人工知能学会研究会資料 SIG-SWO-049-04

RDF グラフから抽出した多様な特徴ベクトルによる教師あり学習

Supervised Learning for Various Feature Vectors from RDF Graphs

南 陽太 [1*]　兼岩 憲 [2]
Yota Minami[1]　Ken Kaneiwa[2]

[1] 電気通信大学 情報理工学域 I 類 コンピュータサイエンスプログラム
[1] Department of Communication Engineering and Informatics,
Faculty of Informatics and Engineering, The University of Electro-Communications
[2] 電気通信大学 大学院情報理工学研究科 情報・ネットワーク工学専攻
[2] Department of Computer and Network Engineering,
Graduate School of Informatics and Engineering, The University of Electro-Communications

Abstract: RDF グラフはリソース間の関係によって結ばれたグラフデータであり，LOD(Linked Open Data) として Web 上に公開されている．この RDF グラフから抽出した特徴を用いて SVM(Support Vector Machine) などの機械学習が盛んに研究されている．本研究では，RDF グラフから各リソースの特徴ベクトルを抽出する方法を提案し，その特徴ベクトルを深層ニューラルネットワークへ適用する．評価実験では，Wikidata，DBpedia，YAGO などの RDF データを用いたリソースのクラス分類において高い正解率を示す．

1　はじめに

Resource Description Framework (RDF) は，Web 上でリソースの属性や関係を記述する枠組みである．セマンティック Web では，このようなリソースの意味的な記述により Web 上でデータの機械可読性を高めようとしている．RDF データは，Linked Open Data (LOD) として Web 上に公開されており，このデータを用いた機械学習 [2][5][7][8] やデータマイニング [3] の研究が，近年盛んに行われている．

大量のデータを活用するために，k 近傍法，決定木学習，SVM など，機械学習の手法が用いられている．その中でも，ニューラルネットワークは多次元量のデータで線形分離不可能な問題にも対応できることから，注目を浴びている．特に，画像，自然言語文などの分類問題に対して高い精度を実現する深層ニューラルネットワークが広く応用されている．

しかし，Web や SNS などのグラフ構造のデータを機械学習に適用させるには，様々な工夫が必要になる [9]．特に，RDF データはメタデータ，オントロジー，属性データなど異なるタイプの意味的リンクがグラフ構造内に埋め込まれている．本研究では，RDF グラフから各リソースの属性や関係性を捉えた多様な特徴ベクトルを抽出する手法を提案する．学習対象のリソースを示すノード（主語）から，近傍のエッジ（述語）やノード（目的語）を組み合わせてリソースの特徴を抽出する．このとき，特徴ベクトルの次元数が大きくなりすぎないよう，特徴毎の情報利得率を基準にして特徴ベクトルの次元削減を行う．さらに，RDF グラフ上のリソースを様々な特徴ベクトルで表現して深層ニューラルネットワークに適用する．その結果，RDF のようなグラフ構造上のデータを用いて，深層学習によって高い精度でリソースの分類問題を学習できる．実際に，Wikidata，DBpedia，YAGO などの RDF データを用いて評価実験を行い，クラス分類における高い正解率を示す．

本稿の構成は，以下の通りである．2 章では，本研究に関連する概念である RDF グラフ，情報利得率，深層ニューラルネットワークの概要を述べる．3 章では，RDF グラフからリソースの特徴集合と特徴ベクトルを抽出する手法について述べる．4 章では，抽出した特徴ベクトルを使用して深層ニューラルネットワークの評価実験を行う．最後に，5 章で本論文の結論と今後の課題を述べる．

*連絡先： 電気通信大学 情報理工学域 I 類 コンピュータサイエンスプログラム
〒 182-8585 東京都調布市調布ヶ丘 1-5-1
E-mail: minami@sw.cei.uec.ac.jp

