

深層強化学習を用いたノード分類

柳田 雄輝[†] 加藤 誠^{††}

[†] 筑波大学 知識情報・図書館学類 〒 305-8550 茨城県つくば市春日 1-2

^{††} 筑波大学 図書館情報メディア系 〒 305-8550 茨城県つくば市春日 1-2

E-mail: [†]tynagi2@klis.tsukuba.ac.jp, ^{††}mpkato@acm.org

あらまし 本研究ではグラフのノード分類に用いるノードを探索でき、かつ、分類根拠を提示できるノード分類手法を提案する。この手法ではグラフ上のパスを根拠にすることで、様々なノード間の関係を捉えることができ、多様な根拠に基づくノードラベルの分類を可能とする。パスを根拠とした分類には強化学習を用い、エージェントに様々なパスを探索させることによって、分類根拠としてふさわしいパスを獲得する。人工データを用いた実験の結果、従来手法とは異なる根拠に基づいて分類が行えることを確認することができた。

キーワード 深層強化学習, グラフニューラルネットワーク, ノード分類, 解釈可能性, データマイニング

1 はじめに

グラフ構造は様々な対象をモデル化できるため、グラフ構造を解析することは重要である。例えば、分子構造は原子をノード、結合をエッジで表すことで表現でき、SNS (Social Networking Service) はユーザや投稿内容をノード、いいねやフォローといったユーザのアクションをエッジで表すことで表現できる。このように、グラフ構造は高い表現力を持つだけでなく、分野を問わず分析対象をモデル化できるため、その解析は重要であるといえる。グラフ構造の解析によって、ユーザ行動や分子の反応などの予測や改善が可能となる。例えば SNS においては、総務省情報流通行政局情報通信政策課情報通信経済室によると、日本人の 8 割がソーシャルメディアを利用しており、その改善には大きな影響が期待される [1]。

このような背景から、グラフ構造の解析手法は広く提案されている。近年では Kipf と Welling が提案した GCN (Graph Convolutional Networks) など、グラフ構造にニューラルネットワークを適用しグラフのノードラベルを分類する手法が提案されている [2]。Kipf と Welling は論文をノード、論文にあるキーワードが含まれているかどうかを表す Bag-of-Words ベクトルをノードが持つ特徴量、論文が引用または引用されている関係をエッジとしたグラフに対して GCN を適用した。これによって、論文に含まれているキーワードに加え、引用または引用されている論文のキーワードも加味したノードの特徴量ベクトルを得ることができている。しかし、既存のノード分類手法ではグラフ内の全ノードについて計算する必要があり、計算に時間がかかってしまうことに加え、分類の予測を妨げる情報を含んでしまう可能性がある。別の例として、図 1 に SNS を表すグラフの例を示す。このグラフでは、ユーザや投稿内容をノード、いいねやフォローといったユーザのアクションをエッジで表している。例えばユーザ A がスパムであるかどうかを分類するタスクを想定する。既存のノード分類手法では各ノードについて集約を行う必要があり、ユーザ A へのノードの情報を集約さ

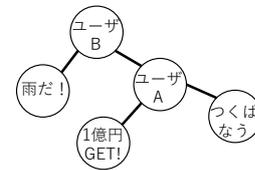


図 1 SNS のグラフの例

せるには、ユーザ A に隣接するユーザ B, “1 億円 GET!”, “つくばなう” の 3 ノードの情報を集約させる必要がある。これら 3 ノードの情報を集約させるには、ユーザ B, “1 億円 GET!”, “つくばなう” の 3 ノードそれぞれについても隣接するノードを集約する計算が必要となる。したがって、結果的に全ノードについて集約の計算を行う必要があり、計算に時間がかかってしまう。加えて、上記の集約計算の結果、ユーザ A がスパムかを予測するにあたり必要のない情報まで含んでしまう可能性がある。図 1 では、“1 億円 GET!” というユーザ A の投稿内容はスパム分類をするにあたり有益な情報と考えられるが、ユーザ B およびユーザ B の投稿内容の情報は必要がないと考えられる。このように分類の予測にあたり有益でない情報を含んでしまった結果、適切な予測ができなくなる可能性がある。

本論文では、分類に用いるノードを探索でき、かつ、どのようなノードに基づいてノードを分類するかを制御可能とする手法を提案する。この手法ではグラフ上のパスをノード分類の根拠として用いることで、ノード分類を行う上で核となる情報を持つ部分だけを捉えることができ、かつ、多様な根拠に基づくノードラベルの分類を可能とする。例えば SNS のグラフにおいて、ユーザがスパムアカウントであるかどうかを予測するノード分類タスクを想定する。このとき、あるユーザがスパムアカウントであるかどうかの予測結果とともに、スパムと判定した根拠となるパスの提示を可能とする。図 1 の例において、パス (ユーザ B, ユーザ A, “つくばなう”) を分類根拠としてユーザ B がスパムでないと判定されたとする。この場合、ユーザ B がフォローしているユーザ A の投稿内容 “つくばなう” を根拠にユーザ B をスパムでないと分類したと解釈できる。このよ

うにしてパスを分類根拠にノードを分類する。

パスを根拠とした分類には強化学習を用い、エージェントにグラフ上のパスを探索させることによって、分類根拠としてふさわしいパスを学習する。強化学習を用いることで、エージェントによってパスを探索的に発見できるほか、学習後には分類根拠としてふさわしいパスのみを得ることができる。各エージェントは分類したいノードを起点にグラフ上を探索し、規定ステップ数以内に分類に適したパスをたどれたと判断した場合には成功という行動を取るようにする。規定回数探索を行い、各エージェントの中で最も多く成功という行動を採用したエージェントに基づいてノードラベルを予測する。エージェントがたどるパスの学習には Q 学習を用いる。エージェントがたどったノード系列を入力とし、隣接する各ノードへの遷移確率と成功とする確率を出力とする関数に基づきパラメータを更新する。

