

P2P ネットワーク上のデータに対する偏りのない機械学習手法

高橋 良希[†] 首藤 一幸[†]

[†] 東京工業大学

〒 152-8552 東京都目黒区大岡山 2-12-1

あらまし 非構造化 P2P ネットワークにおいて、プライバシーなどの理由から各ノードの保有するデータを移動させずに、ネットワーク上に分散されたデータを学習する手法が研究されている。このとき、データを移動させる代わりに学習モデルを移動させることで学習を進める方法があるが、モデルが単純にネットワーク上をランダムウォークする場合、入次数の大きなノードのデータを多く学習してしまうような偏りが生じる。既存手法では、このような偏りを隣接ノードリストを繰り返し更新することで削減する手法が存在するが、隣接ノードを繰り返し更新することはプライベート IP アドレスを使用する端末に対して容易に通信を開始できないような問題を生じる。本研究では、隣接ノードリストを更新せずとも、選択確率を用いることで、ネットワーク上のデータをバランスよく学習する手法を提案する。また、この手法によって隣接ノードリストの更新を行わずとも、既存手法と同等の精度を達成することをシミュレーションを用いた実験で示す。

キーワード P2P, オンライン機械学習, ゴシッププロトコル, Metropolis-Hastings random walk

1. はじめに

機械学習はメールのスパムフィルタリングや通販サイトの推薦システムなど、生活の様々な面で活用されている。これらの学習に必要なデータは、通常サーバ上に集中して管理されるが、P2P のような非集中型のシステムの場合、多くのデータは端末となるピアに蓄積される。端末として PC やスマートフォン、タブレットなどを想定すると、保有するデータのうちユーザのプロフィール情報や検索履歴、センサー情報などの個人情報に関わるデータはプライバシーの問題からピア間で通信することができない。

このとき、データを移動させる代わりに学習モデルを移動させ、モデルがノードを訪問するたびにそのノードの保有するデータを逐次的に学習することで、ネットワーク上のデータを学習することができるが、非構造化ネットワークでは入次数の大きなノードのデータを多く学習してしまうように、学習データサンプルに偏りが生じる。gossip learning [7] では、各ノードが隣接ノードリストを繰り返し更新することによって、グラフ構造によるサンプルデータへの影響を低減している。これにより学習データのサンプル列が一様に近づくように収束している。しかし、既存研究における隣接ノードリストの更新は、プライベート IP アドレスを持つノードに対して容易に通信を開始できないような、NAT 越えの問題を発生させる。

我々は隣接ノードリストを更新せずとも、ノード間に選択確率を設定することで学習データサンプル列の偏りを低減する手法を提案する。提案手法の選択確率は、Metropolis-Hastings random walk (MHRW) [10] に倣った確率を設定する。これにより入次数の大きなノードは選択されにくく、入次数の小さなノードは選択されやすい特性を享受する。

本論文では、まず準備としてオンライン学習アルゴリズムと、既存研究であるモデル送信による学習手法を紹介し、ノード選

択の手法における問題点を提起する (第 2 章)。次にそれを解決するためのノード選択の手法を提案し (第 3 章)、提案した手法と既存手法、ナイーブな手法をそれぞれ実験し結果を考察する (第 4 章)。次に本研究の関連研究を述べ (第 5 章) 最後に本研究のまとめと今後の課題を述べる (第 6 章)。

2. 準備

ネットワーク上に分散された移動できないデータに機械学習を適用する場合、中心となるノードにデータをまとめて一度にまとめて処理することができないため、各訓練データをオンライン学習によって逐次的に学習する。オンライン学習を用いて P2P ネットワーク上のデータを学習する gossip learning [7] では、隣接ノードリストの更新を用いたネットワーク上の一様ランダムなピアサンプリングと、ゴシッププロトコルを用いたモデル送信、オンライン学習によるモデル更新を繰り返すことで、学習モデルがネットワーク上を移動しながら学習が進む。

本章では、解決すべき問題を明確に提示するために、まずオンライン学習と gossip learning [7] を紹介し、次に P2P 上の一様ランダムなピアサンプリング手法である Newscast [8] を紹介する。最後に解決すべき問題を提示する。

