3B3-OS-10a-3

制約付きコミュニティ抽出の高速化と対話的環境の構築

Speeding-up of Constrained Community Detection and Development of its Interactive Environment

仲田 圭佑*1　村田剛志*2
Keisuke Nakata　Tsuyoshi Murata

*1東京工業大学大学院 情報理工学研究科 計算工学専攻
Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology

*2東京工業大学大学院 情報理工学研究科 計算工学専攻
Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology

Recently, many methods for analyzing networks have been proposed. Among them, the methods called community detection based on graph theory have advantages that they can make networks simple and easy to understand. However most of them had not considered the background knowledge of data, thus some methods called constrained community detection which take such background knowledge into consideration have been proposed. In this paper, we propose and discuss the speeding-up and interactive environment for constrained community detection. The proposed method improves the computational efficiency of constrained community detection with its accuracy kept comparable. Our proposed method is a variant of the fast-unfolding method which is known for its computational efficiency. By using the proposed method, we evaluate the performance of the ordering of the stepwise constraint adding in constrained community detection.

1. はじめに

近年，爆発的に普及したソーシャルメディアや WWW のハイパーリンク関係など，データ量が増大しているネットワークを構造的に理解しようとする欲求が高まっている．コミュニティ抽出は，ネットワークを何らかの指標でグループ分けすることでそれを構造的に理解することを目的としている．

コミュニティ抽出の指標として，Newman と Girvan によって提案されたモジュラリティ [Newman 04] がよく使われており，その最適化をおこなう手法が数多く提案されてきた [Clauset 04]．さらに，巨大なネットワークを分析するため，精度を高い水準で維持したままコミュニティ抽出にかかる時間を飛躍的に短縮した Fast-unfolding 法が提案された [Blondel 08]．

一方で，既存のコミュニティ抽出手法の多くはネットワークの背後に存在する知識を活用しないものであったため，背景知識を積極的に活用しようと試みる制約付きコミュニティ抽出手法も提案されている [Eaton 12]．この手法では，モジュラリティを一般化 [Reichardt 06] し，さらに制約項を付加した制約付きハミルトニアンを最適化することでコミュニティ抽出をおこない，背景知識を利用しない手法よりもノイズに強く精度の高い結果を示している．しかし，この手法では最適化の手法として擬似焼きなまし法 [Kirkpatrick 83] を用いており，速度面で課題を残している．

そこで本稿では，擬似焼きなまし法の速度面での課題を解決するため，Fast-unfolding 法を制約付きハミルトニアンの最適化に適用できるよう改良することで高速化する手法を提案する．また，提案手法を用いて，ユーザがコミュニティ抽出結果を対話的に修正する環境の評価をおこなうための予備実験をおこなう．

連絡先: 仲田圭佑，東京工業大学，〒 152-8552 東京都目黒区大岡山 2-12-1 W8-59 東京工業大学 大学院情報理工学研究科 計算工学専攻，nakata.k.ad@m.titech.ac.jp

2. 既存法：関連研究

本節では，提案手法の説明にあたり必要となる既存の指標・手法についての説明と議論をおこなう．

2.1 モジュラリティ

ネットワークのコミュニティ抽出結果を測る指標として，Newman と Girvan によって提案されたモジュラリティ [Newman 04] がよく使われている．モジュラリティの値 Q は以下の式で表される：

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta \left(C_i, C_j \right) \tag{1}$$

ここで，m はグラフのエッジの数，i，j はノードのインデックス，A はグラフの隣接行列，k_i はノード i の次数，C_i はノード i が含まれるコミュニティのインデックス，δ はクロネッカーのデルタである．

コミュニティ抽出の文脈では，この Q 値が大きい分割を探索する手法 (モジュラリティ最適化) がしばしば用いられる．

2.2 Fast-unfolding 法

Fast-unfolding 法 [Blondel 08] は，モジュラリティを非常に高速に精度よく最適化することができる手法である．Fast-unfolding 法は大きくふたつのフェイズに分かれている：

1. 各ノードに関して，その隣接コミュニティのいずれに移動させればもっともモジュラリティが高くなるかを計算し，もしそのモジュラリティの最高値が現在のモジュラリティよりも改善するならば，最高値をとるコミュニティをそのノードに割り当てる．これをすべてのノードに対して，コミュニティが変化しなくなるまで繰り返す．