実験にあたっては、次の 2 種類の人工データを作成し実験に用いた：(1) ノードが観測可能なノードラベルとエッジラベルを持つが特徴量は持たないデータ (2) ノードが特徴量を持つが観測可能なノードラベルやエッジラベルを持たないデータ。いずれのデータもおおよそ 2,000 個のノードを持つが、このうち 100 個のノードは観測できないノードラベルを持つ。本論文の実験においては、この観測できないノードラベルを予測するのが実験内容となる。

実験の結果、既存手法とは異なる根拠に基づいて分類が行えることを確認することができた。

この論文における貢献は以下の 2 つである：

- ノード分類タスクにおいて、分類根拠をパスに基づいて提示する手法を提案した。
- 人工データを用いた実験の結果、提案手法が従来手法とは異なる分類および可視化ができることを示した。

本論文では、2 節にてグラフのノード分類および、強化学習のグラフへの適用に関する関連研究について述べる。3 節ではノード分類タスクの問題設定と提案手法の詳細について述べる。4 節では用いたデータセットと実験の設定、結果を示す。5 節では本論文の結論を述べる。

2 関連研究

本節ではグラフのノード分類および、強化学習のグラフへの適用に関する関連研究について述べる。

従来のノード分類手法としては DeepWalk [3] や LINE [4], node2vec [5] などが提案されている。これらの手法では、各ノードを起点としてランダムウォークを行い、ランダムウォークの過程でたどったノードの系列を用いてノードの埋め込みを得る。その後得た各ノードの埋め込みを機械学習モデルの入力とすることでノード分類を行う。しかしこれらの手法ではノードが持つ特徴量を組み込めない限界を持つ。本論文ではノードが特徴量を持つ場合と持たない場合それぞれのモデルを提案するが、ノードが特徴量を持たない場合のモデルについては類似している。しかし、本論文で提案するノードが特徴量を持たない場合のモデルではノードやエッジが持つラベルを入力に組

み込むため、無向グラフを前提としているこれらの既存研究とは異なる。

ノードが特徴量を持つ場合のノード分類手法としては、主にグラフニューラルネットワークを用いた分類手法が見られる。これは Scarselli らが提案した、隣接ノードや隣接エッジのラベルを集約することでグラフ構造をニューラルネットワークに適用できる手法が元となって広く行われている [6]。この手法ではノード分類問題ではなく、あるグラフが別のグラフの部分グラフとなるかを判定する問題へ適用しているが、ノードやエッジの情報を元に分類問題を解く点では共通している。この手法は隣接するノードの情報しか得られないため、隣接していないノードの特徴を得るためにはその都度学習を行う必要がある限界を持つ。Kipf と Welling が提案した GCN では、グラフ構造の畳み込みによって各ノードがグラフ構造を加味した特徴量を持つように半教師あり学習を行い、ノードラベルを分類している [2]。この手法では隣接ノードに重み付けを行わずに集約を行い学習するが、Veličković らはアテンションによってノードに重み付けをした上で集約する教師あり学習手法、GAT (Graph Attention Networks) を提案している [7]。特定の分野へ適用した手法としては、Choi らの医学の知見を知識グラフで表し、グラフ構造を加味した表現をアテンションを用いて学習する手法が挙げられる [8]。この他にも隣接するノードを全て集約するのではなく、サンプリングによって学習を効率化する手法や、複数のラベルを持つ有向グラフにグラフ畳み込みネットワークを適用する手法などが提案されている [9], [10], [11], [12]。しかし、いずれの手法でも隣接ノードの特徴を用いて学習するため、ノードラベルを予測する際には結果的にグラフ内の全ノードについて計算する必要がある。全ノードについて計算をする場合、分類の予測の妨げとなるノードの特徴も用いて学習されてしまう可能性があり、間違った予測をしてしまう可能性がある。本論文の提案手法でも周辺ノードの情報を用いてノード分類を行う点は類似している。しかし、隣接ノードのみの情報を用いるのではなく、ノードを分類するために必要なノードの情報を選択的に探索し活用するためいずれの手法とも異なる。説明性という観点では、Veličković らが提案した GAT は隣接ノードの重みを表すアテンションによって各隣接ノードの重要度を示せる点で類似しているが、本論文の提案手法では隣接するノードに限らず、パスをノードの分類根拠とする点で異なる。

グラフ内の全ノードについて計算する必要がある限界を解決するためには、分類に用いるノードを選択的に探索する方法が考えられ、これを実現できる方法としては強化学習が挙げられる。強化学習とは、結果の望ましさを元に報酬と呼ばれる値を受け取り、試行錯誤した結果得られた報酬を元に学習する方法である。強化学習をグラフへ適用した研究も存在する。Xian らは強化学習を知識グラフに適用することで、パスを根拠とする説明可能な推薦手法を実現している [13]。Zhou らは強化学習を知識グラフに適用したインタラクティブ推薦システムを提案している [14]。このシステムはエージェントが探索するグラフとして知識グラフを用いることにより、学習を効率化させつつコールドスタート問題に有効な手法となっている。Lee らはグ

