Poincaré GloVe ベクトルのレトロフィッティング

村瀬 敦也　　三輪 誠　　佐々木 裕
豊田工業大学
{sd16085, makoto-miwa, yutaka.sasaki}@toyota-ti.ac.jp

1 はじめに

共参照解析・含意関係認識・質問応答といった幅広い自然言語処理タスクに必要とされる “swallow is-a bird” のような単語の上位下位関係表現に適した空間として，双曲空間が近年注目されている．双曲空間は，直感的には連続する木構造空間と考えられ，木構造で表される単語の上位下位関係の表現を可能とする．

双曲空間を用いた既存の教師あり単語表現手法である Poincaré embeddings [6] は，英語の概念辞書 WordNet [5] に定義された単語の上位下位関係を双曲空間上に表現できるが語彙に制限がある．教師なし学習を行う単語表現手法の Poincaré GloVe [7] は，Wikipedia のようなラベル付けされていない大規模テキスト情報を表現できるが，学習時に上位下位関係を持つ単語ペアの情報を与えていないため，上位下位関係で体系化された WordNet のような階層構造を表現できない．

そこで本研究では，Poincaré embeddings による小規模の高品質なベクトル表現を用いて Poincaré GloVe の大規模なベクトル表現を更新し，大規模な語彙の上位下位関係の表現を行う手法を提案する．具体的には，Poincaré GloVe ベクトルの双曲距離を保ちつつ，Poincaré embeddings ベクトルとの双曲距離に基づいて，Poincaré embeddings のベクトル空間上でベクトルを更新して利用する．ここでは，Poincaré embeddings ベクトルとの関係について，局所的な関係に基づく更新から大域的な関係に基づく更新までがパラメータで決められるモデルを作り，評価した．Baroni2012 [1]・WBLESS [8] データセットを用いてベクトル表現を評価したところ，いずれのデータセットでも局所的にベクトルを更新する場合の方が高い正解率であった．

2 関連研究

2.1 Poincaré embeddings

Nickel らによって提案された Poincaré embeddings [6] では，WordNet に定義された単語の上位下位関係を元に，サンプリング手法を用いて双曲空間に単語をベクトル表現する．作成される単語表現は，空間の中心に近づくほど上位の概念になっていく性質を持つため，ベクトルの座標情報から上位下位関係を推定することが出来る．

2.2 Poincaré GloVe

Tifrea らによって提案された Poincaré GloVe [7] では Wikipedia のラベルなしテキスト情報を元に，重み付き最小二乗法を用いて双曲空間に単語をベクトル表現する．作成される単語表現の分布は上位下位関係と相関があり，単語の上位下位関係を分布情報から推定することが出来ると報告されている．

2.3 Retrofitting

Faruqui らによって提案された Retrofitting [3] では，少量の質の良いデータを用いて大規模な単語ベクトルの改善を行う．WordNet と The Paraphrase Database (PPDB) [4] を用いた実験が行われている．最適化に用いられる目的関数を (1) 式に示す．

$$\underset{\hat{E}}{arg\,min} \sum_{i=1}^{n} \left(\alpha_i \left\| \hat{\mathbf{E}}_{[w_i]} - \mathbf{E}_{[w_i]} \right\|^2 + \sum_{(w_i,w_j)\in\mathcal{G}} \beta ij \left\| \hat{\mathbf{E}}_{[w_i]} - \hat{\mathbf{E}}_{[w_j]} \right\|^2 \right) \tag{1}$$

ここで，E は事前学習済みの単語ベクトル行列，$\hat{\mathbf{E}}$ は最適化したい単語ベクトル行列，w は単語，$\mathcal{G}$ は単語と単語の類似性を二値化したグラフ，α と β はハイパーパラメータである．目的関数の第一項は，事前学習済み