2. 前のフェイズで得られた各コミュニティをそれぞれひとつのノードとみなした新たなグラフを生成する．

この 2 つのフェイズをまとめてパスと呼び，パスを収束するまで繰り返す．第 1 フェイズでは，モジュラリティ全体を再計

算するのではなく，モジュラリティの変化分 ΔQ を計算することで高速化している．ノード x をコミュニティ Y から Z へ移動させたときの ΔQ は，以下の式で表される：

$$\Delta Q = \frac{1}{m}\left(\sum_{i\in Z}\left(A_{ix} - \frac{k_i k_x}{2m}\right) - \sum_{i\in Y}\left(A_{ix} - \frac{k_i k_x}{2m}\right)\right) \quad (2)$$

ここで，k_i はノード i に接続しているエッジの重みの和である．第 2 フェイズでは，第 1 フェイズで得られた各コミュニティをそれぞれひとつのノードとみなした新たなグラフを生成する．このとき，新たなグラフのノード間エッジの重みは，元のグラフのコミュニティ間のエッジの重みの和となり，新たなグラフのセルフループエッジの重みは，元のグラフのコミュニティ内のエッジの重みの和の 2 倍となる．

前回のパスで得られた新たなグラフに対して次のパスを適用し，変化がなくなった時点でのコミュニティ抽出結果を出力する．

2.3 モジュラリティの一般化

モジュラリティ(式 (1)) を一般化したハミルトニアン $\mathcal{H}$ [Reichardt 06] は次の式で表される：

$$\begin{aligned}\mathcal{H} = &-\sum_{i,j} a_{ij}A_{ij}\delta(C_i,C_j)\\ &+\sum_{i,j} b_{ij}(1-A_{ij})\delta(C_i,C_j)\\ &+\sum_{i,j} c_{ij}A_{ij}(1-\delta(C_i,C_j))\\ &-\sum_{i,j} d_{ij}(1-A_{ij})(1-\delta(C_i,C_j))\end{aligned} \quad (3)$$

ここで，モジュラリティと違い，ハミルトニアンが小さい値を取るときに良いコミュニティ抽出となることに注意する必要がある．ハミルトニアンは，a) コミュニティ内にエッジが存在するときに報酬，b) コミュニティ内にエッジが存在しないときに罰則，c) コミュニティ間にエッジが存在するときに罰則，d) コミュニティ間にエッジが存在しないときに報酬を与え，それぞれをパラメータ a，b，c，d によって重み付けしている．

ここで，$a_{ij} = c_{ij} = 1 - \gamma P_{ij}$，$b_{ij} = d_{ij} = \gamma P_{ij}$ とおけば，式 (3) は

$$\mathcal{H} = -\sum_{i,j}(A_{ij} - \gamma P_{ij})\delta(C_i,C_j) \quad (4)$$

と書きなおせる．このとき $\gamma P_{ij} = k_i k_j/2m$ とし，スケールを調整すれば，モジュラリティの定義 (式 (1)) と等価となり，ハミルトニアンがモジュラリティの一般化であることがわかる．

2.4 制約付きコミュニティ抽出

制約付きコミュニティ抽出をおこなう方法のひとつに，前述のハミルトニアン (式 (4)) に制約項を付加した制約付きハミルトニアンを最適化する手法が提案されている [Eaton 12]．制約項 U は，ノードのペアごとに，同じコミュニティに属しているときの制約 u_{ij}(must-link に対応する) と，異なるコミュニティに属しているときの制約 $\overline{u}_{ij}$(cannot-link に対応する) で決定される：

$$U = \sum_{i,j}\left(u_{ij}\left(1-\delta\left(C_i,C_j\right)\right) + \overline{u}_{ij}\delta\left(C_i,C_j\right)\right) \quad (5)$$

すると，制約付きハミルトニアン $\mathcal{H}'$ は次の式で表せる：

$$\begin{aligned}\mathcal{H}' &= \mathcal{H} + \mu U\\ &= -\sum_{i,j}\left(\left(M_{ij} - \mu\Delta U_{ij}\right)\delta\left(C_i,C_j\right)\right) + K\end{aligned} \quad (6)$$

ここで，μ はハミルトニアンと制約項との重みを調整するパラメータ，$M_{ij} = A_{ij} - \gamma P_{ij}$，$\Delta U_{ij} = u_{ij} - \overline{u}_{ij}$，$K = \mu\sum_{i,j} u_{ij}$ である．K はコミュニティ抽出の結果に依らない定数であることに注意する．

Eaton らは擬似焼きなまし法 [Kirkpatrick 83] によって式 (6) の最適化をおこない，ノイズに強く精度の高い結果が得られることを示した [Eaton 12]．