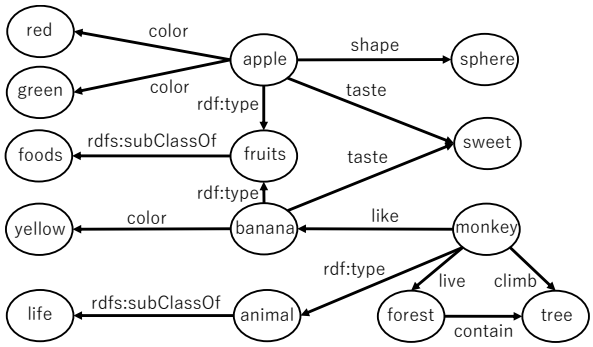

図 1: RDF グラフの例

2 準備

2.1 RDF グラフ

RDF はリソースを記述する枠組みであり，リソース間の関係を主語，述語，目的語の三つ組（RDF トリプル）によって表現する．RDF トリプルは，URI 参照の集合 $\boldsymbol{U}$ ，空ノードの集合 $\boldsymbol{B}$ ，およびリテラルの集合 $\boldsymbol{L}$ から構成される．RDF トリプルを (s, p, o) とすると，主語 s は $\boldsymbol{U} \cup \boldsymbol{B}$ の要素，述語 p は $\boldsymbol{U}$ の要素，目的語 o は $\boldsymbol{U} \cup \boldsymbol{B} \cup \boldsymbol{L}$ の要素である．すなわち，RDF トリプルは $(\boldsymbol{U} \cup \boldsymbol{B}) \times \boldsymbol{U} \times (\boldsymbol{U} \cup \boldsymbol{B} \cup \boldsymbol{L})$ の要素といえる．RDF グラフ G は，RDF トリプルの集合 $\{(s_1, p_1, o_1), \cdots, (s_n, p_n, o_n)\}$ として定義される [4].

例えば，フルーツや動物に関する RDF グラフを，図 1 に表す．

2.2 情報利得率

ある RDF グラフ G を与えたとき，その中で学習対象となる全リソース集合を $\boldsymbol{R}_G$ とする．学習対象となる全リソースの集合 $\boldsymbol{R}_G$ に出現する正例と負例のリソース集合をそれぞれ $\boldsymbol{R}_G^+$，$\boldsymbol{R}_G^-$ とする．また，RDF グラフ G の全特徴の集合を F として，特徴 $f \in F$ を所持するリソースを $\boldsymbol{R}_G^f$，所持しないリソースを $\boldsymbol{R}_G^{-f}$ とする．

荒井ら [1] によると，RDF グラフ G に対して，データの乱雑さを表す情報量 $Info(\boldsymbol{R}_G)$，特徴 f により分割したデータに対する情報量 $Info_f(\boldsymbol{R}_G)$ は，以下のように定義される．

$$
\begin{aligned}
Info(\boldsymbol{R}_G) &= -\sum_{c \in \{+,-\}} \frac{|\boldsymbol{R}_G^c|}{|\boldsymbol{R}_G|} \log_2 \frac{|\boldsymbol{R}_G^c|}{|\boldsymbol{R}_G|} \\
Info_f(\boldsymbol{R}_G) &= \sum_{x \in \{f,-f\}} \frac{|\boldsymbol{R}_G^x|}{|\boldsymbol{R}_G|} Info(\boldsymbol{R}_G^x)
\end{aligned}
$$

特徴 f をクラスラベルと見立てた情報量である分割情報量 $SplitInfo_f(\boldsymbol{R}_G)$，特徴 f によりデータを分割した場合の情報量の減少を表す情報利得 $IG_{\boldsymbol{R}_G}(f)$，情報利得を正規化した情報利得率 $IGR_{\boldsymbol{R}_G}(f)$ は，以下のように定義される．

$$
\begin{aligned}
SplitInfo_f(\boldsymbol{R}_G) &= -\sum_{x \in \{f,-f\}} \frac{|\boldsymbol{R}_G^x|}{|\boldsymbol{R}_G|} \log_2 \frac{|\boldsymbol{R}_G^x|}{|\boldsymbol{R}_G|} \\
IG_{\boldsymbol{R}_G}(f) &= Info(\boldsymbol{R}_G) - Info_f(\boldsymbol{R}_G) \\
IGR_{\boldsymbol{R}_G}(f) &= \frac{IG_{\boldsymbol{R}_G}(f)}{SplitInfo_f(\boldsymbol{R}_G)}
\end{aligned}
$$

