Documentació del Projecte de Percolació de Grafs

Equip de desenvolupament

October 23, 2024

Contents

1 Descripció general del projecte

Aquest projecte està dissenyat per simular la percolació en diferents tipus de grafs i analitzar el comportament de les components connexes a mesura que es varia la probabilitat de percolació. El programa principal `graphPerlocator_beta.py` genera diferents tipus de grafs (graella 2D, graella 3D, Random Geometric Graphs, i caveman graphs) i realitza simulacions de percolació sobre ells. A més, desa els resultats en format CSV i genera gràfics dels resultats.

El programa `plotter.py` permet carregar les dades generades i visualitzar-les gràficament, aplicant filtres i agrupant els resultats per a la seva anàlisi.

2 Requisits

Per executar els scripts, és necessari tenir instal·lades les següents biblioteques de Python:

- `networkx`
- `matplotlib`
- `numpy`
- `pandas`
- `multiprocessing`
- `itertools`

Aquestes biblioteques es poden instal·lar executant la següent comanda:

```
pip install -r requirements.txt
```

3 Execució dels scripts

3.1 Executar el simulador de percolació

El simulador de percolació es pot executar en dos modes diferents: **manual** (m) o **automàtic** (a).

3.1.1 Mode manual (m)

En el mode manual, l'usuari pot introduir els paràmetres de manera interactiva per a cada execució del simulador. Això permet personalitzar els valors per a una simulació específica.

Per executar la simulació en mode manual, utilitzeu la següent comanda:

```
python3 graphPerlocator_beta.py
```

A continuació, l'script us demanarà els següents paràmetres:

- `graph_type`: Tipus de graf (g2d, rgg, g3d, ccg)

- `n`: Mida del graf (ex. 20x20 per a g2d, 20x20x20 per a g3d, nombre de coves per a ccg)
- `k`: Nombre d'iteracions per a cada probabilitat
- `q`: Nombre de valors de probabilitat a estudiar (per exemple, si q=100, les probabilitats seran 0.01, 0.02, ..., 1)
- `perl_vertex`: Si és 1, es realitza la percolació per nodes; si és 0, per arestes.

Aquest mode és adequat per generar simulacions amb configuracions personalitzades i ajustades a valors específics.

3.1.2 Mode automàtic (a)

En el mode automàtic, es pot utilitzar un fitxer de configuració `config.ini` per definir diversos paràmetres i realitzar múltiples execucions de manera automàtica. Això és útil per generar un gran nombre de simulacions amb diferents paràmetres sense haver d'introduir-los manualment.

El fitxer `config.ini` segueix una estructura de clau-valor on es defineixen les opcions a generar. Un exemple de `config.ini` seria:

```
[simulation]
graph_type = g2d
n = 10, 20, 30, 40...
k = 20
q = 100
perl_vertex = 1
```

Amb aquest arxiu, l'script generarà automàticament grafs de tipus 2D (`g2d`) amb mides des de `n=10`, `n=20`... Amb 20 iteracions (`k=20`) i estudiant 100 valors de probabilitat (`q=100`).

Per executar la simulació en mode automàtic, utilitzeu la següent comanda:

```
python3 graphPerlocator_beta.py -a
```

L'script llegirà els paràmetres del fitxer `config.ini` i executarà totes les simulacions definides.

Aquest mode és ideal per realitzar experiments massius amb diferents configuracions sense haver d'intervenir manualment.

3.2 Visualitzar els resultats

Després d'executar les simulacions, podeu visualitzar els resultats generats executant:

```
python3 plotter.py
```

L'script us demanarà els següents paràmetres per carregar i filtrar les dades:

- `graph_type`: Tipus de graf (g2d, rgg, g3d, ccg)
- `n_range`: Rang de mida dels grafs (per exemple, de 10 a 40)

- `prob`: Nombre de valors de probabilitat
- `inst_per_prob`: Nombre d'instàncies per probabilitat
- `perl_vertex`: Si és 1, percolació per nodes; si és 0, per arestes

L'script carregarà les dades des de la carpeta `data` i generarà una visualització gràfica dels resultats.

4 Estructura del projecte i generació de resultats

Els resultats de les simulacions s'emmagatzemen a la carpeta `data`, organitzada de la següent manera:

- `g2d/`, `rgg/`, `g3d/`, `ccg/`: Cada carpeta representa un dels tipus de grafs generats.
- Dins de cada carpeta es creen subcarpetes amb un nom que segueix el format:

  ```
  {graph_type}_n{n}_k{k}_q{q}_pvert{pvert}
  ```

  on:

  - `graph_type`: Tipus de graf (g2d, g3d, etc.)
  - `n`: Mida del graf
  - `k`: Nombre d'iteracions per a cada probabilitat
  - `q`: Nombre de valors de probabilitat
  - `pvert`: Si és 1, percolació per nodes; si és 0, per arestes

Dins de cada subcarpeta es generen els següents arxius:

- `YYYY-MM-DD_HH:MM.csv`: Un arxiu CSV que conté els valors de probabilitat i la mitjana de components connexes.
- `YYYY-MM-DD_HH:MM.png`: Un gràfic que representa la mitjana de components connexes en funció de la probabilitat.

5 Descripció del codi

5.1 graphPerlocator_beta.py

Aquest és el programa principal que genera els grafs i realitza la simulació de percolació. Es divideix en les següents seccions principals:

- **Generació de grafs**: Hi ha funcions per generar diferents tipus de grafs:
  - `generate_connected_grid(n)`: Genera una graella 2D.
  - `generate_connected_3dgrid(n)`: Genera una graella 3D.

  - `generate_connected_rgg(n, radius)`: Genera un Random Geometric Graph (RGG).
  - `generate_connected_caveman_graph(l, k)`: Genera un graf caveman.

- **Simulació de percolació**: La funció `percolate_graph(G, p, perl_vertex=True)` aplica la percolació al graf, eliminant nodes o arestes amb una probabilitat p.

- **Avaluació i mitjana de components connexes**: La funció `evaluate(G, k, q, perl_vertex=True)` realitza múltiples simulacions i calcula la mitjana de components connexes per a diferents probabilitats.

- **Visualització i desament**: La funció `plot_and_save` genera gràfics i desa els resultats en arxius CSV i PNG.

5.2 plotter.py

Aquest programa carrega els resultats generats per `graphPerlocator_beta.py` i permet visualitzar-los amb diferents filtres. Es estructura de la següent forma:

- **Càrrega de dades**: La funció `load_data` carrega els arxius CSV generats, permetent aplicar filtres per tipus de graf, mida, iteracions, etc.

- **Visualització**: La funció `plot_data` genera gràfics dels resultats carregats, mostrant l'evolució de les components connexes en funció de la probabilitat.