3. 提案法:制約付きコミュニティ抽出の高速化

Eaton らによって，制約付きハミルトニアンの最適化をおこなうことで良い精度の結果が得られることが示されたが，その最適化には擬似焼きなまし法が使われており，速度面で課題を残している．そこで，本稿では，既に非常に高速に精度よくモジュラリティを最適化できることが知られている Fast-unfolding 法を制約付きハミルトニアンの最適化へと拡張することで，制約付きコミュニティ抽出の高速化を実現する．提案法の流れは Fast-unfolding 法と全く同一であるが，第 1 フェイズにおいてノードのコミュニティを移動させる際，モジュラリティの変化分 ΔQ ではなく，制約付きハミルトニアンの変化分 $\Delta\mathcal{H}'$ を用いる：

$$\Delta\mathcal{H}' = 2\left(-\sum_{i\in Z}\left(M_{ix} + \mu\Delta U_{ix}\right) + \sum_{i\in Y}\left(M_{ix} + \mu\Delta U_{ix}\right)\right) \quad (7)$$

4. 実験

実験で用いたネットワークを表 1 に示す．いずれも正解ラベルのついている実ネットワークである．実験で使用したパラメータは，$\mu = 1$，$\gamma = 1$，$P_{ij} = k_i k_j/2m$ である．

制約は各ノードにユーザ定義のラベル l_i (i はノードのインデックス) を付与する形式とした．このとき，ラベルが付与されていないノードには $l_i = -1$ のラベルを形式的に付与することで区別する．u_{ij} および $\overline{u}_{ij}$ は次のように定めた：

$$u_{ij} = \begin{cases}1 & (\text{when } l_i = l_j \neq -1),\\ 0 & (\text{otherwise}),\end{cases} \quad (8)$$

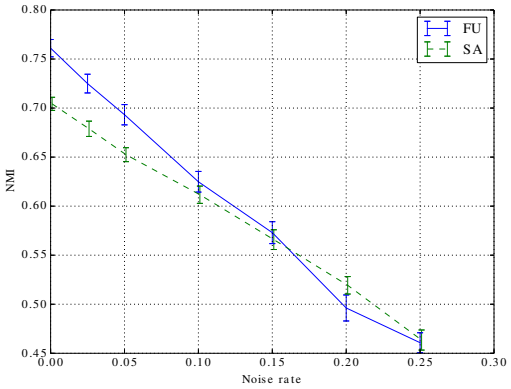
$$\overline{u}_{ij} = \begin{cases}1 & (\text{when } l_i \neq l_j \neq -1),\\ 0 & (\text{otherwise}).\end{cases} \quad (9)$$

二種類のコミュニティ抽出結果 C と C' がどれだけ似ているかを測る指標として，normalized mutual information (NMI) [Strehl 03] を用いた：

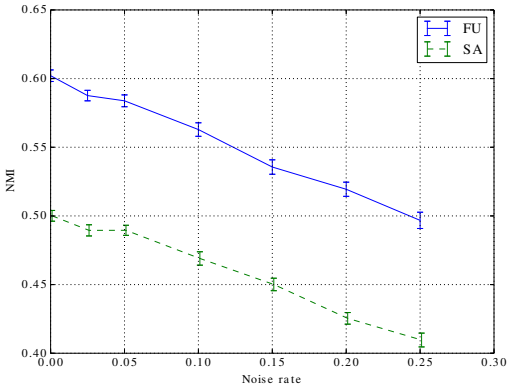
$$\mathrm{NMI}(C,C') = \frac{\sum_c\sum_{c'} n_{cc'}\log\frac{n_{cc'}\cdot n}{n_c\cdot n_{c'}}}{\sqrt{\left(\sum_c n_c\log\frac{n_c}{n}\right)\left(\sum_{c'} n_{c'}\log\frac{n_{c'}}{n}\right)}} \quad (10)$$

表 1: 実験で用いたネットワーク

Network	#nodes	#edges	#communities
Karate [Zachary 77]	34	78	2
Polbooks [Krebs]	105	441	3