2.3 深層ニューラルネットワーク

ニューラルネットワークは，非線形分離可能な機械学習手法の一種である．多次元ベクトルの訓練データを入力として受け取ると，重みとバイアス，活性化関数を用いて層から層へベクトルを出力していき，最後の層の出力と訓練データのクラスラベルから損失関数を計算する．そして，損失関数が最小になるように最適化手法により重みとバイアスを修正していくことで，学習を行う．

なお，層の数が 4 つ以上のニューラルネットワークを特に深層ニューラルネットワークと呼ぶ．

3 RDF からの特徴ベクトル抽出

本章では，RDF グラフから各リソースに対する特徴のパターンを基準として，その特徴集合からベクトル表現を抽出する．また，抽出した特徴ベクトルに各リソースのクラスラベルを付与してニューラルネットワークに適用させ，新たなリソースに対する未知のクラスラベルの予測を可能とする．

3.1 リソースの特徴集合

学習対象のリソース $s \in \boldsymbol{R}_G$ に対して，9 つの特徴パターン $FP = \{p, o, po, *p, *o, *po, pp, ppo, p*o\}$ を基に抽出された各特徴集合を導入する．この特徴パターンの表現において，o と p は目的語と述語を示し，$*$は，述語と目的語を一つ飛ばすことを意味する．

定義 3.1 (目的語の特徴集合) リソース $s \in \boldsymbol{R}_G$ がもつ目的語の特徴集合は，以下のように定義される．

$$
\begin{aligned}
F_o(s) &= \{o \mid (s, p, o) \in G\} \\
F_{*o}(s) &= \{o' \mid (o, p', o') \in G, o \in F_o(s)\}
\end{aligned}
$$

図 1 の例に対して，目的語の特徴集合は，以下のように表される．

$$
\begin{aligned}
F_o(\text{apple}) &= \{\text{red, green, fruits, sphere, sweet}\} \\
F_{*o}(\text{apple}) &= \{\text{foods}\}
\end{aligned}
$$

定義 3.2 (述語と目的語の特徴集合) リソース $s \in \boldsymbol{R}_G$ がもつ述語と目的語の特徴集合は，以下のように定義される．

$$
\begin{aligned}
F_{po}(s) &= \{(p,o) \mid (s,p,o) \in G\} \\
F_{*po}(s) &= \{(p',o') \mid (o,p',o') \in G, o \in F_o(s)\} \\
F_{ppo}(s) &= \{(p,p',o') \mid (o,p',o') \in G, (p,o) \in F_{po}(s)\} \\
F_{p*o}(s) &= \{(p,o') \mid (o,p',o') \in G, (p,o) \in F_{po}(s)\}
\end{aligned}
$$

図 1 の例に対して，述語と目的語の特徴集合は，以下のように表される．

$$
\begin{aligned}
F_{po}(\text{apple}) &= \{\text{(color,green), (color,red), (shape, sphere), (taste,sweet), (rdf:type, fruits)}\} \\
F_{*po}(\text{apple}) &= \{\text{(rdfs:subClassOf,foods)}\} \\
F_{ppo}(\text{apple}) &= \{\text{(rdf:type,rdfs:subClassOf,foods)}\} \\
F_{p*o}(\text{apple}) &= \{\text{(rdf:type,foods)}\}
\end{aligned}
$$

定義 3.3 (述語の特徴集合) リソース $s \in \boldsymbol{R}_G$ がもつ述語の特徴集合は，以下のように定義される．

$$
\begin{aligned}
F_{p}(s) &= \{p \mid (s,p,o) \in G\} \\
F_{*p}(s) &= \{p' \mid (o,p',o') \in G, o \in F_o(s)\} \\
F_{pp}(s) &= \{(p,p') \mid (o,p',o') \in G, (p,o) \in F_{po}(s)\}
\end{aligned}
$$

