後戻りなし Metropolis-Hastings Random Walk による ソーシャルネットワークのグラフサンプリング

岩崎　謙汰†　　首藤　一幸†

† 東京工業大学 情報理工学院 数理・計算科学系
E-mail: †iwasaki.k.ah@m.titech.ac.jp

あらまし　ランダムウォークによるグラフサンプリング手法は，実際のソーシャルネットワークの情報取得の制限下でも複雑ネットワークの様々な性質を推定するため有用である．特に Metropolis-Hastings Random Walk は定常分布が一様分布であるため，サンプルした結果をそのまま推定に使用できるという利点を持つ．しかし，次数が低いノードに留まりやすいため，広範囲にサンプルするのに多くのステップ数が必要となり，推定の精度を下げる原因となっている．本研究では，これらの問題点を解決した後戻りなし Metropolis-Hastings Random Walk を提案する．提案手法は，少ないステップ数で広範囲にサンプリングを行うことが可能であり，既存手法よりもネットワークの特徴の推定の精度が高い．本論文ではこのことを実験により示した．

キーワード　ソーシャルネットワーク，グラフサンプリング，ランダムウォーク

1. はじめに

昨今，Online Social Networks (OSNs) や Peer-to-peer (P2P) ネットワーク，world wide web (WWW) などの複雑ネットワークのノードおよびトポロジの特徴量を推定する研究が注目されている．ネットワークの特徴量の具体例としては，次数分布やクラスタ係数などが挙げられる．しかし，これらのネットワークの全体情報は一般的に公開されておらず，ネットワークの特徴量を推定することは簡単ではない．例えば，ノードを一様独立に選択し直接取得する一様独立サンプリングのような手法を，このようなノード ID が未知のネットワークに適用することは実現不可能である．現実的には，ソーシャルネットワークの API やスクライピングなどを繰り返し使用することでサンプリングを行うため，クローリングによるグラフサンプリング手法を考える必要がある．この時，ノードのサンプル集合に偏りが生じてはいけないため，マルコフ連鎖によって不偏サンプリングを達成できるランダムウォークベースの手法が有用だとされている [1].

ランダムウォークによる不偏サンプリング手法には 2 つの代表的な手法が存在する．通常のランダムウォークによってノードをサンプルした後，事後処理を行うことで特徴量を推定する Random Walk Re-Weighting (RWRW) と Metropolis-Hastings (MH) アルゴリズムによって一様にノードをサンプルすることで不偏サンプリングを達成する Metropolis-Hastings Random Walk (MHRW) である．この 2 手法の関係はトレードオフの関係であることが [1] で考察されている．トレードオフの関係を簡単にまとめたのが表 1 である．本論文では MHRW に着目し，MHRW の長所である ready-to-use 性を残しつつ，短所であるサンプル範囲の狭さを改善した手法を提案する．

MHRW の長所は，定常分布が一様分布であるためサンプルしたノードを計算時にそのまま使用することができる点である．

表 1　RWRW と MHRW の比較

	RWRW	MHRW
サンプリング後の処理	必要	ready-to-use
サンプル範囲	大きい	小さい

この性質を ready-to-use 性と呼ぶこととする．ready-to-use 性は，データの使い勝手がいいだけでなく，他人にサンプルノードデータを配布しやすいといった利点を持つ．これに対し通常のランダムウォークの場合，次数が高いノードに偏ったサンプリングが行われるため，サンプル後の特徴量推定時にノードの次数の偏りを排除するような重み付け計算を行わなければ正しい推定を行うことができない．この場合，多次元スケーリングや階層的クラスタリングなど，複雑なデータ分析の場面では適用困難といった欠点を持つ [1].

