10. Networkモデル(その1)

2016/1/21

1　はじめに

今回は、ネットワーク (グラフ) をプログラムと実装するとともに、ランダムネットワークを構成しましょう。今回は、無向ネットワーク、つまり辺に向きの無いものを考えましょう。

その準備として新たにプロジェクト Network を作成します。以下の URL から Java ソースファイルを取得し、解凍した後、プロジェクト Network のソースファイルとして置きます。

http://http://aoba.cc.saga-u.ac.jp/lecture
/ModelingAndSimulation/javasrc/Network/src1.zip

また、ライブラリ MyLib も設定しましょう。

2　Node クラスと Edge クラス

ネットワーク (グラフ) は、頂点 (Node) を向きの無い辺 (Edge) で結んだものです。構成後のネットワークの探索などを行えるためには、

- 頂点に接続している辺
- 辺の両端の頂点

が分かる必要があります。

表 1: Node クラスのコンストラクタ

コンストラクタと説明
public Node(String label) ラベル label を与えて初期化する。

表 2: Node クラスのメソッド

修飾子と型	メソッドと説明
void	addEdge(Edge edge) 頂点に辺 edge を接続する。
List<Edge>	getEdges() 頂点に接続している辺の一覧を返す。
String	getLabel() 頂点のラベルを返す。
boolean	isNeighbour(Node node) 指定した頂点 node が隣接頂点ならば true を返す。
List<Node>	neighbours() 隣接頂点の一覧を返す。
void	removeEdge(Edge edge) 頂点から辺 edge を切り離す
String	toString() 頂点のラベルを返す。

表 3: Edge クラスのコンストラクタ

コンストラクタと説明
public Edge(String label) ラベル label を与えて初期化する。

表 4: Edge クラスのメソッド

修飾子と型	メソッドと説明
void	addEnd(Node node) 辺に node を付ける。
List<Node>	getEnds() 辺の端点の頂点の一覧を得る。
Node	getEnd(Node node) 辺の node と反対側の頂点を得る。
String	getLabel() 辺のラベルを得る。
boolean	hasEnd(Node node) node が辺の端点であれば true を返す。
void	setLabel(String label) 辺のラベルを label に変更する。

3 BaseNetwork クラス

ネットワークを実際に構成するための基本的なクラスとして BaseNetwork を定義します。このクラスは具体的なネットワーク構造を有しない抽象クラスとして定義しておきます。

表 5: BaseNetwork クラスのコンストラクタ

コンストラクタと説明
public BaseNetwork(String label) ラベル label を与えて初期化する。

表 6: BaseNetwork クラスのメソッド

修飾子と型	メソッドと説明
boolean	addNode(Node node)— ネットワークに頂点 node を追加する。
Edge	connectNodes(Node n1, Node n2) 二つの頂点 n1 と n2 を結ぶ新たな辺を生成し、その辺を返す。ラベルは自動的に生成する。
Edge	connectNodes(Node n1, Node n2, String label) 二つの頂点 n1 と n2 を結ぶ新たな辺をラベル label を指定して生成し、その辺を返す。
abstract	boolean createNetwork() 実際にネットワークを構成する抽象メソッド。
String	generatePajekData() ネットワークに対応した pajek 用のデータを文字列として生成する。
List<Edge>	getEdges(Node node) 指定した頂点 node に接続している辺の一覧を返す。
String	getLabel() ネットワークのラベルを返す。
List<Node>	getNodes() ネットワークに含まれる頂点の一覧を返す。
int	getNumNode() ネットワークに含まれる頂点の数を返す。
void	outputPajekData(String filename) pajek 用のデータを生成し、filename で指定されたファイルに出力する。

クラス BaseNetwork の main には、簡単な例として 3 頂点の完全ネットワーク (図 1) を記載しています。通常、抽象クラスのインスタンスを生成することはできません。そこで、Program 3.1 では、抽象クラス BaseNetwork のインスタンス network の生成時に、抽象メソッド createNetwork を実装しています。この中では、最初に頂点を 3 個生成した後、connectNodes メソッドを用いて二つの頂点を結び付けています。最後に、pajek データをファイルに出力しています。

