs of dissimilar points

For cluster 1 and cluster 2
- 5 red points in cluster 1 and 6 blue points in cluster 2
  - can form $5 \times 6 = 30$ dissimilar pairs
- 1 blue point in cluster 1 and 2 red points in cluster 2
  - can form $1 \times 2 = 2$ dissimilar pairs

$$TN = (6 \times 5) + (2 \times 1) = 32$$

$$\begin{aligned} TP &= \binom{5}{2} + \binom{6}{2} + \binom{2}{2} = 26, \\ FP &= (5\times1) + (6\times2) = 17, \\ FN &= (5\times2) + (6\times1) = 16, \\ TN &= (6\times5) + (2\times1) = 32. \end{aligned}$$

$$P = \frac{26}{26+17} = 0.60$$ Subtopic

$$R = \frac{26}{26+16} = 0.61$$

Precision meaning

$$P = \frac{TP}{TP + FP}$$

- **True Positive (TP) :**
  - when **similar points** are assigned to the **same communities**
  - This is considered a correct decision.
- **False Positive (FP)** :
  - when **dissimilar points** are assigned to the **same communities**
  - This is considered an incorrect decision
- Larger TP means more similar points are clustered into the same cluster
  - Larger TP means better community
  - Does TP alone give a good measure of community detection? (Hint: cluster all points into 1 community)
- Smaller FP means purer each community is
  - Good communities have small FP
  - Does FP alone give a good measure of community detection? (Hint: split a pure community to multiple communities)

Precision considers both TP and FP, give higher score to communities that assign similar points to the same communities and dissimilar points to different communities. **The larger precision is, the better the communities are.**

Recall meaning

$$R = \frac{TP}{TP + FN}$$

- **True Positive (TP) :**
  - when **similar points** are assigned to the **same communities**
  - This is considered a correct decision.
- **False Negative (FN) :**
  - when **similar points** are assigned to **different communities**
  - This is considered an incorrect decision
- Larger TP means more similar points are clustered into the same cluster
  - Larger TP means better community
  - Does TP alone give a good measure of community detection? (Hint: 1 community)
- FN detects if similar points are assigned to different communities
  - Does FN alone give a good measure of community detection? (Hint: 1 community)

Recall considers both TP and FN. **The larger recall is, the better the communities are.** However, if you cluster all data points into one community, you still get a good recall. That's the problem of recall.

F-measure

- Either P or R measures one aspect of the performance, to integrate them into one measure, we can use the harmonic mean of precision of recall

$$F = 2 \cdot \frac{P \cdot R}{P + R}$$

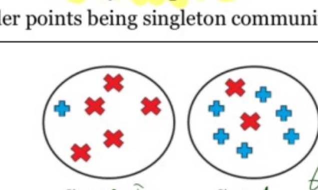
- For example 1
$$F = 2\frac{P \times R}{P + R} = 2 \times \frac{0.6 \times 0.61}{0.6 + 0.61} \approx 0.6$$
- For example 2
$$F = 2\frac{P \times R}{P + R} = 2 \times \frac{0.428 \times 0.428}{0.428 + 0.428} \approx 0.428$$

Purity

- In purity, we assume the majority of a community represents the community
- Hence, we use the label of the majority against the label of each member to evaluate the algorithm
- The purity is then defined as the fraction of instances that have labels equal to the community's majority label

Suppose the algorithm detects K communities $(P_1, \dots P_K)$
Purity= $\frac{\text{number of majoririty instances in } P_1 + \dots + \text{number of majoririty instances in } P_K}{N}$
N is the number of nodes in the network

- The **larger** purity score are, the better the communities
- Purity can be easily tampered
  - consider points being singleton communities (of size 1)



Suppose the algorithm detects K communities $(P_1, \dots P_K)$
Purity= $\frac{\text{number of majoririty instances in } P_1 + \dots + \text{number of majoririty instances in } P_K}{N}$
N is the number of nodes in the network

$$\text{Purity} = \frac{\text{number of majority instances in cluster 1} + \text{number of majority instances in cluster 2}}{N} = \frac{6+5}{14} = 0.78$$

without ground truth

Patterns in dynamic networks(4)

- Decreasing probability of new connections between two nodes with increasing distance
  - Why?
    - Two users with short distance are more likely to know each other or have similar interests than two users with long distance

- Many new connections occur as triadic closures
- What does this indicate?
  - Friend of my friend is my friend
  - Leading to high clustering coefficient

- The density of the graph increases as the network grows
  - density= $\frac{\#\text{edges}}{\#\text{possible edges}}$
  - The number of edges increases faster than the number of nodes does

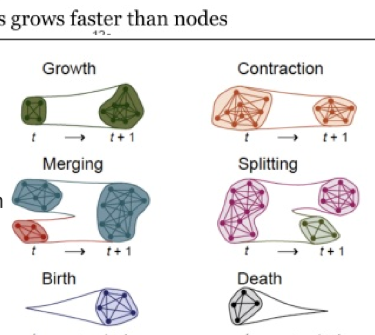
$$E(t) \propto V(t)^{\alpha}$$

- Densification exponent: $1 \le \alpha \le 2$:
  - α =1: linear growth – constant out-degree
  - α =2: quadratic growth – clique

E(t) and V(t) are numbers of edges and nodes respectively at time *t*

In growing networks, diameter shrinks over time
- What does it tell us?
  - As network grows, small-world phenomenon is more obvious
- Why does the diameter shrink?
  - Densification: edges grows faster than nodes

Community evolution



Community detection in evolving networks

Evolutionary clustering

Assume communities change smoothly
Minimize an objective function that considers
- **Snapshot Cost**. Communities at different times (**SC**)
- **Temporal Cost**. How communities evolve (**TC**)

Objective function is defined as

$$Cost = \alpha SC + (1-\alpha)TC$$
$$0 \le \alpha \le 1$$

E.g. If we use spectral clustering for each snapshot

$$\begin{aligned} Cost_t &= \alpha\ SC + (1-\alpha)\ TC \\ &= \alpha\ Tr\left(X_t^T L X_t\right) + (1-\alpha)\ TC \end{aligned}$$

One choice of TC is $TC = ||X_t - X_{t-1}||^2$