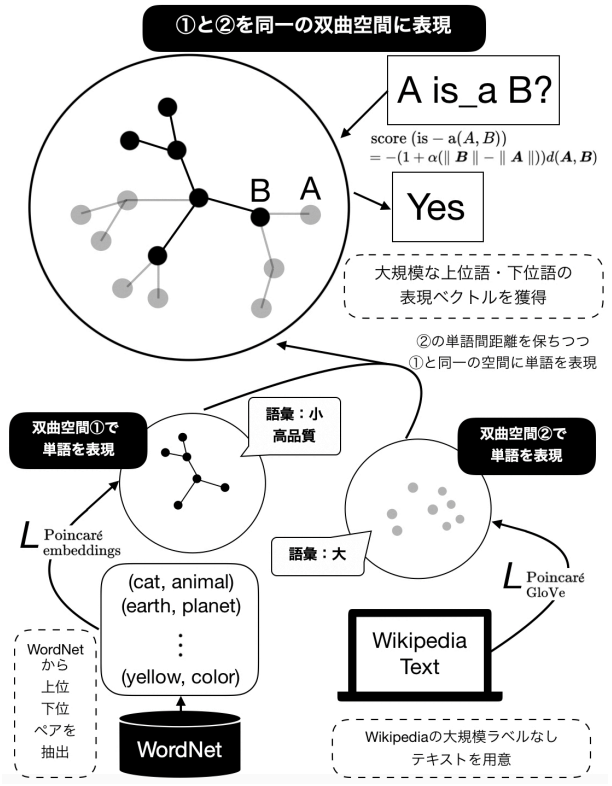

図1　提案手法の全体像

ベクトルと最適化により新しく作成するベクトルの距離が近いことを保証している．第二項は，最適化するベクトルにグラフ情報を反映させている．

3　提案手法

本研究では，Poincaré embeddings のベクトルとの双曲距離に基づいて，Poincaré GloVe のベクトルをその距離を保つように Poincaré embeddings 空間上に配置・更新し，作成したベクトル表現の座標情報から上位下位スコアを計算・評価する手法を提案する．手法の全体像を図 1に示す．まず，Poincaré embeddings と Poincaré GloVe の両手法を用いて，それぞれのベクトル表現を作成し，これらのベクトル表現を元に手法の前提となるベクトル（基盤ベクトル）を作成する．次に，目的関数を最小化するようにベクトルを更新することで，Poincaré GloVe ベクトルを Poincaré embeddings のベクトル空間で表現する．最後に，作成したベクトル間の距離とノルムの情報を用いて，上位下位評価を行う．本節の以降は，基盤ベクトルの作成，最適化，評価方法の詳細について説明する．

3.1　基盤ベクトルの準備

まず，Poincaré embeddings のベクトル行列 $\boldsymbol{E_E}$ と Poincaré GloVe のベクトル行列 $\boldsymbol{E_G}$ を作成し，これらを用いて最適化したいベクトルの初期値を設定する．$\boldsymbol{E_E}$ は，WordNet から抽出した上位下位ペアデータセット $\mathcal{D}_E$ を元に Poincaré embeddings により作成し，$\boldsymbol{E_G}$ は Wikipedia のテキストデータ $\mathcal{D}_G$ を元に Poincaré GloVe により作成する．$\boldsymbol{E_E}$ と $\boldsymbol{E_G}$ を用いた初期値の設定方法を (2) 式に示す．

$$\hat{\mathbf{E}}_{[w]} = \begin{cases} \mathbf{E}_{\mathbf{E}[w]} & \text{if } w \in \mathcal{D}_E \\ \mathbf{E}_{\mathbf{G}[w]} & \text{otherwise} \end{cases} \tag{2}$$