MHRW の短所は，通常のランダムウォークに比べて，広範囲のサンプリングを行うために非常に多くのステップ数が必要となる点である．MHRW は高次数ノードへの訪問の偏りを無くすために受理確率に応じて訪問拒否を行っており，その結果次数が低いノードに対してサンプルが重複する確率が高く，広範囲へのサンプリングが遅くなってしまう．ネットワークデータの単位時間あたりの取得量に制限があることを考えると，少ないステップ数かつ高い精度でネットワークの特徴量を推定できる方が望ましい．広範囲にノードサンプリングすることで推定値の分散が小さくなり，高精度の推定を達成できる．本研究の提案手法は，既存の MHRW より広範囲にサンプリングできるようにすることで推定精度の向上を試みる．

本論文では，一つ前のサンプルノードには遷移しないというルールを MHRW に追加することで，MHRW の長所である ready-to-use 性を保ったまま，MHRW よりも広範囲にサンプリングすることで推定の精度を向上させる手法を提案する．これを後戻りなし Metropolis-Hastings Random Walk (NBMHRW)

と呼ぶことする．一つ前のサンプルノードには遷移しないことによって，次数が低いノードに何度も留まる可能性を排除している．本論文では，後戻りしないルールを MHRW のアルゴリズムを設計し，MHRW よりも広範囲にサンプリングができていることを実験によって示した．また特徴量の推定の精度の比較実験を行い，良いサンプルノード集合を取得できていることを示した．特徴量の推定では ready-to-use 性を保っていることも確認する．最後に付録で，ready-to-use 性が保てていることを数学による証明によって示した．

2. 準　　備

本章では，提案手法のベースとなる MHRW について説明する．

2.1 定義・前提

本論文ではソーシャルネットワークをグラフ $G(V,E)$ で表す．$V=\{v_1,v_2,...,v_n\}$ はノード（頂点）の集合であり，全ノード数を $n=|V|$ とする．E はエッジの集合である．頂点 $v\in V$ と隣接しているノードの集合を $N(v)\stackrel{\text{def}}{=}\{w\in V:(v,w)\in E\}$ とする．頂点 v_i の次数を d_i または k_{v_i} と表すとする．$d_i=k_{v_i}=|N(v_i)|$ である．全てのノードの次数の総和を $D\stackrel{\text{def}}{=}\sum_{i=1}^{n}d_i=2|E|$ とする．

我々は，グラフ G に対して次のような前提を置いて議論を進める．

- G は重みなしの無向グラフである．
- G は連結グラフである．
- ノード ID のクエリによりそのノードの隣接ノードリストを取得することができる．
- クローリングによって取得できる情報は，初期ノードとクエリによって得られる情報のみである．

2.2 Metropolis-Hastings Random Walk (MHRW)

一般的なランダムウォークによるサンプリング処理の順序は，まず最初のノードを選択し，その後は反復的に処理を行う．反復的な操作では，滞在しているノードの全ての隣接ノードを取得する．そして決められた遷移確率によって次に遷移するノードを隣接ノードリストから決める．

任意のノードの定常分布が一様分布に収束するように遷移確率を調整したのが MHRW である．Metropolis-Hastings アルゴリズムは，直接サンプルするのが難しい確率分布 μ からサンプルするための一般的なマルコフ連鎖モンテカルロ法 (MCMC) である．今回の場合では，一様分布 $\mu_v=\frac{1}{|V|}$ からノードをサンプルするのが目標である．なぜなら，一様分布でサンプルすることによって ready-to-use 性を保つことができるからである．これは次のような遷移確率で達成できることが知られている．

$$P_{v,w}^{MH}=\begin{cases}\min(\frac{1}{k_v},\frac{1}{k_w}) & w\in N(v)\\ 1-\sum_{y\neq v}P_{v,y}^{MH} & w=v\\ 0 & \text{otherwise}\end{cases}\qquad(1)$$