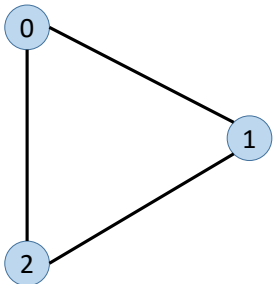

図 1: 3 頂点の完全ネットワーク

課題 1　クラス `BaseNetwork` を実行し、出力された `.net` ファイルを pajek を使って作図しなさい。

pajek を使ったサイズ方法を説明します。pajek を起動します。「network」という部分にあるフォルダのアイコンをクリックすることで、ファイルを選択します。次に、メニューから「Draw→Network」を選ぶことで作図することができます。

課題 2　クラス `BaseNetwork` の `main` を書き換えて、図 2 を生成しなさい。なお、pajek で頂点の配置を自動的に見やすい形にするには、ネットワーク表示画面において、「Layout→Energy→Kamada-Kawai→Free」とすることで可能です。また、頂点をマウスでドラッグして移動することも可能です。

また、ネットワーク表示画面から「Export」メニューを選ぶことで、図をファイルとして出力することができます。

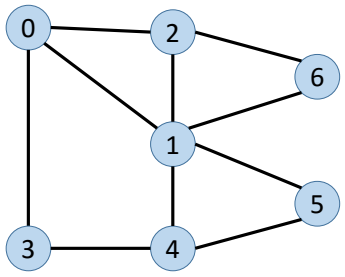

図 2: サンプルネットワーク

4　Erdös-Rényi のランダムネットワーク

講義で扱った Erdös-Rényi のランダムネットワークを実際に構成しましょう。頂点数を N、辺の総数 L とする。頂点数 N の完全グラフにおける辺の数との比を p

```
static public void main(String args[]) throws IOException {
    //抽象クラスのインスタンスを生成
    BaseNetwork network = new BaseNetwork("testNetwork") {
        //実装の無いメソッドを具体的に記述
        @Override
        public boolean createNetwork() {
            int n = 3;
            for (int i = 0; i < n; i++) {
                addNode(new Node(String.valueOf(i)));
            }
            connectNodes(nodes.get(0), nodes.get(1));
            connectNodes(nodes.get(1), nodes.get(2));
            connectNodes(nodes.get(2), nodes.get(0));
            return true;
        }
    };
    network.createNetwork();
    //pajek用データを出力
    network.outputPajekData(network.getLabel() + ".net");
}
```

Program 3.1: BaseNetwork クラスの main メソッドに記載された、3 頂点の完全ネットワークの例

とする。

$$L = p\frac{N\left(N-1\right)}{2} \tag{4.1}$$

ランダムネットワークは、でたらめに頂点の組を L 個選び辺を生成することで構成することができます。

ランダムネットワークのクラス ER を抽象クラス BaseNetwork の拡張として定義します。配布した networkModels の下に含まれています。

表 7: ER クラスのコンストラクタ

コンストラクタと説明
public ER(int n, int numEdges) 頂点数 n と辺の数 numEdges を与えて初期化する。

構成するには、メソッド createNetwork を用います。このメソッドは、さらに二つの private メソッドを呼びます。メソッド createNodes は、n 個の頂点を生成します。頂点はリスト nodes に保存されています。メソッド createEdges は、リスト nodes からでたらめに二つの頂点を選び辺を生成することを numEdges 回繰り返します。

課題 3　メソッド createEdges を実際に記述しなさい。

課題 4　メソッド main では、実際に頂点数 n と辺の数 numEdges を指定してネットワークを構成し、pajek 用のファイルを出力しています。n を 100 として、numEdges を 10、100、200、500 と変化させ、構成されるランダムネットワークを pajek を用いて表示し、様子を観察しなさい。