図 1 の例に対して，述語の特徴集合は，以下のように表される．

$$
\begin{aligned}
F_{p}(\text{apple}) &= \{\text{color, shape, taste, rdf:type}\} \\
F_{*p}(\text{apple}) &= \{\text{rdfs:subClassOf}\} \\
F_{pp}(\text{apple}) &= \{\text{(rdf:type,rdfs:subClassOf)}\}
\end{aligned}
$$

fp をある特徴パターンとしたとき，すべてのリソース $s \in \boldsymbol{R}_G$ に対する特徴集合を

$$F_{fp}(\boldsymbol{R}_G) = \bigcup_{s \in \boldsymbol{R}_G} F_{fp}(s)$$

とする．

3.2 リソースの特徴ベクトル

特徴ベクトルとは，特徴の集合 $F_{fp}(s)$ をベクトルで表現したものを指す．

定義 3.4 (特徴ベクトル) $F_{fp}(\boldsymbol{R}_G) = \{f_1, f_2, \ldots, f_n\}$ 内の特徴列 $[f_1, f_2, \ldots, f_n]$ が与えられたとする．このとき，リソース $s \in \boldsymbol{R}_G$ の fp による特徴ベクトル $V_{fp}(s)$ は以下のように定義される．

$$V_{fp}(s) = [x_1^{fp}, x_2^{fp}, \ldots, x_n^{fp}]$$

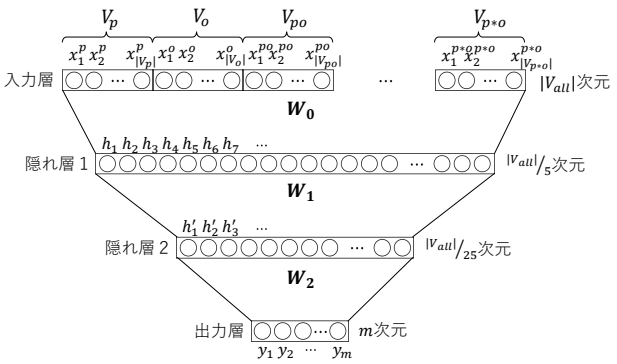

図 2: 特徴ベクトル V_{all} による深層ニューラルネットワーク

但し，

$$x_i^{fp} = \begin{cases} 1 & \text{if } f_i \in F_{fp}(s) \\ 0 & \text{otherwise} \end{cases}$$

すべての $fp \in FP$ による特徴ベクトル $V_{fp}(s)$ の結合を $V_{all}(s)$ とする．特徴ベクトル $V_{fp}(s)$ の次元を $|V_{fp}(s)|$ と表すと，$V_{all}(s)$ の次元は $|V_{all}(s)| = \sum_{fp \in FP} |V_{fp}(s)|$ となる．

図 1 の例において，リソースの集合 $\boldsymbol{R}_G$ を {apple, monkey}，$F_p(\boldsymbol{R}_G)$ の特徴列を [color, shape, rdf:type, like, live, taste, climb] とすると，以下はリソース apple に対する述語の特徴ベクトルである．

$$V_p(\text{apple}) = [1,1,1,0,0,1,0]$$

図 2 は，リソースの m クラス分類問題を学習するために，特徴ベクトルを適用した深層ニューラルネットワークモデルである．このとき，いくつかの特徴パターン fp を選択して，それらの特徴ベクトルを結合する．図 2 では，すべての特徴パターンによる特徴ベクトル V_{all} を $|V_{all}|$ 次元の入力層としている．

3.3 情報利得率による次元の削減

特徴パターンによっては，特徴ベクトルの次元数が大きすぎて学習に現実的ではない時間がかかってしまう．そこで，情報利得率を用いて分類に有用な特徴を判別し，特徴ベクトルの次元を削減する．