また、本論文では分類問題を例に話を進めるが、他のオンライン学習においても適用可能である。

2.1 分類問題におけるオンライン学習

分類の目的はある入力ベクトル \mathbf{x} に対し、いくつかの離散クラスのうちのいずれかを割り当てることである。訓練データが与えられたとき、パラメータが \mathbf{w} のモデルによる i 番目の入力ベクトル \mathbf{x}_i に対する予測と、実際のクラスラベルとの間に誤差を観測できる。モデルによる i 番目のデータに対する誤差関数を $E_i(\mathbf{w})$ とすると、 n 個のデータ全体の誤差関数の一つとして

Algorithm 1 Gossip Learning Framework

```
1: currentModel ← initModel()
2: loop
3:   wait( $\Delta$ )
4:    $p \leftarrow \text{selectPeer}()$ 
5:   send currentModel to  $p$ 
6: end loop
7: procedure ONRECEIVEMODEL( $m$ )
8:    $m.\text{updateModel}(x, y)$ 
9:   currentModel ←  $m$ 
10: end procedure
```

$$E = \sum_{i=1}^n E_i(\mathbf{w}) \quad (1)$$

を定義できる。学習の目的は訓練データに対する誤差関数を減少させることによって、未知のデータをうまく分類できるようにすることである。データ全体の誤差を減少させる最適化手法の一つである勾配降下法では、誤差関数を減少させるような τ 回目のパラメータの更新を

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta_\tau \nabla E \quad (2)$$

に従って行う。一方、一つのデータの誤差に対して逐次的にパラメータを更新するオンライン学習手法の一つである確率的勾配降下法では、誤差関数が(1)のように各データ点の誤差の和で表される場合、パラメータ \mathbf{w} を

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta_\tau \nabla E_i \quad (3)$$

に従って更新する。ここで、確率的勾配降下法の収束性を保証するために、学習におけるパラメータ η_τ が $\sum_\tau \eta_\tau^2 < \infty$ かつ $\sum_\tau \eta_\tau = \infty$ の条件を満たさなければならないことに注意する[4]。オンライン学習では、新しいデータを得るたびにこの更新を行い、誤差が収束するまで繰り返す。ここでオンライン学習の重要な性質として、誤差関数が凸関数である場合、データサンプルが互いに独立かつ同一の確率分布から生成されると仮定すると、学習によるパラメータはほとんど確実に最小解に収束することが保証される[4]。つまり、各ノードが学習データを保有している場合、ノードをネットワーク上から独立かつ一様ランダムに選択し、モデルを送信することで、モデルパラメータは誤差の最小解に収束することが保証される。次に紹介するgossip learning [7]では、隣接ノードリストを十分な回数シャッフルした後に、ランダムな隣接ノードにモデルを送信することで、ネットワーク上の一様ランダムなノードのサンプリングを行っている。

2.2 gossip learning

Algorithm 1 に gossip learning [7] の擬似コードを載せる。各ノードは、まずモデルパラメータを適当な値で初期化し、次に Δ で示されるインターバルでモデルの送信を行う。モデルの送信先はselectPeerによって隣接ノードの中から一様ランダムに選ばれるが、Newscast [8]と呼ばれるピアサンプリング手法を用い、隣接ノードリストは各サイクルごとに繰り返しシャッフルされる。モデルを受け取ったノードは、updateModelによって自身の保有するデータに基づいたオンライン学習を用いてモデルパラメータを更新する。

Newscast [8]では、隣接ノードのリストを繰り返し更新することにより、一様ランダムなノードのサンプル列を得る。この手法では、各ノードは隣接ノードのリストをビュー (view) と呼ばれる構造体で管理し、一定のサイクルごとにプッシュ (push) とプル (pull) によってビューを繰り返し交換する。プッシュでは、自身の持つビューを自身のアドレスとともにランダムな隣接ノードに送信する。これを受け取ったノードは、受け取ったノードアドレスのリストと自身のビューを適切にマージして、これを新しいビューとして更新する。プルでは、ランダムな隣接ノードに、そのノードが保有するビューを要求する。相手からビューの返信があれば、これと自身のビューをマージし、新しいビューとして更新する。各サイクルおきにこの操作を十分な回数行うことで、隣接ノードからランダムに選択するノードはネットワーク上から独立かつ一様ランダムに選択されたサンプル列に収束する。