ラフラベルの分類において、グラフ内の核となる情報がある部分に注目することで、分類の予測を妨げる情報を含まないようにする GAM (Graph Attention Model) を提案している [15]. GAM の実現にあたっては強化学習を用いており、エージェントに各ノードを起点にグラフを探索させている. エージェントがたどったノードの系列をリカレントニューラルネットワークに逐次的に入力し、出力となる得られたベクトルを足し合わせたベクトルを元にグラフを分類する. この手法は強化学習のエージェントにグラフを探索させることで分類根拠にノイズが入らないようにする点や、グラフ内の分類に有用なパスを根拠に分類を行っている点が本論文と類似している. その一方で、Xian らや Zhou らが提案した手法は推薦のための手法であり、Lee らが提案した GAM もグラフ分類を行うモデルであってノード分類には適用できない点で本論文と異なる. また、Akujuobi らも GAM をノード分類へ適用できるようにしたモデルを提案しているが、半教師あり学習で学習している点や、説明性を求めている点で本論文と異なる [16].

例えば論文と論文の著者のグラフのように、ノードの種類が 2 種類以上あるグラフを異種グラフと呼ぶが、異種グラフにおいてパスを根拠にノード分類を行う方法として、メタパスを用いた手法も提案されている. メタパスとはパスを抽象化したパスであり、例えば論文と論文の著者のグラフでは、(著者, 論文, 著者) のメタパスによって論文の共著関係を表現できる. このようなメタパスを予め決め、モデルに組み込むことでグラフ内に存在する関係性を捉えられる. Dong らは指定したメタパスにしたがってランダムウォークを行い、その結果得たノードの埋め込みを入力としてノードを分類する手法を提案した [17]. Fu らはアテンションを用いることで複数のメタパスを組み込めるノード分類手法を提案した [18]. Wang らは GAT を異種グラフに適用できるように拡張した手法を提案した [19]. いずれの手法もグラフ内の関係性をパスで捉えられる点では本論文と類似する. しかし、メタパスを用いた手法では人手でメタパスを決める必要がある一方、本研究のような強化学習のエージェントにグラフを探索させる手法では分類根拠となるパスを探索的に発見できる点で異なる.

3 提案手法

本節では、まずノード分類タスクの問題設定を示す. その後、提案手法を 2 種類の人工データへ適用する方法についてそれぞれ述べる.

3.1 問題設定

ノード分類タスクの問題設定は、あるグラフ $G = (V, E)$ が与えられたとき、各ノード $v \in V$ を入力とし、出力として v のラベル l を予測する問題を解くことを意味する. ここで V は G のノード集合を表し、 E は G のエッジ集合を表す. 例えば SNS のグラフにおいて、ユーザがスパムアカウントであるかどうかを予測するノード分類タスクを想定する. この場合、ユーザが入力となり、出力としてそのユーザがスパムアカウントで

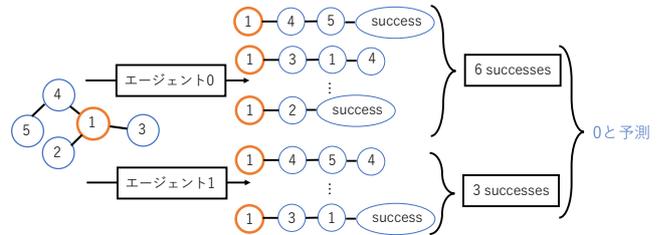


図 2 提案手法の概略図

あるかどうかを得ることになる.

3.2 深層強化学習に基づくノード分類

本手法では、 k クラス分類問題において、エージェント l がラベル l を予測するように行動させることで、そのラベルに分類するにあたって妥当な根拠となるパスをエージェントが判断可能とする. エージェント l は分類対象とするノードを起点とし、ラベルがクラス l に属する可能性が高いと判断した場合に成功という特殊な判断を下す. これに基づき、エージェント l はノードラベルが l であるときに成功と判断する確率が高くなり、ノードラベルが l 以外のときに成功と判断する確率が低くなるように学習させる. 図 2 にモデルの概略図を示す. この図ではノード 1 のラベルを予測する場合の例を示している. k 個のエージェントすべてについて、隣接するノードへ遷移または成功と判断できる最大回数を c (≥ 1) としこれをステップ数と呼ぶ. 図 2 では、エージェントが t 回遷移した後、 $t+1$ 回目に遷移する状況を示している. エージェントが成功と判断した場合およびステップ数が c 回に達した時点で終了とする. エージェントは分類したいノード v_0 を起点として、 v_0 のラベルを予測する上で有用な情報を得べくグラフ上を探索する. したがって、分類するノード v_0 および v_0 を起点にエージェントがたどったノード列 $\{v_0, \dots, v_t\}$ が入力となり、次に遷移するノードおよび成功と判断する確率を出力として得る. 様々なパスをたどることでより分類根拠として適したパスを発見できる. そのため、以上の試行を各エージェントについて d 回繰り返す. その後、 d 回試行した結果得た各エージェントの成功の回数を計算する. エージェント l の成功回数を u_l とすれば、各エージェントの成功回数は $\mathbf{u} = (u_1, \dots, u_l, \dots, u_k)$ と表せる. \mathbf{u} を入力としてノードラベルの予測結果を出力とし、本論文では $\arg \max_l u_l$ を予測ラベルとする. なお、GAT を提案した Veličković らも実験に用いている PPI (Protein-Protein Interaction) dataset など、観測可能なノードラベルやエッジラベルを持たないグラフも存在する [7]. 本論文では、ノードラベルとエッジラベルを持つグラフだけでなく、このようなグラフへの適用も想定して手法を提案する.