定義 3.5 (情報利得率による特徴の削減) 情報利得率のボーダーを $0 < \epsilon \leqq 1$ とした場合，情報利得率によって削減された特徴集合は，以下のように定義される．

$$F_{fp}^{\epsilon}(\boldsymbol{R}_G) = \{f \in F_{fp}(\boldsymbol{R}_G) \mid IGR_{\boldsymbol{R}_G}(f) \geq \epsilon\}$$

表 1: 特徴パターン毎の次元数（ボーダー $\epsilon = 0.15$）

	実験 1	実験 2	実験 3	実験 4	実験 5	実験 6	実験 7	実験 8	実験 9	実験 10
V_p	6	17	31	28	69	20	1	2	2	0
V_o	2	17	17	22	63	101	4	309	40	52
V_{po}	2	16	21	26	50	96	4	314	40	52
V_{*p}	10	71	41	79	157	92	0	13	4	0
V_{*o}	149	1242	699	115	399	689	11	3179	1982	2359
V_{*po}	159	1355	737	130	411	812	11	3362	2003	2522
V_{pp}	66	261	222	251	347	189	6	21	8	0
V_{ppo}	157	1506	1228	164	299	889	14	3802	2080	2583
V_{p*o}	161	1433	1189	166	329	768	14	3695	2058	2445

表 2: 特徴パターン毎の正解率（ボーダー $\epsilon = 0.15$）

	実験 1	実験 2	実験 3	実験 4	実験 5	実験 6	実験 7	実験 8	実験 9	実験 10
V_p	0.6100	0.9100	0.7700	**1.0000**	**0.9900**	0.8250	0.5600	0.7800	0.5350	-
V_o	0.4450	0.8000	0.6800	0.8850	0.8600	0.9500	0.6050	**0.8900**	0.8000	0.5797
V_{po}	0.4450	0.8100	0.7500	0.8900	0.7850	**0.9550**	0.6050	**0.8900**	0.8050	0.5810
V_{*p}	0.5900	0.8700	0.7000	0.8350	0.9200	0.9150	-	0.6600	0.8100	-
V_{*o}	0.8850	0.8800	0.7600	0.8600	0.9150	0.8800	0.6800	**0.8900**	**0.9200**	**0.5824**
V_{*po}	**0.9000**	0.8900	0.7500	0.8900	0.9300	0.8950	0.6800	0.8700	0.9150	0.5661
V_{pp}	0.6500	**0.9300**	0.8100	0.9600	0.9600	0.9100	0.5600	**0.8900**	0.8200	-
V_{ppo}	**0.9000**	0.9200	0.8100	0.9200	0.9200	0.9150	**0.6850**	0.8800	0.9150	0.5819
V_{p*o}	**0.9000**	0.9000	**0.8300**	0.9200	0.9200	0.9100	**0.6850**	0.8800	0.9150	0.5685

ある特徴パターン $fp \in FP$ 内の特徴 $f \in F_{fp}(\boldsymbol{R}_G)$ の中で n 番目に大きい情報利得率を ϵ_n とする．情報利得率によってその特徴パターン $fp \in FP$ 内の上位 n 番目までの特徴集合を $F_{fp}^{\epsilon_n}(\boldsymbol{R}_G)$ と表す．特徴集合 $F_{fp}^{\epsilon}(\boldsymbol{R}_G)$，$F_{fp}^{\epsilon_n}(\boldsymbol{R}_G)$ から作成したリソース $s \in \boldsymbol{R}_G$ の特徴ベクトルを $V_{fp}^{\epsilon}(s)$，$V_{fp}^{\epsilon_n}(s)$ と表す．$V_{all}^{\epsilon}(s)$ はすべての $fp \in FP$ の特徴ベクトル $V_{fp}^{\epsilon}(s)$ の結合である．但し，$V_{all}^{\epsilon_n}(s)$ は各々の特徴パターン $fp \in FP$ における特徴集合 $F_{fp}(\boldsymbol{R}_G)$ 内の特徴 f から n 番目に大きい情報利得率 ϵ_n を用いる．また，リソースから遠い特徴ほどリソース間の類似度に及ぼす影響を少なくする割引係数 λ も導入する．特徴パターンの距離が一つ遠くなる毎に，その特徴パターンの次元数を λ 倍する．例えば，特徴パターン $fp \in \{p, o, po\}$ の次元数が 100，割引係数 λ が 0.3 だった場合，特徴パターン $fp \in \{*p, *o, *po, pp, ppo, p*o\}$ の次元数は 30 となる．

