

クラスタ構造解析アプリケーション (ver0.9)

ファイル構成

- README.md : セットアップ説明ドキュメント
- config.py : 設定ファイル
- enot_main.py : 実行メインスクリプト
- data/ : 入力データ用ディレクトリ

クラスタリング実行方法

config.py で設定した後に以下のコマンドで実行して下さい。

```
python enot_main.py
```

出力結果はconfig.pyで指定されたファイルとして出力されます。

config.py

- infile_path : 入力データCSV のパス
- infile_directed_type : 入力データの指向性タイプ
 - 1 : 指向性
 - 2 : 無指向性
- vertices_file_path : ノード名を定義したファイル
- total_nodes : 全ノード数
- outfile : 出力ファイル名
- p_algo_type : Paの算出方法
 - 1 : Power method
 - 2 : Arnoldi method
- p_conv_threshold : Power methodに於けるPaの収束しきい値 (Rosvall(2010)では 1.0e−15)
- teleport_type : Teleportationのタイプ
 - 1 : standard teleportation
 - 2 : smart recorded teleportation
 - 3 : smart unrecorded teleportation
- tau : τ の値
- quality_method : 最適化の方法（ここは任意に追加、変更できます。詳しくは“新規手法の導入方法”を御覧ください）
 - 1 : Map equation
 - 2 : Modularity
- division_type : 解析タイプ
 - 1 : two-level
 - 2 : 階層化
- num_trial : 各階層モジュール毎の分割リトライ回数（論文的には100回ですがnode数に合わせて調整した方が現実的と思われます）

python における設定値（特に調整する必要は無いです）

- threshold_search : 高精度の計算で0に対応するための設定値
- myfloat : float の精度
- seed_var : random node pick のための乱数シード値

入力データフォーマット

データフォーマットは遷移の重さで表し、 [source node id, target node id, weight] で表現されています。(Link list format)

実行結果

出力ファイル

tree map format の csv で出力されます。
各行は
[階層構造, 滞在確率, ノード（モジュール）名, ノード（モジュール）index]
となっています。

参考

```
# Codelength = 3.48419 bits.
1:1:1 0.0384615 "7" 6
1:1:2 0.0384615 "8" 7
1:1:3 0.0384615 "9" 8
1:2:1 0.0384615 "4" 3
1:2:2 0.0384615 "5" 4
```

可視化HTML

最適化の手法 MapEquation と Modularity の解析結果をモジュールごとに色分けしてネットワークをブラウザで確認することが出来ます。各手法で解析を実行した後に vis_html/vis.html を safari か firefox で開いて下さい。（出力結果を読み込む関係で、chromeやIEではセキュリティの制御により表示することが出来ません。）

新規手法の導入方法

1.quarity.py内の__new__関数内に新たな分岐をつくる

```
def __new__(cls):
    if cf.quality_method == 1: # use map equation for communities' quality estimation
        import mapequation as mp
        new_cls = mp.Map
    elif cf.quality_method == 2: # use modularity for communities' quality estimation
        import modularity as ml
        new_cls = ml.Modularity
    elif cf.quality_method == 3: #新規評価方法に対して、新しい番号をふる。config.py内 quality_methodの値に対応。
        import someNewMethod as sn
        new_cls = sn.someNewMethod
    else:
        print("error: in config.py, undefined number of quality_method was selected.")
        sys.exit(1)
```

2.評価式を実装するモジュール(.py)をlibフォルダ内に作成

実装に求められること:

bool check_network_got_better(self, ql_before, ql_after)
bool check_network_converged(self, ql_before, ql_after)
float get_quality_value(self, __modules, w, p_a)
の3つの関数が必要です。
実際に実装する場合は、util/someNewMethod.pyにサンプルファイルを用意したのでこれを修正、libフォルダに移動して使用します。

変更点

ver 0.9

- MapEquation と Modularity によるクラスタ構造解析処理実装
- クラスタリング結果可視化処理実装