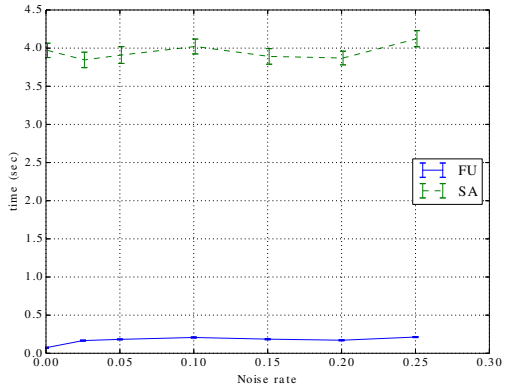


(a) Karate

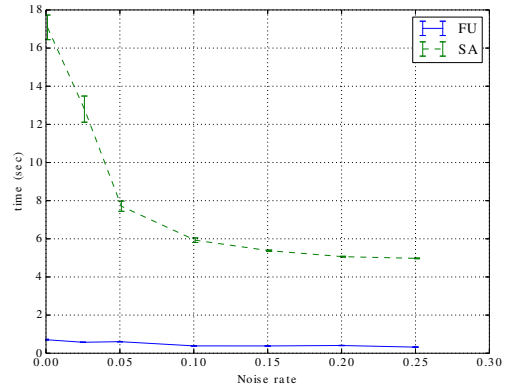


(b) Polbooks

図 1: 擬似焼きなまし法と提案法によって得られた NMI の比較．緑の破線が擬似焼きなまし法 (SA)，青の実線が提案法 (FU)．縦軸は NMI，横軸は Noise rate (エッジをランダムに消去・追加する割合)．全ノードの 20%に制約を付与した．エラーバーは標準誤差である．



(a) Karate



(b) Polbooks

図 2: 擬似焼きなまし法と提案法による計算時間の比較．設定は図 1 と同様．ただし縦軸は計算時間 (秒) である．

ここで，c，c' はコミュニティ抽出結果 C，C' の各コミュニティのインデックス，n はノードの総数，$n_{cc'}$ は c と c' の両方に属するノードの数，n_c，$n_{c'}$ はそれぞれ c，c' に属するノードの数である．C と C' が似たコミュニティ抽出結果であるほど，NMI は大きくなる．

C を抽出したコミュニティ，C' を正解ラベルとすることで，コミュニティ抽出結果の精度を測る．

4.1 擬似焼きなまし法と提案法の比較

本小節では，最初にまとめて制約を付与した場合の制約付きハミルトニアンの最適化について議論する．

図 1 では，擬似焼きなまし法と提案法において NMI を比較した．提案法は，Karate ネットワークでは擬似焼きなまし法と同等，Polbooks ネットワークではそれより良い精度でコミュニティ抽出をおこなえたことがわかる．ノイズに強い特徴を持っていた擬似焼きなまし法と比べても，提案法は同等のロバスト性を示しており，精度の面で提案法は擬似焼きなまし法と同等あるいはそれより良いことが示された．Fast-unfolding 法が他の最適化手法よりも精度の良いコミュニティ抽出をおこなえることは既に示唆されており [Blondel 08]，それと整合性のとれた結果である．

図 2 は速度の比較である．提案法は，擬似焼きなまし法よりも高速化したことがわかる．Fast-unfolding 法はノード数がおよそ 100 万のネットワークにおいても約 3 秒で結果を返す [Blondel 08]．今回は巨大なネットワークでの実験はおこなわなかったが，提案法は Fast-unfolding 法の変形であるため，同等のパフォーマンスが期待できる．

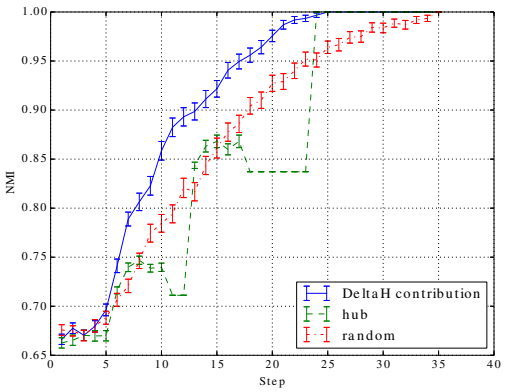
以上から，提案法は擬似焼きなまし法と同等あるいはそれより良い精度で，非常に高速に制約付きハミルトニアンを最適化できることがわかった．

5. 対話的な制約付与の予備実験

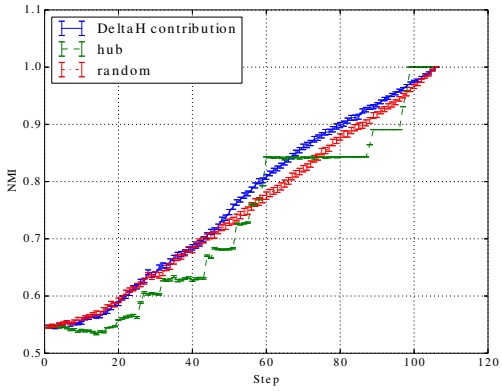
本小節では，制約をひとつずつ順次付与していった場合の制約付きハミルトニアンの最適化について議論する．つまり，制約なしの状態から開始し，なんらかの基準を元に決めた順番でひとつずつノードを選び制約を付与する．