行動: 強化学習における行動とは、エージェントが現在の状態において次に行える行動のことを示す. 本手法においては、エージェントは現在いるノード v_t から、隣接するあるノード $v \in \mathcal{N}(v_t)$ への遷移あるいは成功のいずれか 1 つの行動を採用する. なお、ここで隣接するノードの集合を $\mathcal{N}(v)$ とする. 成功を採用した場合、エージェントの遷移は終了し、次の繰り返し

しに進む。

報酬： 報酬とは結果の望ましさを表す値であり、この値を元にエージェントの学習を行う。本論文では t 回目の遷移でエージェントが得る報酬を r_t と表す。強化学習では学習が安定しにくいいため、本手法では、報酬を $r_t \in [-1, 1]$ の範囲にクリッピングすることで学習が安定しやすくなるようにする。報酬は次のように与える。エージェント l について開始ノードのラベルが l だった場合には、成功という行動を採用する回数が多い方が望ましい。そのため、成功を採用すると $r_t = 1$ 、成功せずに規定回数行動した場合には $r_t = -1$ 、それ以外の場合には $r_t = 0$ とする。開始ノードのラベルが l 以外だった場合には、成功を採用する回数が少ない方が望ましい。そのため、成功を採用すると $r_t = -1$ 、成功以外の行動を採用した場合には引き続き成功以外の行動を採用することが望ましいため $r_t = 1/c$ の報酬を与える。 $1/c$ ずつ報酬を与えることで、一度も成功を採用せずに c 回遷移した場合に合計 1 の報酬を得られることになる。このように報酬を設計することで、エージェント l がラベル l を担当するように探索しないと報酬をもらえなくなるため、各エージェントが根拠となるパスをたどるように動作させる。

状態： 状態とはある時刻における、対象の様子を表現したもののことである。これは、状態が強化学習の入力となる値であることを意味している。本論文の実験では、現在たどり着いているノードの観測可能なノードラベルを示す one-hot ベクトルと、これまで観測可能な各ノードラベルをたどってきた回数を表すベクトルを連結したものを状態とする。観測可能なノードラベルを持たないグラフに適用する際には、たどってきたノード列が状態となる。

パラメータの更新： 本論文では行動価値関数を用いたオソドックスな強化学習を採用する。行動価値関数とはある状態および行動において、その後理想的な行動を採用した際に将来的に得られることが期待される割引報酬和を値とする関数である。この行動価値関数を学習することで最適な行動が採用できるようになる。パラメータの更新には代表的な学習方法である Q 学習を用いる。Q 学習は次の式で表される：

$$Q(s_t, a_t) = Q(s_t, a_t) + \eta \cdot (r_{t+1} + \gamma \max_a Q(s_{t+1}, a_t) - Q(s_t, a_t)) \quad (1)$$

$Q(s_t, a_t)$ は t 回目の遷移における状態 s_t と行動 a_t における行動価値を表す。 η は学習率を示し、 γ は将来的な報酬をどれだけ割り引くかを表す割引率を示す。行動価値関数を更新する際には $Q(s_t, a_t) = r_{t+1} + \gamma \max_a Q(s_{t+1}, a_t)$ が成り立っていると理想的であるため、 $E(s_t, a_t) = r_{t+1} + \gamma \max_a Q(s_{t+1}, a_t) - Q(s_t, a_t)$ とすると、損失関数は次のように定義できる：

$$L = \begin{cases} \{E(s_t, a_t)\}^2 & (-1 \leq E(s_t, a_t) \leq 1) \\ |E(s_t, a_t)| & (\text{otherwise}) \end{cases} \quad (2)$$

ここでは誤差関数の出力値が大きくなり、学習が安定しなくなることを防ぐため Huber Loss を用いている。

遷移確率の計算： 遷移確率の計算にあたり、最初に行動の採用方法について示す。行動の採用にあたって、エージェント

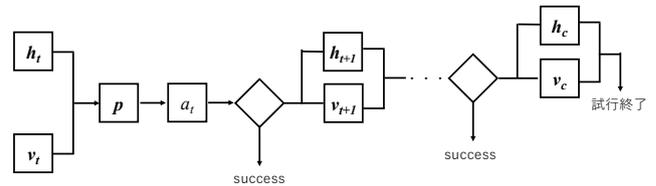


図 3 エージェントの遷移過程

がまだ十分に学習されていないときは様々なパスを探索すべきで、十分学習された段階では、今までの学習結果を活用すべきである。これを実現するため、採用する行動は ϵ -greedy 法の ϵ の値を減衰させたものに基づいて決定する。本手法では、 $\epsilon = 0.5 / (\text{episode} + 1)$ とし、確率 ϵ で採用できる行動の中からランダムに行動を決定し、これを探索と呼ぶ。確率 $1 - \epsilon$ で、採用できる行動の中で最も得られる報酬が多くなると期待される行動を確率的に採用し、これを活用と呼ぶ。なお、1 回のノードラベルの予測を 1 episode と呼ぶ。ノードラベルを持つグラフについてはノードラベルの選択が行動となり、そのラベルの中からランダムに 1 つのノードへ遷移する。ノードラベルを持たないグラフについては、隣接する各ノード自体への遷移確率をもとに遷移する。