しかしこの手法には問題点があり、Newscast [8]によって新しく参照されたノードがプライベート IP アドレスを用いている場合、NAT 越えの問題からこのノードに対して容易に新しく通信を開始することができない。NAT 越えの問題とは、プライベート IP アドレスをグローバル IP アドレスに変換する NAT という技術を用いた際に、内部ネットワークから外部ネットワークへの通信は開始できるが、外部ネットワークから内部ネットワークに対しては、リクエストを適切な内部のホストに関連付けられないために通信を開始できないという問題である。現在、家庭用マシンからクラスタまで、多くのコンピュータがプライベート IP アドレスを使用しているため、この手法は現実的ではない。この問題を避けるために、我々は隣接ノードを更新せずとも、選択確率を用いるノード選択の手法を提案する。

3. 提案手法

我々が提案する手法では、モデルが学習する訓練データのサンプル列をネットワーク上からバランスよく取得するために、選択確率に基づいたモデルの送信先ノード選択を行う。Algorithm 2 に提案手法によるノード選択手法の擬似コードを示し、図 1 に本手法によるノード選択確率の例を示す。

選択確率は Metropolis-Hastings random walk (MHRW) に倣った確率を設定する。図 1 から分かるように、この選択確率は入次数の大きなノードに対して低い確率を、入次数の小さなノードに対して高い選択確率を設定し、ノードの訪問確率を調整している。ただし、この選択確率により取得するサンプル列が一様分布に従う収束を達成する保証は現時点で確認できてい

Algorithm 2 Proposed Peer Select Algorithm

```
1: function MHRW-BASEDPEERSELECT()
2:    $p \leftarrow$  select a neighbor at random
3:    $\alpha \leftarrow \frac{\text{indegree}(self)}{\text{indegree}(p)}$ 
4:   if random.uniform(0, 1) <  $\alpha$  then return  $p$ 
5:   else return  $self$ 
6:   end if
7: end function
8:
9: function MH-BASEDPEERSELECTWITHOUTSELFSEL()
10: loop
11:    $p \leftarrow$  select a neighbor at random
12:    $\alpha \leftarrow \frac{\text{indegree}(self)}{\text{indegree}(p)}$ 
13:   if random.uniform(0, 1) <  $\alpha$  then return  $p$ 
14:   end if
15: end loop
16: end function
```

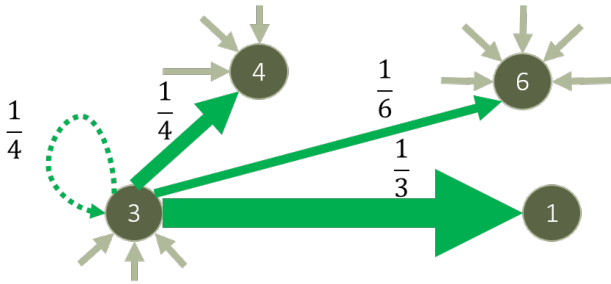


図1 提案手法によるノード選択確率の例

ない。

具体的にモデルの送信先ノードは次のように決定する。まず隣接ノードからランダムにノード p を選出し、自ノード $self$ と p の入次数の比に応じた採択率 α を決定する。 α が 1 よりも大きな場合は p を選択し、 α が 1 を下回る場合は、確率 α で p を選択し、確率 $1 - \alpha$ で $self$ を選択する。

この手法を用いるために、各ノードは隣接ノードの入次数を取得する必要がある。各ノードは自身の入次数の情報を通信し合わなければならない。しかし、このための通信コストは、モデルを受け取った際の返信に自身の入次数の情報を付随させることで小さく抑えることが可能である。このとき、初期状態に一樣な選択確率を仮定した場合でも、モデル送信を数回繰り返すだけで正確な選択確率を設定できるため、学習に大きな影響を与えない。

表1 実験マシンの構成

OS	Ubuntu 14.04.2 LTS
CPU	Intel 2.4 GHz Xeon E5620
メモリ	32 GB RAM
Java	Java SE 8 Update 45