図 3 では，提案法においてノードに制約をひとつずつ付加していった際の NMI を比較した．制約付きハミルトニアンの改善が小さかったノード順に制約を追加する方法は，ランダムに追加するよりもわずかに精度の上がり方が良いことがわかる．一方，次数の大きなノード順に制約を追加する方法は，ランダムよりもむしろ精度の上がり方が悪くなっている．

以上から，ランダムに制約を追加するよりも，精度の上がり方が良い順番が存在することが確認できた．制約追加ノードの順番を決める戦略では，「真の値との整合が取れていないノード組のうち，その間違いを定量化した値がもっとも大きな組から選択していく (不確実性サンプリング [Lewis 94] の類似戦



(a) Karate



(b) Polbooks

図 3: 制約をひとつずつ付加していった時の NMI の変化．青の実線が最初のパスの第 1 フェイズにおいて制約付きハミルトニアンの改善が小さかったノード順に制約を追加した場合 (DeltaH contribution)，緑の破線が次数の大きなノード順に制約を追加した場合 (hub)，赤の鎖線がランダムに制約を追加した場合 (random)．縦軸が NMI，横軸が制約の数．エラーバーは標準誤差である．

略)」(以下，これを欲張り戦略と呼ぶ) ことを目標としているが，制約追加の順番を決める際，人間による戦略は，欲張り戦略よりも優れていることが示唆されている [山田 14]．よって，実際にユーザが対話的に制約追加をおこなう場合，精度の上がり方は図 3 よりも良くなることが示唆される．

6. おわりに

本稿では Fast-unfolding 法を制約項付きハミルトニアンの最適化に適用できるよう改良を加え，精度と速度の両方を高い水準で得ることができることを確認した．また，提案手法を用いて，ユーザがコミュニティ抽出結果を対話的に修正する環境を構築し，制約追加ノードの順番による効果についても考察をおこなった．

残された課題として，第一に，制約追加時に前ステップの Fast-unfolding 法で得られた階層構造のコミュニティ抽出結果を再利用することによる高速化が挙げられる．対話的環境では順次制約を付与するため，前ステップの結果を再利用しやすく，計算時間の短縮が期待される．第二に，欲張り戦略をさらに考察し，効果を確認する必要がある．制約の順番を決める戦略では，人間による戦略が欲張り戦略よりも優れていることが示唆されている．しかし，人間の選択を支援するための補助として，次のステップでの制約の提案を欲張り戦略によっておこなうことは対話的環境の向上に貢献する可能性がある．人間のとる戦略を模倣するヒューリスティックな戦略についても考察の余地がある．

参考文献

[Blondel 08] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E.: Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 2008, No. 10, p. P10008 (2008)

[Clauset 04] Clauset, A., Newman, M. E. J., and Moore, C.: Finding community structure in very large networks, *Phys. Rev. E*, Vol. 70, p. 066111 (2004)

[Eaton 12] Eaton, E. and Mansbach, R.: A spin-glass model for semi-supervised community detection, in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*, pp. 900–906, AAAI Press (2012)

[Kirkpatrick 83] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P.: Optimization by simulated annealing, *Science*, Vol. 220, pp. 671–680 (1983)

[Krebs] Krebs, V.: Books about US politics, Nodes represent books about US politics sold by the online bookseller Amazon.com. Edges represent frequent co-purchasing of books by the same buyers, as indicated by the "customers who bought this book also bought these other books" feature on Amazon.

[Lewis 94] Lewis, D. D. and Gale, W. A.: A Sequential Algorithm for Training Text Classifiers, in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pp. 3–12, New York, NY, USA (1994), Springer-Verlag New York, Inc.

[Newman 04] Newman, M. E. J. and Girvan, M.: Finding and evaluating community structure in networks, *Phys. Rev. E*, Vol. 69, p. 026113 (2004)

[Reichardt 06] Reichardt, J. and Bornholdt, S.: Statistical mechanics of community detection, *Phys. Rev. E*, Vol. 74, p. 016110 (2006)

[Strehl 03] Strehl, A. and Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions, *The Journal of Machine Learning Research*, Vol. 3, pp. 583–617 (2003)

[Zachary 77] Zachary, W.: An information flow model for conflict and fission in small groups, *Journal of Anthropological Research*, Vol. 33, pp. 452–473 (1977)

[山田 14] 山田 誠二, 水上 淳貴, 岡部 正幸：インタラクティブ制約付きクラスタリングにおける制約選択を支援するインタラクションデザイン, 人工知能学会論文誌, Vol. 29, No. 2, pp. 259–267 (2014)