具体的な遷移確率の例を図 1 に示した．

この遷移確率による定常分布は $\pi_v^{MH}=\frac{1}{|V|}$ であり，我々が

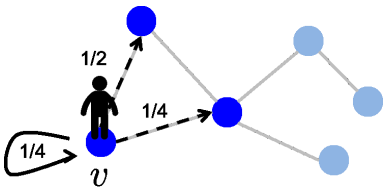

図 1　ノード v での MHRW の遷移確率

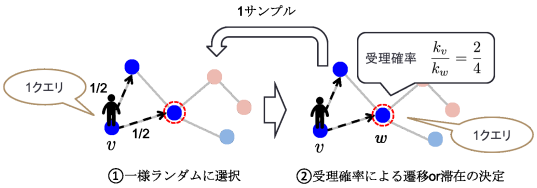

図 2　実装上の MHRW の手順

目標としている一様分布と同じである．しかし，遷移確率を算出するためには，滞在しているノードの次数と全ての隣接ノードの次数を知る必要があるため，クエリ数が膨大に増えてしまう．したがって，実装上では次のような Algorithm1 によって MHRW の遷移確率による遷移を表現することができる．

Algorithm 1 MHRW

```
sample_list ← 空のリスト
now_node ← 最初のノード
sample_list に now_node を追加
while 停止条件を満たしていない do
  neighbor_list ← now_node の隣接リスト
  next_node ← neighbor_list から一様ランダムに選択する
  p ← 一様分布 U(0,1) から乱数を生成する
  if p ≦ k_now_node / k_next_node then
    now_node ← next_node
    sample_list に now_node を追加
  else
    now_node に滞在する
    sample_list に now_node を追加
  end if
end while
```

MHRW の毎回の反復では図 2 のような処理が行われる．現在ノード v にいる時，隣接ノード w を一様ランダムに選択し，受理確率 $\min(1,\frac{k_v}{k_w})$ によって受理または拒否を決定し，受理の場合遷移を行う．拒否の場合は，現在のノード v を再びサンプルする．MHRW は現在のノードより次数が小さいノードを選択した場合，必ず受理され遷移する．逆に現在のノードより次数が高いノードを選択した場合，拒否される場合がある．これによって，高次数へのサンプルの偏りを排除している．

しかし，MHRW はランダムウォークに比べてサンプル範囲が小さいという欠点を持つ．なぜなら遷移を拒否する場合，現在滞在しているノードをサンプルするからである．さらに滞在しているノードの次数が低いほど遷移の受理確率が低くなり，何度も同じノードをサンプルする場合が起こりうる．

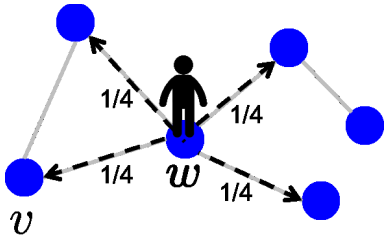

図 3　始点 ノード w での NBMHRW の遷移確率

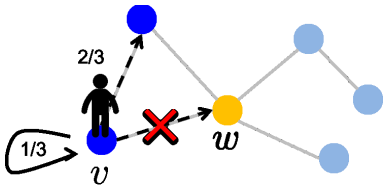

図 4　ノード w から遷移した後の ノード v での NBMHRW の遷移確率

3. 後戻りなし Metropolis-Hastings Random Walk

本章では，提案手法である NBMHRW について述べる．NBMHRW は一つ前にサンプルしたノードを避けてサンプリングを行うことで，MHRW のサンプル範囲が小さいという欠点を克服した MHRW である．一つ前にサンプルしたノードは，必ずすでにサンプルしたことがあるノードであるため，そのノードを避けることによってまだ訪れたことがないノードをサンプルする可能性が高くなり，広範囲のサンプリングを実現しやすくなると予想した．

3.1 NBMHRW の遷移確率

NBMHRW の遷移確率は基本的には MHRW と同じである．違いは，一つ前に訪れたノードを記憶し，MHRW の遷移確率でそのノードが選択された場合，もう一度 MHRW の遷移確率で選び直す点である．しかし，最初のステップの時は例外である．なぜなら，最初のステップ時は一つ前のノードが存在しないからである．この時は通常の MHRW の遷移確率から次のノードを選択する．