4. 実験

表1に実験を行ったマシンの構成を示す。実験はイベントベースのP2PシミュレータであるPeerSim^(注1)上で行った。

a) データセット

学習に用いるデータセットには、UCIデータベースリポジトリ^(注2)から入手したSpambaseを用いる。データ数は4601個で、各データは57つの特徴量と0か1のクラスを持つ。各クラスのデータ数の比は1813:2788で、ランダムに選出した461個のデータを検証用データとし、残りの4140個のデータを訓練用データとした。

b) ネットワーク

ネットワークにはBarabasi-Albert model (BA model) [1] を有向グラフに適用して生成したネットワークを使用する。BA modelは多くの実ネットワークが有する特徴の一つであるスケールフリー性を擬似的に発生させるアルゴリズムである。スケールフリー性とは、わずかなノードが大きな次数を持ち、多くのノードが小さな次数を持つ性質であり、Gnutella [6] のような実際のP2Pネットワークもこの性質を有する。

また、入次数が0のノードに対してはランダムなノードの隣接ノードにこのノードを追加することで、全ノードの入次数が必ず1以上になるように調整を加えた。これは入次数が0のノードが存在するときにこのノードが常に初期化された学習モデルをネットワーク上に配信することを防止するためである。このアルゴリズムによって生成されたグラフは、次数分布がべき乗則に従っているため(図2)、スケールフリー性を有していることが確認できる。ネットワークサイズは学習データ数と同じ4140とし、各ノードの出次数は10とした。また、実験は同様のアルゴリズムで生成した5種類のネットワークに対して行った。

c) 学習方法

実験には三つのノード選択の手法を用い、それぞれの結果を比較する。一つ目は既存手法のノード選択方法であるNewscast [8] を用いる手法で、スケールフリーネットワークを初期状態として、各ノードは一定の時間区間ごとに隣接ノードのリストであるビューを交換しながら、ビューから一様ランダムに選んだノードにモデルを送信する。二つ目は単純ランダムウォークに倣うノード選択手法で、各ノードは固定のビューの中から一様ランダムに隣接ノードを選択し、学習モデルを送信する。三つ目はMHRWベースのノード選択手法である。

また、モデルの更新にはサポートベクタマシン (SVM) とロジスティック回帰の二つの手法に対し確率的勾配降下法を用い

(注1) : <http://peersim.sourceforge.net/>

(注2) : <https://archive.ics.uci.edu/ml/datasets.html>

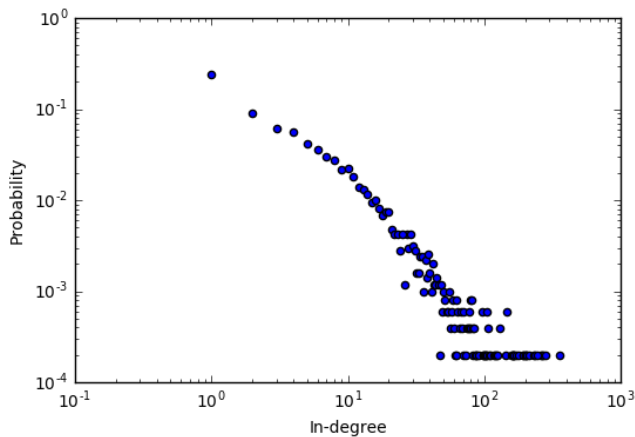


図2 入次数の分布

たアルゴリズムを用いた。また、二通りアルゴリズムに対し、モデル受信時に直前のモデルをマージする場合と、上書きによって書き換える場合の計四通りの実験を行った。

d) 検証方法

検証には検証用データに対する各ノードの予測誤差の推移を用いる。学習が一定期間進むごとに、各ノードは最新の学習モデルで共通の検証用データに対して予測を行い、実際のクラスとの間に0-1誤差を算出する。この誤差の総和を検証用データセットの数で割った値をそのノードの予測誤差とする。実験では各ノードの予測誤差の推移を観測することで、モデルの精度と収束速度を確認する。

e) 実験結果

図3に全ノードの予測誤差の平均値の推移を示す。実験は乱数を変えて生成した5種類のネットワークを用いたが、いずれも以下に述べる内容とほぼ同様の結果を得たため、代表として一つの実験結果を示す。x軸は学習のサイクル。y軸は予測誤差を示し、誤差が下がるほど全体のモデルの精度が良くなっていることがわかる。このグラフでは、まず全手法において予測誤差の平均が1ノードのSVMによる予測誤差を大きく上回っていることがわかる。ここで、既存研究であるgossip learningの論文[7]では1ノードのSVMの誤差まで収束をしているが、今回はこれを再現できなかったため現時点の結果として本論文には記載する。

