# Tổng quan về thuật toán Louvain

Thuật toán **Louvain** là một trong những thuật toán phổ biến nhất để phát hiện cộng đồng trong đồ thị. Nó được thiết kế để tối ưu hóa **Modularity**, một chỉ số đo lường chất lượng của phân hoạch cộng đồng. Louvain có ưu điểm là nhanh, có thể áp dụng cho đồ thị lớn và có khả năng phát hiện các cấu trúc phân cấp trong đồ thị.

## Quy trình cơ bản của Louvain

1. **Khởi tạo**:
    - Mỗi nút ban đầu được coi là một cộng đồng riêng biệt.
2. **Giai đoạn 1 - Tối ưu hóa cục bộ**:
    - Với mỗi nút, thuật toán thử di chuyển nó vào cộng đồng của các nút láng giềng sao cho **Modularity** tăng lên.
    - Quy trình này lặp lại cho đến khi không còn thay đổi nào làm tăng Modularity.
3. **Giai đoạn 2 - Tổng hợp cộng đồng**:
    - Mỗi cộng đồng được coi như một nút duy nhất trong đồ thị mới.
    - Trọng số cạnh giữa các cộng đồng được tính dựa trên số cạnh ban đầu.
4. **Tiếp tục lặp lại**:
    - Quay lại bước 1 với đồ thị mới cho đến khi Modularity không tăng nữa.

## Đặc điểm nổi bật

- **Thời gian chạy**: Louvain rất nhanh nhờ chiến lược tối ưu hóa cục bộ và gộp cộng đồng.
- **Phân cấp**: Thuật toán có thể tạo ra một cấu trúc phân cấp cộng đồng.
- **Tối ưu hóa Modularity**: Louvain hoạt động rất tốt với các đồ thị có cấu trúc cộng đồng rõ ràng.

# Tóm tắt các chỉ số đánh giá việc phát hiện cộng đồng

## 1. Modularity

- **Ý nghĩa**: Đo mức độ nút bên trong cộng đồng kết nối chặt chẽ so với nút giữa các cộng đồng.
- **Công thức**:

$$Q = \frac{1}{2m} \sum_{i,j} \left[ A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

  Trong đó:

  - $A_{ij}$: Trọng số cạnh giữa nút $i$ và $j$.
  - $k_i$: Tổng trọng số các cạnh của nút $i$.
  - $m$: Tổng trọng số các cạnh trong đồ thị.
  - $\delta(c_i, c_j)$: Bằng 1 nếu $i, j$ thuộc cùng một cộng đồng.

- **Phạm vi**: $[-1, 1]$. Giá trị càng cao càng tốt (gần 1).

## 2. Conductance

- **Ý nghĩa**: Đo mức độ giao cắt giữa cộng đồng và phần còn lại của đồ thị.
- **Công thức**:

$$\phi(S) = \frac{\text{Số cạnh cắt giữa } S \text{ và } \bar{S}}{\min(\text{Volume của } S, \text{Volume của } \bar{S})}$$

- **Phạm vi**: [0, 1]. Giá trị càng thấp càng tốt.

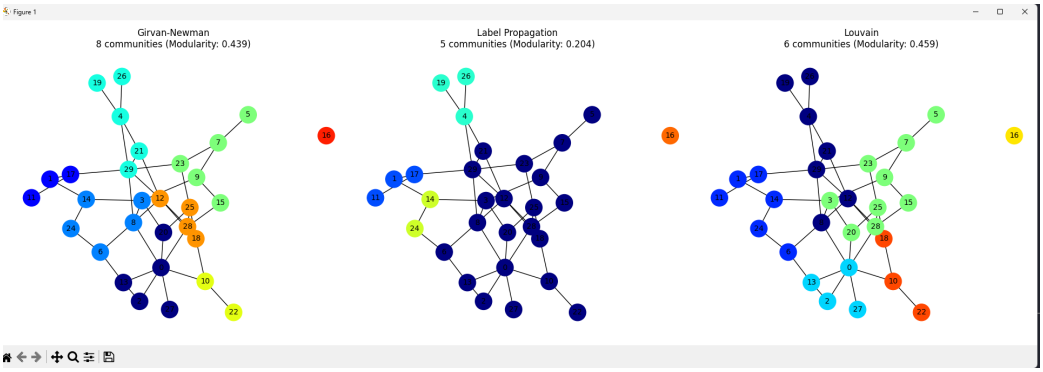
## 3. Normalized Cut

- **Ý nghĩa**: Đo tổng chi phí chia đồ thị thành các cộng đồng.
- **Công thức**:

$$\text{NCut}(S) = \frac{\text{Cạnh cắt giữa } S \text{ và } \bar{S}}{\text{Volume của } S} + \frac{\text{Cạnh cắt giữa } S \text{ và } \bar{S}}{\text{Volume của } \bar{S}}$$

- **Phạm vi**: [0, $\infty$]. Giá trị càng thấp càng tốt.