図 3 で最初のステップ時の NBMHRW 遷移確率を示し，図 4 でノード w から遷移した後のノード v での NBMHRW の遷移確率の例を示した．

しかし，MHRW 同様に遷移確率を算出するためには，全ての隣接ノードの次数を知る必要があるため膨大なクエリ数が必要となり実用的でない．したがって実装上では図 6 のような手順で次のノードを決定する．そのアルゴリズムを Algorithm2 に示した．

4. 実験・評価

本論文では，提案手法を評価するために既存手法である MHRW との比較を行う．

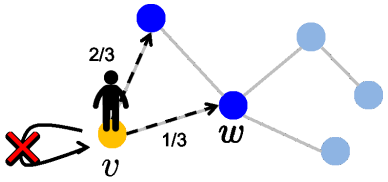

図 5　ノード v に滞在した後の ノード v での NBMHRW の遷移確率

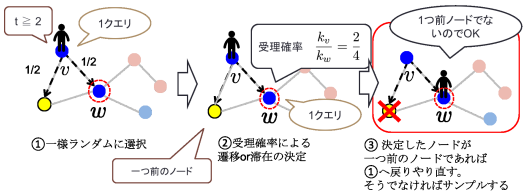

図 6　実装上の NBMHRW の手順

Algorithm 2 NBMHRW

```
sample_list ← 空のリスト
now_node ← 最初のノード
sample_list に now_node を追加
neighbor_list ← now_node の隣接リスト
next_node ← neighbor_list から一様ランダムに選択する
p ← 一様分布 U(0,1) から乱数を生成する
pre_node ← now_node
if p ≦ k_now_node / k_next_node then
  now_node ← next_node
end if
sample_list に now_node を追加
while 停止条件を満たしていない do
  neighbor_list ← now_node の隣接リスト
  pre_node ← sample_list の後ろから 2 番目のノード
  while next_node == pre_node do
    next_node ← neighbor_list から一様ランダムに選択する
    p ← 一様分布 U(0,1) から乱数を生成する
    if p ≧ k_now_node / k_next_node then
      next_node ← now_node
    end if
  end while
  now_node ← next_node
  sample_list に now_node を追加
end while
```

4.1 データセット

Stanford Network Analysis Project (SNAP) (注1) のデータセットで公開されているソーシャルネットワーク・引用ネットワークを用いて実験を行った．表 2 に各データセットの概要を示す．実用的にグラフサンプリングが行われるケースは，対象となるソーシャルネットワークは未知であるが，実験では全体像を知っているグラフデータに対して，シミュレーションを行う．

4.1.1 Amazon ネットワーク

Amazon Web サイトをクローリングして集められたネットワークである．このネットワークは購入者がどの商品とどの商

(注1)：https://snap.stanford.edu/data/

表 2　データセット

ネットワーク	全ノード数 n	平均次数	平均クラスタ係数
Amazon	334,863	5.530	0.3967
DBLP	317,080	6.622	0.6324

品が一緒に買われやすいかに注目している．ノードは商品であり，頻繁に一緒に購入される商品（ノード）に対しては無向エッジが張られる．

4.1.2　DBLP ネットワーク

DBLP computer science bibliography が提供する，コンピュータ科学の研究論文の共著者ネットワークである．ノードは論文の著者であり，一度でも共著関係になると無向エッジが張られる．

4.2　実験環境

既存手法と提案手法を Python と NetworkX(注2) を用いて実装した．本実験は MacBook Pro (Retina, 13-inch, Early 2015)，プロセッサ 3.1 GHz Intel Core i7，メモリ 16 GB で行った．