その前提で考察を行うと、提案手法は隣接ノードリストの更新を行わないにも関わらず、既存手法とほぼ同程度の収束値と収束速度を達成していることがわかる。しかし、マージ処理を行わない学習アルゴリズムでは、提案手法・既存手法ともにランダムウォークに倣ったナイーブな選択手法に対して精度で逆転されている。これについては現在調査中なため、今後の研究

課題としたい。ただ初期の収束速度に関しては、いずれの学習アルゴリズムにおいてもランダムウォークに倣った手法よりも優れている。

図4は上記の実験において、学習終了時である35000サイクル後の各ノードの予測誤差を大きい順にプロットしたグラフである。x軸は予測誤差の大ききでソートされたノードを表し、y軸はx番目に誤差の高かったノードの予測誤差を表す。全手法において、予測誤差が0.5を上回るノードから1ノードにおける誤差に近い誤差を持つノードまで、満遍なく存在することがわかる。このグラフにおけるほぼ学習ができていないモデルに対して特に考察をすることで、全モデルが1ノードにおける誤差の値まで収束することを今後の目標としたい。

5. 関連研究

P2P上のデータを対象とした機械学習手法は過去にいくつか存在する。P2P上の機械学習の主な目標の一つに、集中型のアプローチによって得られる学習モデルの精度にできる限り近づくことが挙げられる[5][2]。しかし、集中型の学習と比べ、P2Pを対象とした学習手法には、スケーラビリティや、ピアの挿入や離脱、学習データの増減、耐故障性、プライバシー、セキュリティなどいくつかの考慮すべき課題がある[5][3]。また、ネットワーク上の広範囲のデータを予測に反映させるために、各所の局所的なデータを学習したモデルをマージすることにより新しいモデルを生成する手法[2][12][3][9]や、各ノードによる投票によって予測を行う手法[11]が存在する。モデルを送信することで学習を行うこれらの研究では、通信先となるノードをランダムな隣接ノードから選択したり[9],[12]、分散ハッシュテーブルを用いて選択したり[3]するが、本研究では、モデルの送信先ノードを度数に従った確率で決定する点においてこれらの研究とは異なる。

6. まとめと今後の研究課題

本論文では、隣接ノードリストを更新せずとも、モデルの送信先ノードをMHRWに倣った選択によって決定することで、ネットワーク上のデータをバランスよく学習する手法を提案した。また、シミュレーションを用いて実験を行い、提案手法は既存手法のデメリットを排しながらも既存手法と同程度の予測精度と収束速度を達成することを確認した。

また、今後の研究課題の一つに全ノードの誤差の収束値を最小にすることが挙げられる。これは学習サンプルを一様分布に従うように取得するとともに、サンプルを独立に取得することによる改善が考えられる。今回提案手法として用いたランダムウォークベースのノード選択手法では、モデルが隣接ノードに沿ったデータを学習するため、学習するデータのサンプル列はサンプル間の相関が強くなる。ネットワーク上の学習データを独立に取得するためには、例えばn個飛ばしでサンプリングを行うなど、直前にサンプリングしたデータとの相関を十分に減少させる方法がある。ただし、このようにデータを間引くサンプリングは十分なnの値を推定する必要があり、また一回のサンプリングに必要なノード間の通信回数が増えるという問題も