# Nhận xét phân tích việc phát hiện cộng đồng




Hình 1: Các cộng đồng từ các thuật toán

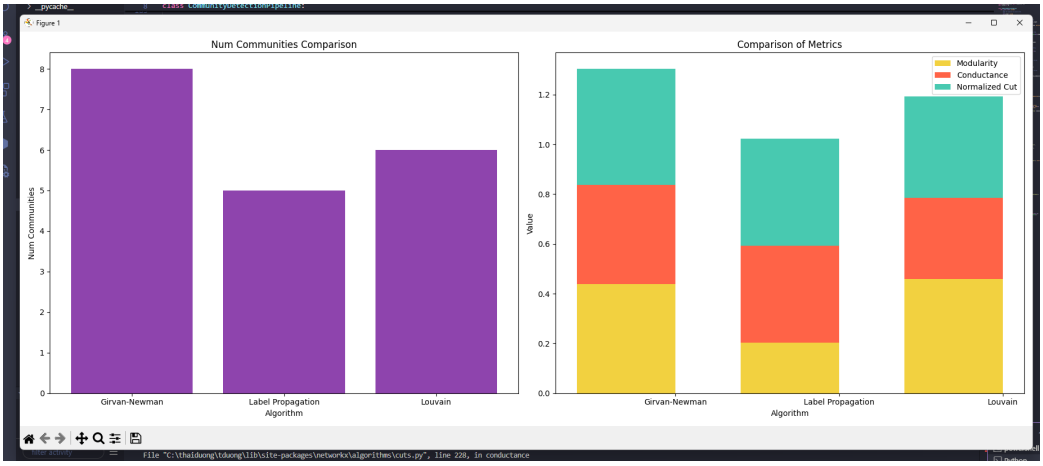
## Kết quả quan sát

1. **Girvan-Newman**:
   - Phân chia thành 8 cộng đồng, Modularity = 0.439.
   - Số lượng cộng đồng lớn, phù hợp cho đồ thị có cấu trúc kết nối lỏng lẻo.
   - Thời gian chạy chậm với đồ thị lớn do tính toán liên tục betweenness centrality.
2. **Label Propagation**:
   - Phân chia thành 5 cộng đồng, Modularity = 0.204.
   - Kết quả cộng đồng không ổn định do nhãn được lan truyền ngẫu nhiên.
   - Tốc độ nhanh, nhưng độ chính xác thấp trên đồ thị phức tạp.
3. **Louvain**:
   - Phân chia thành 6 cộng đồng, Modularity = 0.459.
   - Cộng đồng được tối ưu hóa rõ ràng, modularity cao nhất.
   - Phù hợp nhất với đồ thị có cấu trúc kết nối mạnh trong các cộng đồng.

### Nhận xét đánh giá

- **Modularity**: Louvain cho kết quả tốt nhất, tiếp theo là Girvan-Newman. Label Propagation có modularity thấp, phản ánh sự không ổn định của thuật toán.
- **Conductance**: Louvain thường có giá trị thấp hơn, chứng tỏ các cộng đồng ít giao cắt hơn.
- **Normalized Cut**: Louvain cũng vượt trội, chứng minh khả năng tối ưu hóa tốt trên đồ thị lớn.

## Phân tích mạng xã hội tự chọn




Hình 2: Trực quan so sánh các thuật toán

### Mô tả mạng xã hội

Chọn một đồ thị mô phỏng mạng xã hội với 30 nút và xác suất kết nối $p = 0.1$:

```
G_social = nx.erdos_renyi_graph(n=30, p=0.1, seed=42)
pipeline_social = CommunityDetectionPipeline(G_social)
pipeline_social.run_pipeline()
pipeline_social.visualize_all_communities()
pipeline_social.compare_algorithms()
```

## Kết quả nhận xét

- Với đồ thị này, thuật toán Louvain tiếp tục cho kết quả tốt nhất, với Modularity đạt giá trị cao nhất và số lượng cộng đồng hợp lý.
- Girvan-Newman có thể tạo ra số lượng cộng đồng lớn hơn, phù hợp với đồ thị có ít liên kết giữa các nút.
- Label Propagation cho kết quả kém ổn định, đặc biệt khi cấu trúc đồ thị phức tạp hoặc không đồng nhất.

## Kết luận

Louvain là lựa chọn tối ưu cho đồ thị mạng xã hội lớn và phức tạp, đặc biệt khi cần tính toán nhanh và tối ưu Modularity. Girvan-Newman phù hợp hơn cho phân tích chi tiết với đồ thị nhỏ, trong khi Label Propagation thích hợp cho các ứng dụng cần kết quả nhanh và đơn giản.