4.3　実験内容

本論文の実験では，Amazon，DBLP のそれぞれのソーシャルネットワークに対して，MHRW と提案手法である NBMHRW の 2 手法を用いて 10000 クエリを上限としノード ID のサンプリングを行った．それを 1000 回行い，そのノード ID 集合について分析を行った．始点ノードについては，始点ノード集合を 1000 個一様ランダムに選び，各々のサンプリング手法の始点ノードとした．現実にはネットワークが未知であるため，始点を一様ランダムに選択することは不可能だが，最初の方の十分多くのサンプリングノードを破棄する burn-in を行うことでクローリングでも一様ランダムに始点ノードを選択することは可能である [1]．

図 7 は 2 手法のサンプル集合の重複ありのサンプル数の平均を表している．MHRW は 2 クエリで 1 サンプルを取得するため，10000 クエリでは半分の 5000 サンプル取得している．一方で NBMHRW は一つ前のノードを選択した場合，拒否を行いもう一度サンプルをやり直すため MHRW よりサンプル数が少なくなっていることがわかる．それに対し，図 8 では，2 手法のサンプル集合の固有ノード数の平均を表している．つまり，この値が大きいほど，広範囲にサンプリングしていると言える．一般的にサンプル数が大きければ固有ノード数も大きくなるため，MHRW の方が固有ノード数が大きくなることが予想されるが，実際は NBMHRW の方が固有ノード数が大きくなっている．これは，MHRW が同じノードを何度もサンプルしてしまう可能性が高いのに対し，NBMHRW はまだ訪れていないノードに遷移しやすいためである．結果 NBMHRW の方が広範囲にサンプリングできていることがわかる．

図 9 では，2 手法のサンプル集合を用いた，グラフの特徴量推定の誤差を表している．今回は平均クラスタ係数という複雑ネットワークの密さを表す指標を用いた [2]．この実験を行うことで，2 手法のサンプル集合がグラフ全体の特徴を捉えているかどうかと，ready-to-use 性を持っているかどうかを示した．誤差の評価は正規化平均二乗誤差 [3] を用いた．正規化平均二乗誤差が小さいほど特徴量推定の精度が高く，良いサンプル集合を集められていることがわかる．結果を見ると，NBMHRW の方が正規化平均二乗誤差が小さく，NBMHRW が MHRW より優位であることがわかる．

(注2)：https://networkx.github.io/

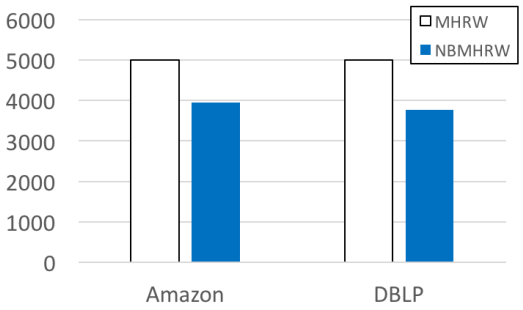

図 7　10000 クエリ当たりの重複ありサンプル数

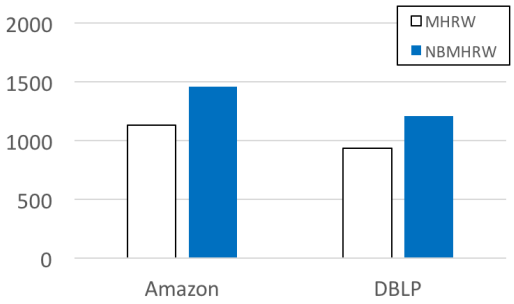

図 8　10000 クエリ当たりサンプルの固有ノード数

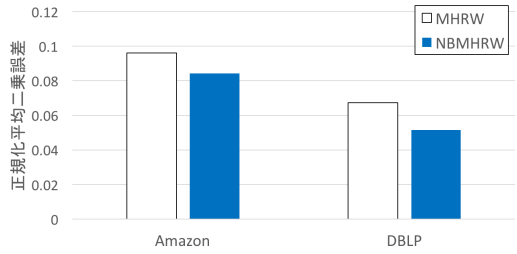

図 9　平均クラスタ係数推定の正規化平均二乗誤差

5.　まとめと今後の課題

本論文では，後戻りなし Metropolis-Hastings Random Walk によるソーシャルネットワークのグラフサンプリング手法を提案した．既存研究との差分として，ready-to-use 性を保ちつつ，広範囲にサンプリングできる手法であることを理論と実験により示した．今回の実験では，平均クラスタ係数という一つの特徴量のみ推定となったが，あらゆる特徴量について実験を行うこと，そして RWRW や他の MHRW の改良手法との比較実験を行うことを今後の課題とする．