次に、探索の場合と活用の場合に分けて遷移確率の計算方法を示す。まず、探索の場合における遷移確率の計算方法を示す。ノードに観測可能なラベルがある場合には、隣接するノードラベル及び成功のいずれかの行動から、無作為に一つの行動を採用する。ノードに観測可能なラベルを持たない場合については成功を採用する確率を別途決める。これは、隣接ノード数が多い場合に、無作為に行動を採用するのみでは成功を十分に採用しないまま学習が進むおそれがあるからである。そのため、ノードラベルを持たない場合には確率 0.5 で成功を採用し、残りの確率で無作為に隣接ノードを選択して遷移するようにした。

次に、活用の場合について、本論文の実験における遷移確率の計算方法を示す。エージェントは線形層のみのニューラルネットワークによって計算された遷移確率に基づいて行動を決定する。Mnih らは Q 学習において、強化学習の入力である状態をニューラルネットワークを用いて計算し、出力に行動を得る手法、DQN (Deep Q-Network) を提案している [20]。これによって、本論文における遷移確率の計算部分は柔軟に構成できることが示されている。しかし本論文では解釈性のあるモデルを構築することが目的であるため、線形層のみによるモデルを用いて実験を行う。

図 3 にエージェントの遷移過程を表した図を示す。 t 回目に遷移したノードが持つ、観測可能なノードラベルを示す one-hot ベクトルを v_t 、これまで観測可能な各ノードラベルをたどってきた回数を表すベクトルを h_t 、表現行列を S_1, S_2 で表す。なお、各ノードラベルにおいて、エージェントは各エッジラベルに加え、成功という行動も採用できるため、 S_1, S_2 はエッジのラベル数を e 、ノードのラベル数を f としていずれも $S \in \mathbb{R}^{(e+1) \times f}$ とする。 h_t については、例えば観測可能なノードラベルが 3 種類あり、ラベル 0 を 2 回、ラベル 1 を 1 回、ラベル 2 を 3 回たどったとする。このとき、 $h_t = (2, 1, 3)$ と表せる。現在および

これまでの遷移で得たノードラベルを用いて次に採用する行動を決定するため、式 (5) によって、 \mathbf{v}_t と \mathbf{h}_t を入力とした確率的な行動の採用方法を実現する。

$$\mathbf{v}'_t = \mathbf{S}_1 \mathbf{v}_t \quad (3)$$

$$\mathbf{h}'_t = \mathbf{S}_2 \mathbf{h}_t \quad (4)$$

$$\mathbf{p} = \text{softmax}(\mathbf{v}'_t + \mathbf{h}'_t) \quad (5)$$

成功を採用する確率は次のように表せる：

$$\mathbf{p}(\text{success}) = p_{e+1} = \mathbf{s}_{1e+1} \cdot \mathbf{v}_t + \mathbf{s}_{2e+1} \cdot \mathbf{h}_t \quad (6)$$

ただし、 \mathbf{s}_{ie+1} は \mathbf{S}_i の $e+1$ 行目を表す行ベクトルである。

ノードに観測可能なラベルを持たない場合のモデルについては、ノードの特徴量を用いて遷移確率を計算する。 t 回目に遷移したノードの特徴量ベクトルを \mathbf{v}_t 、表現行列を \mathbf{M} 、隣接するあるノードの特徴量ベクトルを \mathbf{n} とする。 t 回目に遷移したノードの特徴量とそのノードに隣接する各ノードの特徴量の組み合わせによって遷移確率を計算するため、特徴量ベクトル \mathbf{n} を持つノードへの遷移確率 q_i を次のように計算する：

$$q_i = \mathbf{v}_t^T \mathbf{M} \mathbf{n} \quad (7)$$

開始ノードに加え、これまでの遷移でたどってきたノードに基づいて成功とする確率を得るため、成功とする確率 q' は次のように計算する：

$$\mathbf{v} = \text{cat} \left(\mathbf{v}_0, \frac{1}{t+1} \sum_{j=0}^t \mathbf{v}_j \right) \quad (8)$$

$$q' = \text{sigmoid}(\mathbf{m}^T \mathbf{v}) \quad (9)$$

ここで \mathbf{m} は線形変換を表すベクトルで、 $\text{cat}(\mathbf{a}, \mathbf{b})$ は \mathbf{a} へ後ろから \mathbf{b} を連結したことを示す。つまり、開始ノードの特徴量と今までたどってきたノードの特徴量の平均を連結したベクトル \mathbf{v} を入力として成功とする確率を予測する。 q' は確率であり、 $q' \in [0, 1]$ を満たす必要があるためシグモイド関数を用いて変換する。隣接する各ノードについて q_i を計算して得た遷移確率ベクトル \mathbf{q} と q' から、次の式で得られるベクトルの各値をもとに、確率的に次に採用する行動を決定する。

$$\mathbf{p} = \text{cat}((1 - q') \text{softmax}(\mathbf{q}), q') \quad (10)$$

4 実験

本論文では多様な形式のデータに対して分類根拠の可視化を行うにあたり、観測できないノードラベルを予測する 2 値分類タスクを解く。このタスクを解くにあたり、本論文では人工データを用いた実験を行ったため、まず作成した人工データの概要を説明する。その後、本論文で提案する手法での実験設定と実験結果を示す。

4.1 データセット

次の 2 種類の人工データを作成し実験に用いた：(1) ノードが観測可能なノードラベルとエッジラベルを持つ特徴量は持

表 1 人工データの概要

	データ 1	データ 2
ノード	2,654	1,759
エッジ	5,308	3,518
観測可能なノードラベル	4	1
エッジラベル	5	1
特徴量	持たない	各ノード 3 次元