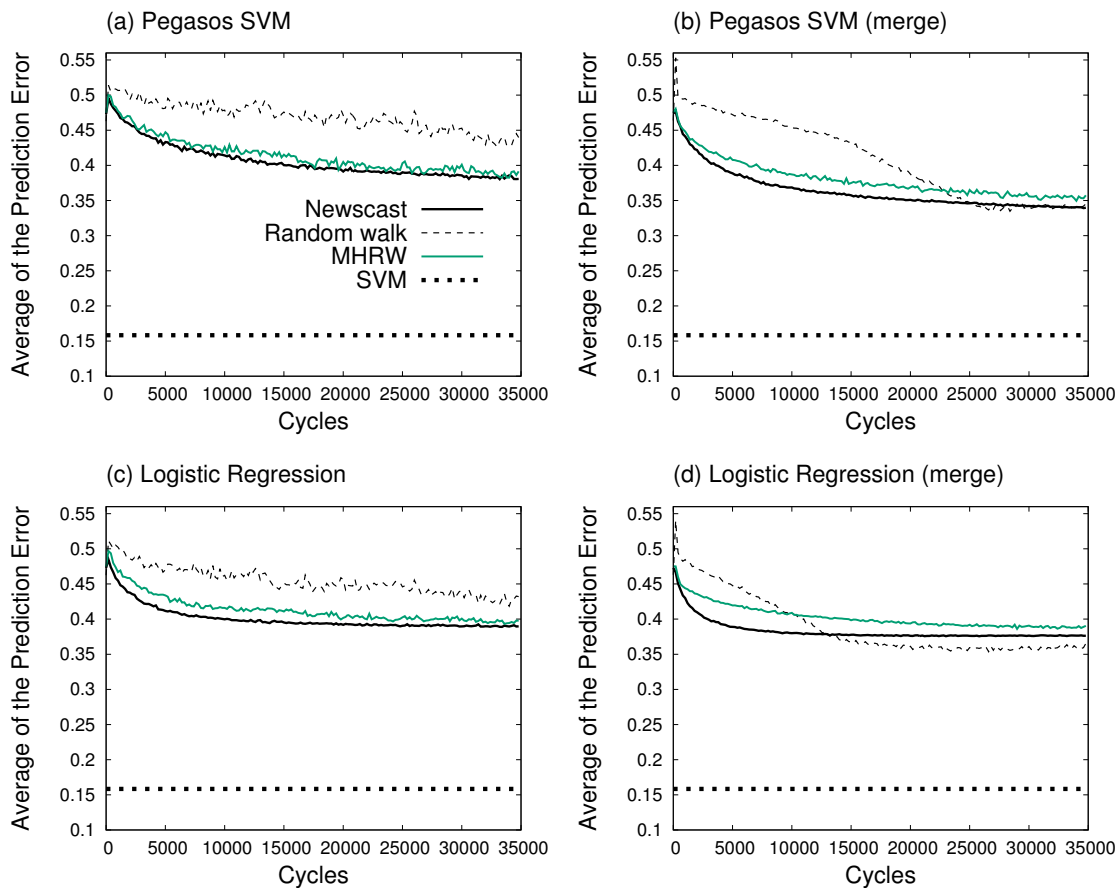


図 3 全ノードの予測誤差の平均

発生する。

また、別の研究課題として適切な終了判定を設定することが挙げられる。本研究では学習の終了判定を行わず、全ノードの予測誤差の平均がある程度変化しなくなった時点进行学习の終了としたが、実際のシステムでは各ノードが全体のノードの予測誤差を知ることは困難なため、各ノードの終了判定を考慮する必要がある。例えば、自ノードの現在保有するモデルの予測誤差がある値以下になった場合にゴシッププロトコルによるモデルの送信を停止するような終了判定が考えられる。ただし、この終了判定では終了判定を下したノードのデータを他の学習モデルが学習できなくなるという問題が挙げられる。

謝 辞

本研究は JSPS 科研費 25700008 および 16K12406 の助成を受けたものです。本研究の一部は、国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務として行われました。

文 献

- [1] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, Vol. 74, No. 1, p. 47, 2002.
- [2] Hock Hee Ang, Vivekanand Gopalkrishnan, Steven CH Hoi, and Wee Keong Ng. Cascade rsvm in peer-to-peer networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 55–70. Springer, 2008.
- [3] Hock Hee Ang, Vivekanand Gopalkrishnan, Wee Keong Ng, and Steven Hoi. Communication-efficient classification in p2p networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 83–98. Springer, 2009.
- [4] Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, Vol. 17, No. 9, p. 142, 1998.
- [5] Souptik Datta, Kanishka Bhaduri, Chris Giannella, Ran Wolff, and Hillol Kargupta. Distributed data mining in peer-to-peer networks. *IEEE Internet Computing*, Vol. 10, No. 4, pp. 18–26, 2006.
- [6] E. Gregori, L. Cherkasova, G. Cugola, F. Panzieri, and G.P. Picco. *Web Engineering and Peer-to-Peer Computing: NETWORKING 2002 Workshops, Pisa, Italy, May 19-24, 2002, Revised Papers*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003.
- [7] István Hegedűs. *Gossip-Based Machine Learning in Fully Distributed Environments*. PhD thesis, szte.
- [8] Márk Jelasity, Wojtek Kowalczyk, and Maarten van Steen. Newscast computing. 2012.
- [9] Wojtek Kowalczyk, Nikos A Vlassis, et al. Newscast em. In *NIPS*, pp. 713–720, 2004.
- [10] 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.
- [11] Ping Luo, Hui Xiong, Kevin Lü, and Zhongzhi Shi. Distributed classification in peer-to-peer networks. In *Proceed-*