以下は，RDF グラフから情報利得率で次元を削減して特徴ベクトルを作成する手順である．

1. RDF グラフを探索してすべてのリソースに対する特徴集合 $F_{fp}(\boldsymbol{R}_G)$ を生成する．
2. 正例と負例のリソース集合 $\boldsymbol{R}_G^+$，$\boldsymbol{R}_G^-$ から各特徴 $f \in F_{fp}(\boldsymbol{R}_G)$ の情報利得率を計算する．
3. ボーダー ϵ より小さい情報利得率の特徴を削減した集合 $F_{fp}^{\epsilon}(\boldsymbol{R}_G)$ から，各リソース $s \in \boldsymbol{R}_G$ の特徴ベクトル $V_{fp}^{\epsilon}(s)$ を作成する．

4 実験

本研究の有用性を示すために，実 RDF データから各リソースの特徴ベクトルを抽出する．対象リソースに 2 値のクラスラベルを付与し，深層ニューラルネットワークを用いた 2 クラス分類の正解率を評価する．また，RDF カーネルによる SVM の性能と比較するために，荒井ら [2] の実験結果を用いる．

RDF カーネルには，リソースから遠い特徴ほどグラフ間の類似度に及ぼす影響を少なくするために，割引係数が存在する．荒井ら [2] の実験では，Partial SubTree カーネルは $0.0001, 0.001, 0.01, 0.1$ に設定し，その他のカーネルでは 0.1 から 1.0 まで 0.1 刻みで設定してあ

表 3: RDF カーネルによる SVM との比較

	実験 1	実験 2	実験 3	実験 4	実験 5	実験 6	実験 7	実験 8	実験 9	実験 10	平均
vec	0.9450	0.9600	0.8600	1.0000	0.9950	0.9850	0.6700	0.9600	0.9350	0.5567	0.8867
skip	0.8800	**0.9800**	0.8200	**1.0000**	0.9900	**0.9900**	**0.7350**	0.9300	**0.9350**	0.5817	0.8842
hop	0.7550	0.9700	0.8100	**1.0000**	**0.9950**	0.9750	0.7100	0.9200	0.9000	**0.6033**	0.8638
pro	**0.9450**	0.8400	**0.8300**	0.9400	0.9700	0.9650	0.6950	**0.9800**	0.9300	0.5367	0.8632
walk	0.6250	0.7300	0.7900	0.8300	0.7700	0.8750	0.6050	0.8500	0.8950	0.5183	0.7488
path	0.6250	0.7300	0.8000	0.8250	0.7850	0.8650	0.6050	0.8600	0.8950	0.5133	0.7503
full	0.6300	0.7700	0.8100	0.9150	0.8650	0.9200	0.6200	0.8300	0.8950	0.5150	0.7770
partial	0.5950	0.7300	0.7900	0.9200	0.8950	0.9250	0.6050	0.8200	0.8900	0.5417	0.7712

る．ただし，実験 10 の PRO カーネルについては，割引係数を 0.5 に固定している．

以下に示す RDF グラフにおける 10 種類のクラス分類を学習する．実験 1∼3 は Wikidata[1]，実験 4∼6 は日本語 DBpedia[2]，実験 7∼9 は YAGO[3]，実験 10 は LiveJournal[4]の FOAF データを使用する．