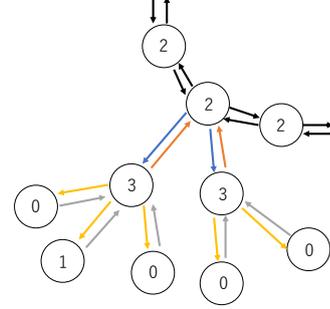


図 4 データ 1 のグラフ例

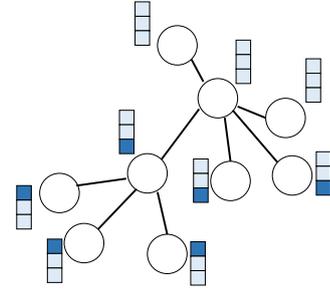


図 5 データ 2 のグラフ例

たないデータ (2) ノードが特徴量を持つ観測可能なノードラベルやエッジラベルを持たないデータ。表 1 に各データの統計情報を示す。いずれも観測できないノードラベルを持つノードを 100 個持ち、このラベルを予測するのが実験内容となる。

(1) のデータ (以下、データ 1) では、観測できないノードラベルは特定のパスに強く依存して決定されるように作成した。データ 1 のグラフの例を図 4 に示す。データ 1 では、観測できないノードラベルはノードラベルを (2, 3) とたどった先にある観測可能なノードラベルに依存している。このため、ノードラベルを (2, 3) とたどった後に、 $i \in \{0, 1\}$ として、ノードラベル i へ遷移した場合には予測したいラベルは i である可能性が高くなるように作成されているといえる。なお、エッジラベルは観測可能なノードラベル間の接続関係を表している。これは例えばある SNS を表す異種グラフにおいて、ユーザはユーザをフォローするか、投稿をシェアか「いいね」ができるとする。この SNS の場合、システム上ユーザが投稿をフォローすることはできない。このような関係性を表現するためにエッジラベルを用いている。(2) のデータ (以下、データ 2) では、特定のパスをたどった後に遷移できるノード v に依存して観測できないノードラベルが付与されている。データ 2 のグラフの例を図 5 に示す。データ 2 では $i \in \{0, 1\}$ として観測できないラベルが i であった場合、 v の特徴量ベクトルの i 次元目が高くなる

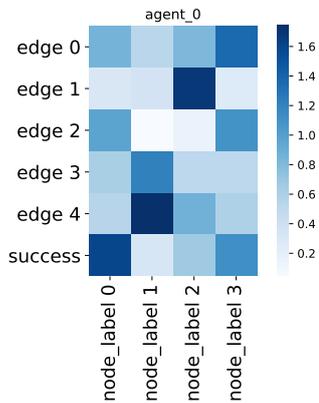


図 6 データ 1 におけるエージェント 0 のパラメータ

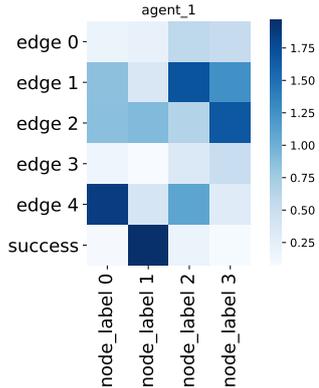


図 7 データ 1 におけるエージェント 1 のパラメータ

ように作成されている。このように、特徴量のある次元がデータ 1 のエッジラベルに対応しているため、エージェントは特定の特徴量を持つノードへ遷移する行動を取り続けた後に成功とするか、遷移を続けるかを選択する戦略が最適となる。

4.2 実験設定

実験で採用したパラメータを示す。ユーザが分類根拠を理解できるモデルとするため、ステップ数は 3 とした。本実験では 2 値分類を解くため、エージェント数は 2 となる。学習率は 0.01 とし、Akujuobi らの実験と同様に、バッチサイズは 128、エポック数は 30 とした [16]。また、パラメータの更新にあたっては、近い時間の内容から連続的に学習するとパラメータを十分に学習しにくい問題がある。そこで本実験にあたっては Lin が提案している Experience Replay を用いることで、複数のステップから学習を進められるようにした [21]。なお、Experience Replay のメモリサイズは 1024 とした。Q 学習にあたっては、提案手法ではステップ数に応じた時間減衰させる必要はないので $\gamma = 1.0$ とした。実験にあたっては、観測できないノードラベルを持つノード 100 個のラベルを予測するが、100 個のノードのうち 80 個を訓練用データとし、20 個をテストデータとして用いた。このことから、本実験においては 80 episode が 1 epoch と表される。