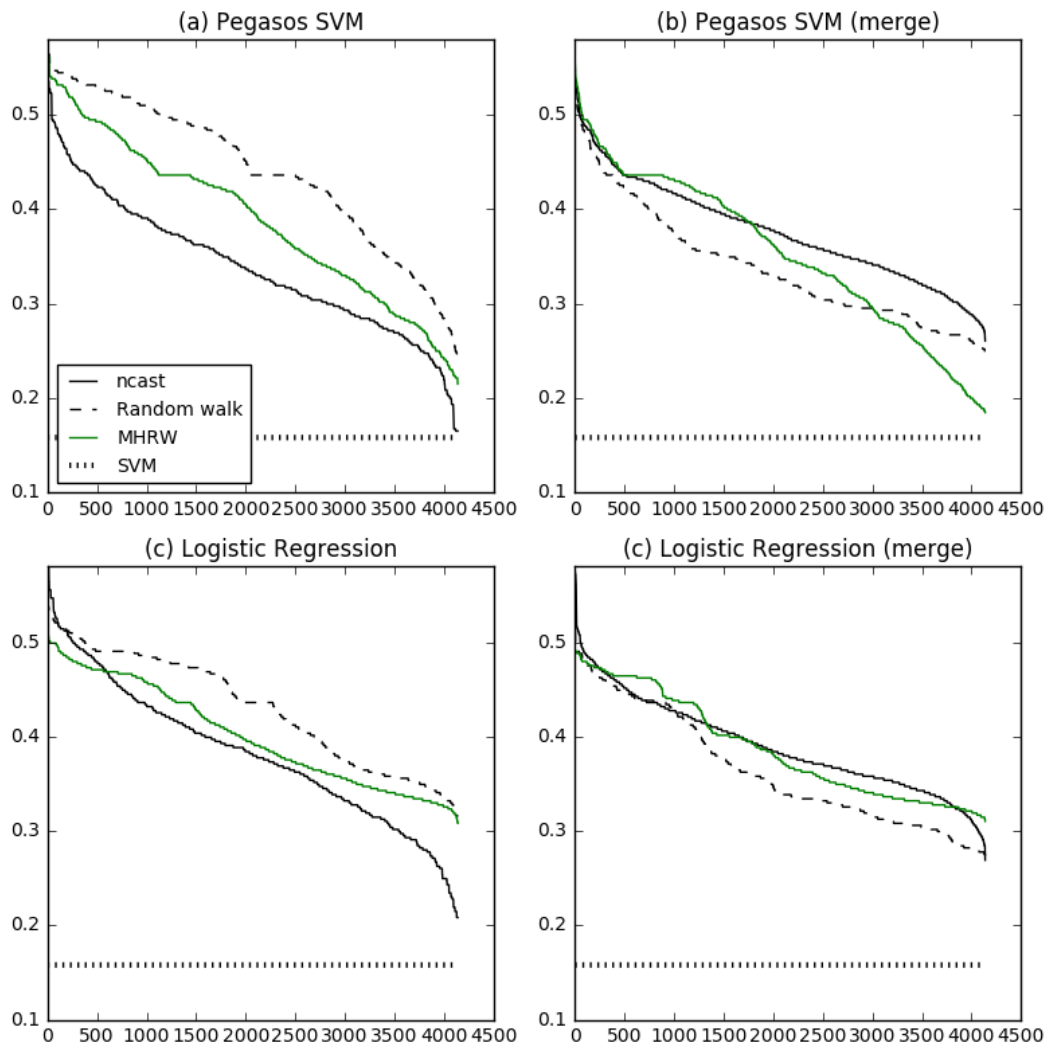


図 4 35000 サイクル後の各ノードの識別誤差の分布.

ings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 968–976. ACM, 2007.

- [12] Stefan Siersdorfer and Sergej Sizov. Automatic document organization in a p2p environment. In *European Conference on Information Retrieval*, pp. 265–276. Springer, 2006.