文　献

[1]　Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina

Markopoulou. Practical recommendations on crawling online social networks. *Selected Areas in Communications, IEEE Journal on*, Vol. 29, No. 9, pp. 1872–1892, 2011.

[2] L da F Costa, Francisco A Rodrigues, Gonzalo Travieso, and Paulino Ribeiro Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, Vol. 56, No. 1, pp. 167–242, 2007.

[3] Bruno Ribeiro and Don Towsley. Estimating and sampling graphs with multidimensional random walks. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 390–403. ACM, 2010.

[4] Galin L Jones, et al. On the markov chain central limit theorem. *Probability surveys*, Vol. 1, pp. 299–320, 2004.

[5] Chul-Ho Lee, Xin Xu, and Do Young Eun. Beyond random walk and metropolis-hastings samplers: why you should not backtrack for unbiased graph sampling. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 40, pp. 319–330. ACM, 2012.

付　　録

1. 不可逆なマルコフ連鎖の生成

付録では，NBMHRW による不偏サンプリングが MHRW と同じ定常分布で達成できることを数学的証明によって示す．順序としては，MHRW が可逆性のある既約な有限マルコフ連鎖であることを示し，次に可逆性のある既約な有限マルコフ連鎖から，同じ定常分布 $\boldsymbol{\pi}$ の非可逆で既約な有限マルコフ連鎖を作ることができることを示す [5].

扱うネットワークの仮定から，既約性と有限性は自明である．ここでは MHRW の可逆性を示す．

可逆性を示すためには，詳細釣り合い条件が成立つことを証明すれば良い．つまり，

$$\pi_v P^{MH}_{v,w} = \pi_w P^{MH}_{w,v} (v, w \in V)$$

であることを示せれば良い．これは MHRW の定常分布が $\pi v^{MH} = \frac{1}{|V|}$ であることと，式 (1) で紹介した $\pi_v P^{MH}_{v,w}$ の定義から容易に証明することができる．ゆえに MHRW は可逆性を持つ．

次に，可逆なマルコフ連鎖から非可逆なマルコフ連鎖を生成できることを示す [5]．つまり，後戻りなしのランダムウォークにしても特徴量推定の時には同じ定常分布を使用できることを示し，ready-to-use 性を保っていることを証明する．

一般的な可逆性のある既約な有限マルコフ連鎖 $\{X_t \in V, t = 0, 1...\}$ は遷移確率行列 $\mathbf{P} = \{P(i,j)\}_{i,j\in V}$，定常分布 $\boldsymbol{\pi} = [\pi(i), i \in V]$ とする．任意の関数 $f : V \to \mathbb{R}$ に対して次のような推定量を定義する．

$$\hat{\mu_t}(f) \stackrel{\text{def}}{=} \frac{1}{t}\sum_{s=1}^{t} f(X_s)$$

定常分布 $\boldsymbol{\pi}$ に関する関数 f の期待値は次のように与えられる．

$$\mathbb{E}_{\boldsymbol{\pi}}(f) \stackrel{\text{def}}{=} \sum_{i\in V} \pi(i) f(i).$$

[4] では $\{X_t\}$ が定常分布 $\boldsymbol{\pi}$ の有限で既約なマルコフ連鎖であるとき，任意の初期分布 $\mathbb{P}\{X_0 = v\}, v \in V, (t \to \infty)$ に対して

$$\hat{\mu_t}(f) \to \mathbb{E}_{\boldsymbol{\pi}}(f) \; almost\, surely \, (a.s.)$$

が成立つ．ただし $\mathbb{E}_{\boldsymbol{\pi}}(|f|) < \infty$ とする．

この可逆なマルコフ連鎖から，後戻り無しのランダムウォーク，つまり同じ定常分布 $\boldsymbol{\pi}$ の非可逆で既約な有限マルコフ連鎖を作る [5].