4.3 実験結果

データ 1 について、実験で得られたパラメータのヒートマップ

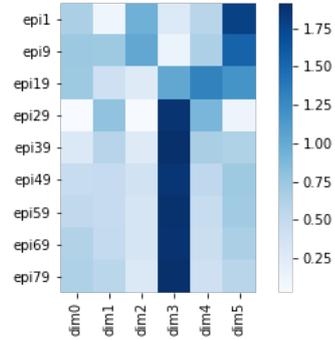


図 8 データ 2 におけるエージェント 0 のパラメータ

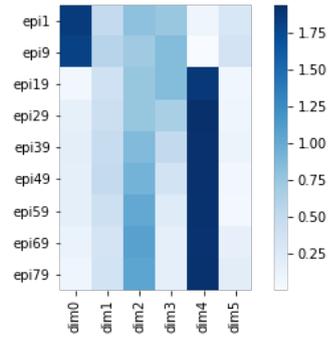


図 9 データ 2 におけるエージェント 1 のパラメータ

を 図 6 と 図 7 に示す。これらの図におけるパラメータとは、式 (3), (4) の $S_1 + S_2$ の各成分の値のことを意味している。図 6 はエージェント 0 が観測できないノードラベル 0 を予測するための学習で得たパラメータを示し、図 7 はエージェント 1 について観測できないノードラベル 1 を予測するための学習で得たパラメータを表したものである。いずれも図の行がエッジラベルおよび成功を表しており、列がノードラベルを表している。各セルがパラメータであり、色が濃いほどそのノードラベルにおいてそのエッジラベルを選択しやすいことがわかる。ノードラベルは状態に対応し、エッジラベルは行動に対応するため、入力である各状態に対して、どの出力つまり行動を採用しやすいかが表されている。なお、比較のため、ヒートマップを描画する際には各列について標準化を行っている。図 6 より、エージェント 0 はノードラベルが 0 のときに成功、ノードラベルが 1 のときにエッジ 4、ノードラベルが 2 のときにエッジ 1、ノードラベルが 3 のときにエッジ 2 を採用しやすい。このため、エージェント 0 はノードラベルを (2, 3) とたどる可能性が高く、ノードラベルが 0 であれば成功とし、ノードラベルが 1 であれば遷移を続けている可能性が高い。したがって、エージェント 0 は最適な行動を学習できていると考えられる。図 7 より、エージェント 1 はノードラベルが 0 のときにエッジ 4、ノードラベルが 1 のときに成功、ノードラベルが 2 のときにエッジ 1、ノードラベルが 3 のときにエッジ 2 を採用しやすい。エージェント 1 もノードラベルを (2, 3) とたどる可能性が高い。そして、ノードラベルが 1 であれば成功とし、ノードラベルが 0 であれば遷移を続けている可能性が高いため、エージェント 0 同様に最適な行動を学習できていると考えられる。

データ 2 について、実験で得られたパラメータの一部をヒ-

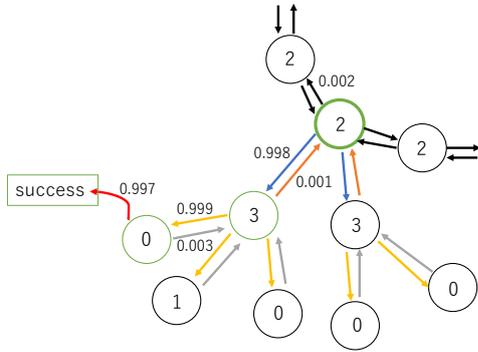


図 10 データ 1 におけるエージェント 0 の遷移事例

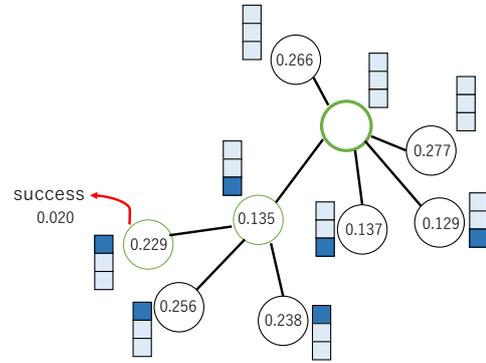


図 11 データ 2 におけるエージェント 1 の遷移事例

トマップで表した図を図 8 と図 9 に示す。ここでは式 (9) の \mathbf{m} をパラメータとして示した。図 8 はエージェント 0 がノードラベル 0 を予測するための学習で得たパラメータを示し、図 9 はエージェント 1 について同様のパラメータを表したものである。いずれの図も行が episode 数を表しており、列がパラメータ \mathbf{m} の次元を表している。つまり、各行が \mathbf{m}^T に相当し、学習が進むにつれて下の行の値を取るように示されている。各セルが \mathbf{m} の各成分の値を示しており、色が濃いほどその次元のパラメータにかかる特徴量の重要度が高い。式 (9) より、 \mathbf{m} の i 次元目が \mathbf{v} の i 次元目にかかるため、これらの図は \mathbf{v} の各次元の重要度を表しているといえる。なお、比較のため、ヒートマップを描画する際には各行について標準化を行っている。図 8 より、 \mathbf{v} の 3 次元目が成功するかどうかに関わっている。このことから、エージェント 0 はこれまで遷移してきた全ノードの特徴量の 0 次元目が成功に強く関わっているといえる。したがって、エージェント 0 は最適な行動を学習できていると考えられる。図 9 より、 \mathbf{v} の 4 次元目が成功するかどうかに関わっている。このことから、エージェント 1 はこれまで遷移してきた全ノードの特徴量の 1 次元目が成功に強く関わっているといえる。したがって、エージェント 1 は最適な行動を学習できていると考えられる。

次に、各データでの実験におけるエージェントの遷移事例を示す。図 10 はデータ 1 におけるエージェント 0 の遷移事例である。図 10 では、観測できないノードラベルを持つノードを開始ノードとして、エージェントが分類根拠となるパスを探索した事例を示している。この図では各エッジラベルをエッジの色で表現し、エージェントがたどったノードを緑色で示している。エッジに付記されている数値はそのエッジラベルの採用確率を表している。図中の“success”は成功を表しており、“success”へ遷移した場合エージェントはその試行での探索を終了する。図 10 の遷移事例では、エージェントはノードラベルを (2, 3, 0) とたどった後に成功と判断している。エッジラベルの採用確率については、エッジラベルは青色、黄色を採用する確率が非常に高い。これは本データにおいてノードラベルを (2, 3) とたどりやすいことを意味している。この遷移事例においてもエージェントはノードラベルを (2, 3) とたどった後に遷移できるノードの観測可能なラベルをもとに、成功とするかどうかを判断していると考えられる。したがって、このエージェントは想