実験 1 男女の 2 クラス分類 (284464 トリプル，正例 100，負例 100)
実験 2 海と湖の 2 クラス分類 (177340 トリプル，正例 50，負例 50)
実験 3 純利益の大小による企業の 2 クラス分類 (104523 トリプル，正例 50，負例 50)
実験 4 山と川の 2 クラス分類 (33000 トリプル，正例 100，負例 100)
実験 5 科学者と芸術家の 2 クラス分類 (46822 トリプル，正例 100，負例 100)
実験 6 興行収入の大小による映画の 2 クラス分類 (64521 トリプル，正例 100，負例 100)
実験 7 男女の 2 クラス分類 (40724 トリプル，正例 100，負例 100)
実験 8 GDP の大小による国の 2 クラス分類 (108517 トリプル，正例 50，負例 50)
実験 9 人口密度の大小による場所の 2 クラス分類 (51748 トリプル，正例 100，負例 100)
実験 10 4 つの年代による人の 2 クラス分類 (472965 トリプル，正例 75×4，負例 75×4)

RDF グラフから 9 種類の特徴ベクトルを抽出する処理は，Java 8 により実装している．その際，効率良く RDF グラフを探索するために，SPARQL 検索エンジン FROST 1.1.1[5]を使用する．

次に，各リソース毎の特徴ベクトルを深層ニューラルネットワークに適用して 10 分割交差検証を行い，正解率の平均を計算する．実験 10 は年代を 4 つに分けて各年代クラスを判定する 2 クラス分類を行い，その平均を計算する．深層ニューラルネットワークは Python 3.7 上の Keras ライブラリで実装する．

ニューラルネットワークモデルは，入力層と隠れ層 (1∼4 層) の活性化関数を Relu 関数，出力層の活性化関数を Softmax 関数とする．また，隠れ層の次元は前の層の次元から 5 分の 1 に減らし，出力層は 2 次元とする．ミニバッチのサイズは 32 とし，目的関数には categorical_crossentropy，最適化手法には Adam を用いる．Epoch は 200 とし，教師データの損失関数を監視して 10 エポック以上値が減少しなければ，学習を止める．

実験環境は，OS : Windows 10 Education，CPU : Intel(R) Core(TM) i7-5820K @3.30GHz 3.30GHz，実装メモリ : 64.0GB である．

各特徴パターン $fp \in FP$ の特徴ベクトル V_{fp} と全パターンを結合した特徴ベクトル V_{all} を用いて学習する．その際，情報利得率の固定ボーダー $\epsilon = 0.15$ と各々の特徴パターン $fp \in FP$ 内の上位 n 番目のボーダー ϵ_n と割引係数 $\lambda = 0.3$ によって特徴の数（次元）を制限した実験を行う．n は 50 から 500 まで 50 刻みで設定する．ボーダー $\epsilon = 0.15$ のとき，特徴ベクトルの次元数を表 1，各特徴ベクトルに対するクラス分類の正解率を表 2 に示す．情報利得率 $\epsilon = 0.15$ 以上の特徴が存在しない（次元数 0）ときは未実験とし，-(ハイフン) を記入する．太字は，実験内の最大正解率を表す．表 2 の結果から，実験 4，5 では V_p が最も分類に有用であることが分かる．V_{po} よりも V_p が分類に有用だった理由は，V_p 内の述語「標高」や「流域面積」などに目的語を取ると特徴の汎用性が失われるからと考えられ

[1]Wikidata. https://www.wikidata.org/
[2]DBpedia Japanese. http://http://ja.dbpedia.org/
[3]YAGO. https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/
[4]LiveJournal. https://www.livejournal.com
[5]FROST. http://www.sw.cei.uec.ac.jp/frost/

る．実験 2 では，情報利得率を用いた次元削減によって，V_{*po} の次元を 16913 から 1355 に減らしながら高い正解率を得た．さらに実験 1，3，7 では V_{p*o}，実験 1 では V_{*po}，実験 2，8 では V_{pp} で高い正解率を示す．これはリソースのノードから遠い 2 つ先の述語（エッジ）や目的語（ノード）が分類に寄与しており興味深い結果といえる．