NBRW によるランダムウォークを $\{X'_t \in V, t = 0, 1...\}$ とする．現在のノードが X'_t のとき，次のノード X'_{t+1} の決定のされ方は現在のノード X'_t だけでなく，一つ前のノード X'_{t-1} にも依存する．このため，$\{X'_t\}_{t\geqq 0}$ 自体は V の状態空間ではマルコフ連鎖になり得ない．したがって，マルコフ連鎖を作れるように次のような状態空間を定義する．

$$\Omega \stackrel{\text{def}}{=} \{(i,j) : i, v \in V \,\text{s.t.}\, P(i,j) > 0\} \subseteqq V \times V$$

$|\Omega| < \infty$ であり，$Z'_t \stackrel{\text{def}}{=} (X'_{t-1}, X'_t) \in \Omega \, (t \geqq 1)$ とする．簡単に表記するために e_{ij}は状態 $(i,j) \in \Omega$ を表すとする．ただし $e_{ij} \neq e_{ji}$である．$\mathbf{P'} \stackrel{\text{def}}{=} \{P'(e_{ij}, e_{lk})\}_{e_{ij}, e_{lk} \in \Omega}$ を状態空間 Ω 上の既約なマルコフ連鎖 $\{Z'_t \in \Omega, t = 1, 2, ...\}$ の遷移確率行列とする．この時定義より $P'(e_{ij}, e_{lk}) = 0 \, (\forall j \neq l)$．マルコフ連鎖 $\{Z'_t\}$ の定常分布 $\boldsymbol{\pi'} \stackrel{\text{def}}{=} [\pi'(e_{ij}), e_{ij} \in \Omega]$ が次のように与えられたとする．

$$\pi'(e_{ij}) = \pi(i)P(i,j), \;\; e_{ij} \in \Omega$$

この時元のランダムウォーク $\{X_t\}$ の可逆性から $\pi'(e_{ij}) = \pi'(e_{ji})$ である．また，定常状態中のノード j にいる NBRW によるランダムウォーク $\{X'_t\}$ の確率は，元の可逆マルコフ連鎖 $\{X_t\}$ の定常分布の $\pi(j) \, (\forall j \in V)$ と同じである．

$$\sum_{i\in V : e_{ij} \in \Omega} \pi'(e_{ij}) = \sum_{i \in V} \pi(i)P(i,j) = \pi(j), \forall j \in V$$

最初の等号は，$P(v,w) = 0, \forall (v,w) \notin \Omega$ によって導かれる．特に任意の関数 $f : V \to \mathbb{R}$ に対して，もう一つの関数 $g : \Omega \to \mathbb{R}$, $g(e_{ij}) = f(j)$ としたとき，

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\pi'}}(g) \;&=\; \sum_{e_{ij}\in\Omega} g(e_{ij})\pi'(e_{ij}) = \sum_{j\in V}\sum_{i\in V} f(i)\pi(i)P(i,j) \\ &= \sum_{j\in V} f(j)\pi(j) = \mathbb{E}_{\boldsymbol{\pi}}(f)\end{aligned}$$

大数の法則より

$$\frac{1}{t}\sum_{s=1}^{t} g(Z'_s) = \frac{1}{t}\sum_{s=1}^{t} f(X'_s) \to \mathbb{E}_{\boldsymbol{\pi'}}(g) = \mathbb{E}_{\boldsymbol{\pi}}(f) \, a.s.,$$

すなわち，$\sum_{s=1}^{t} g(Z'_s)/t = \sum_{s=1}^{t} f(X'_s)/t$ は $\mathbb{E}_{\boldsymbol{\pi}}(f)$ の不偏推定量である．