定通りの遷移ができていると推測できる。このように、パスによって分類根拠を可視化できていることから、隣接ノードの重みのみを示せる GAT などの従来手法とは異なる形で分類根拠を提示できたといえる。図 10 で示した事例では、黄色のエッジラベルを採用した際にはノードラベルが 0 のノードへ遷移する可能性が高いが、ノードラベルが 1 のノードへ遷移する可能性もある。この場合の事例も調べた結果、エージェント 0 は予測したいノードラベルは 0 であるにも関わらず成功と判断していなかった。このことから、エージェントが自身のたどったノードラベルに依存して行動を決めていることが推測できる。

図 11 はデータ 2 におけるエージェント 1 の遷移事例である。図 11 も同様に、観測できないノードラベルを持つノードを開始ノードとして、エージェントが分類根拠となるパスを探索した事例を示している。この図では各ノードが持つ特徴量の各成分を色の濃さで表現し、エージェントがたどったノードを緑色で示している。例えば 3 次元ベクトルにおいて、特徴量の 0 次元目の値が大きく、他の次元の値は小さい場合には、1 番上の正方形の色を濃く、ほかの正方形の色を薄くして表現している。ノードに書かれている数値は、エージェントがたどった 1 つ前のノードからそのノードへの遷移確率を表している。図中の“success”は成功を表しており、“success”へ遷移した場合エージェントはその試行での探索を終了する。図 11 の遷移事例では、エージェントは特徴量の各次元の値が小さいノードを開始ノードとして、(開始ノード、2 次元目の値が高いノード、0 次元目の値が高いノード、2 次元目の値が高いノード) とたどっており、成功とは判断せずに規定ステップ数に到達している。各ノードへの遷移確率については、殊更に遷移確率が高いノードは存在せず、どの隣接ノードについてもある程度の確率で遷移することがわかる。図 11 より、エージェントはたどってきたノードの特徴量の 1 次元目を重要視する。したがって、この遷移事例においてエージェントは (開始ノード、2 次元目の値が高いノード、0 次元目の値が高いノード) とたどった結果、特徴量の 1 次元目の平均が低く、分類根拠とするにふさわしいノードが発見できたと判断しなかったと考えられる。これはこの試行では分類根拠となるパスが見つけれず、成功と判断しなかったことを意味している。遷移確率については適切に学習できているとは言い難いが、本データにおいてふさわしい分類根拠となるパスを探索および可視化できている。したがって、データ

1と同様、従来手法とは異なる形で分類根拠を提示することができたといえる。

5 ま と め

本論文では、分類に用いるノードを探索でき、かつ、どのようなノードに基づいてノードを分類するかを制御可能とする手法を提案した。この手法ではグラフ上のパスをノード分類の根拠として用いることで、ノード分類を行う上で核となる情報を持つ部分だけを捉えることができ、かつ、多様な根拠に基づくノードラベルの分類を可能とした。パスを根拠とした分類には強化学習を用い、エージェントにグラフ上のパスを探索させることによって、分類根拠としてふさわしいパスを学習した。実験では次の2種類の人工データを作成し実験に用いた：(1) ノードが観測可能なノードラベルとエッジラベルを持つが特徴量は持たないデータ (2) ノードが特徴量を持つが観測可能なノードラベルやエッジラベルを持たないデータ。本論文の実験においては、これらの2種類のデータについて、観測できないノードラベルを予測するタスクに取り組んだ。実験の結果、既存手法とは異なる根拠に基づいて分類が行えることを確認することができた。この論文における貢献は以下の2つである：(i) ノード分類タスクにおいて、分類根拠をパスに基づいて提示する手法を提案した。(ii) 人工データを用いた実験を行い、提案手法が従来手法とは異なる分類および可視化ができることを示した。

今後の課題としては、提案手法の実データへの適用や、ノードの特徴量とラベルが両方とも存在するモデルの提案、分類精度の向上などを挙げる。

謝辞 本研究は JSPS 科研費 18H03243 の助成を受けたものです。ここに記して謝意を表します。

文 献

- [1] 総務省情報流通行政局情報通信政策課情報通信経済室。ICTによるインクルージョンの実現に関する調査研究。 https://www.soumu.go.jp/johotsusintokei/linkdata/h30_03_houkoku.pdf, (accessed 2020-07-15).
- [2] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [3] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.
- [4] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077, 2015.
- [5] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016.
- [6] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.
- [7] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [8] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. GRAM: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 787–795, 2017.
- [9] Will Hamilton, Zhitaoying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.
- [10] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983, 2018.
- [11] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.
- [12] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*, 2019.
- [13] Yikun Xian, Zuohui Fu, S Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 285–294, 2019.
- [14] Sijin Zhou, Xinyi Dai, Haokun Chen, Weinan Zhang, Kan Ren, Ruiming Tang, Xiuqiang He, and Yong Yu. Interactive recommender system via knowledge graph-enhanced reinforcement learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 179–188, 2020.
- [15] John Boaz Lee, Ryan Rossi, and Xiangnan Kong. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1666–1674, 2018.
- [16] Uchenna Akujuobi, Qiannan Zhang, Han Yufei, and Xiangliang Zhang. Recurrent attention walk for semi-supervised classification. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 16–24, 2020.
- [17] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 135–144, 2017.
- [18] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*, pages 2331–2341, 2020.
- [19] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032, 2019.
- [20] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fiedjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- [21] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321, 1992.