表 3 は，特徴パターンを選択せず全パターンの結合ベクトルで高い正解率を示す．表 3 の上部 vec には，n を 50 から 500 まで 50 刻みで設定した際の実験 1 から実験 10 の正解率の最大値を示す．表 3 の下部は，荒井ら [2] による SVM の実験結果である．skip，hop，pro，walk，path，full，partial は，SVM にグラフ構造のデータを適用させるためのカーネル関数である．skip，hop，pro は荒井らが，walk，path，full，partial は Lösch ら [6] が考案している．SVM の結果と比較して，vec は下線で示した 6 割で skip カーネルの最高値と同等かより高い正解率を実現している．実験 1，3，5，8 では skip カーネルに勝っており，実験 4，9 では skip カーネルと引き分けている．また，skip カーネルの正解率の平均と本手法の平均を比較すると，本手法の平均が勝っている．このことから，特徴ベクトルによる深層ニューラルネットワークは SVM と比較して高い正解率を実現している．

5 まとめ

本研究では，RDF グラフにおいてリソースから近傍の述語や目的語の組み合わせ（特徴パターン）による多様な特徴集合とそれを変換した特徴ベクトルの抽出方法を提案した．リソースの 2 クラス分類問題を学習したとき，実 RDF データの違いによってどの特徴パターンが分類に有効か明らかにしている．さらに，情報利得率を用いて次元を削減した特徴ベクトルを使用して深層ニューラルネットワークで学習することで，2 クラス分類の高い正解率を実現した．

今後の課題として，特徴ベクトルの次元を減らす情報利得率のボーダーをどのように適切に決定すればよいか，ニューラルネットワークのハイパーパラメータをどう適切に設定するかが考えられる．また，多クラス分類や他の問題（主語と述語からの目的語の推定など）を学習できるように，特徴ベクトルの表現を拡張する必要がある．

参考文献

[1] 荒井大地, 兼岩憲. RDF グラフの冗長な特徴表現に対するカーネル関数とその高速計算. 人工知能学会論文誌, Vol. 32, No. 1, pp. B–G34_1–12, 2017.

[2] 荒井大地, 兼岩憲. RDF グラフの多様性に対する汎用カーネル関数. 人工知能学会論文誌, Vol. 33, No. 5, pp. B–I12_1–14, 2018.

[3] 廣橋美紀, 兼岩憲. RDF データに対するグラフパターンマイニング. 日本データベース学会和文論文誌, Vol. 17-J, No. 1, 2019.

[4] 兼岩憲. RDF と RDF スキーマの推論. 人工知能学会誌, Vol. 26, No. 5, pp. 473–481, 2011.

[5] 兼岩憲, 長井拓馬. 極小 RDF 推論に基づく記述論理 SROIQ の概念生成. 人工知能学会論文誌, Vol. 35, No. 1, pp. B–J62_1–13, 2020（掲載予定）.

[6] Uta Lösch, Stephan Bloehdorn, and Achim Rettinger. Graph kernels for RDF data. In *Proceedings of the 9th Extended Semantic Web Conference (ESWC 2012)*, pp. 134–148, 2012.

[7] 大貫陽平, 貫井駿, 村田剛志, 稲木誓哉, 邱シュウレ, 渡部雅夫, 岡本洋. DNN による RDF 上の単語間の関係の予測. 人工知能学会全国大会論文集 第 31 回全国大会 (2017), pp. 1A33–1A33. 人工知能学会, 2017.

[8] Petar Ristoski, Jessica Rosati, Tommaso Di Noia, Renato De Leone, and Heiko Paulheim. RDF2vec: RDF graph embeddings and their applications. *Journal of Semantic Web*, Vol. 10, No. 4, pp. 721–752, 2019.

[9] 立花誠人, 村田剛志. 構造特徴とグラフ畳み込みを用いたネットワークの半教師あり学習. 人工知能学会論文誌, Vol. 34, No. 5, pp. B–IC2_1–8, 2019.