ここで，$\boldsymbol{E_E}$ の語彙 V_E と $\boldsymbol{E_G}$ の語彙 V_G の関係は (3) 式で表されることに注意されたい．

$$V_E \subset V_G \tag{3}$$

3.2　ベクトル更新による最適化

前提となるベクトル表現 $\hat{\mathbf{E}}$ を作成した後は，Poincaré GloVe ベクトルの双曲距離を保ちつつ，Poincaré embeddings のベクトル空間にベクトルを表現出来るよう，$\hat{\mathbf{E}}$ 中の $\mathcal{D}_E$ に含まれない語彙のベクトルのみを更新して目的関数を最小化する．具体的には，(4) 式の目的関数を最小化するよう，Riemannian Adam [2] を用いて，$\hat{\mathbf{E}}$ の更新を行う．

$$\mathcal{L} = \sum_{w_i \in \mathcal{D}_E} \sum_{w_j \in \mathbf{k}_{[w_i]}, w_j \notin \mathcal{D}_E} \Big\| d\left(\mathbf{E}_{\mathbf{E}[w_i]}, \hat{\mathbf{E}}_{[w_j]}\right) - d\left(\mathbf{E}_{\mathbf{G}[w_i]}, \mathbf{E}_{\mathbf{G}[w_j]}\right) \Big\| \tag{4}$$

$\mathbf{k}_{[w_i]}$ は $\mathbf{E}_{\mathbf{G}[w_i]}$ に近い $\mathbf{E_G}$ 中の近傍 k 単語（$\mathcal{D}_E$ は除く）であり，$d(\cdot)$ は双曲空間における二点間の距離関数である．この目的関数は，retrofitting [3] における元のベクトルと最適化後のベクトルを近づけるという考え方を受け継いでいる．なお，双曲空間における二点 $\boldsymbol{u}$, $\boldsymbol{v}$ 間の距離は (5) 式で計算する．

$$d(\boldsymbol{u}, \boldsymbol{v}) = \operatorname{arccosh}\left(1 + 2\frac{\|\boldsymbol{u} - \boldsymbol{v}\|^2}{(1 - \|\boldsymbol{u}\|^2)\,(1 - \|\boldsymbol{v}\|^2)}\right) \tag{5}$$

ここでは，(4) 式における $\mathbf{k}_{[w]}$ の設定方法を二通り提案し，それぞれの設定によるベクトルの更新を大域的更新・局所的更新と名付ける．各更新方法について説明する．

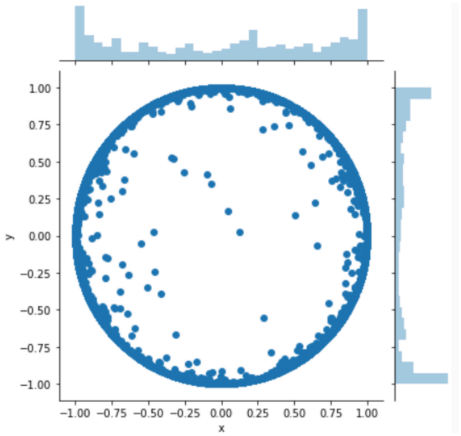

図2　Poincaré embeddings [6] の分布

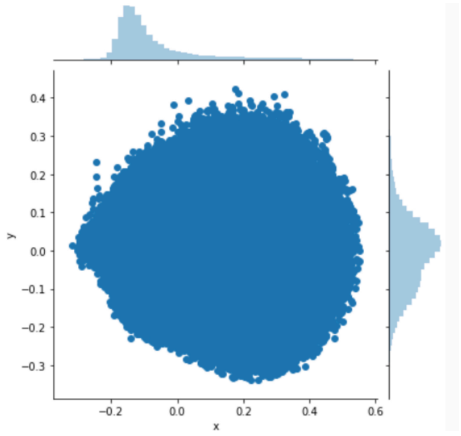

図3　Poincaré GloVe [7] の分布

3.2.1　大域的更新

大域的更新は，(4) 式における $\mathbf{k}_{[w]}$ の単語数が最大の場合を指す．つまり，(6) 式が成り立つ．

$$|\mathbf{k}_{[w]}| = |V_G - V_E| \tag{6}$$

この更新方法は，ベクトル全体に渡って更新を行う大域的な方法である．

3.2.2　局所的更新

局所的更新は，(4) 式における $\mathbf{k}_{[w]}$ の単語数が十分小さい場合を指す．つまり，n を十分小さい整数として (7) 式が成り立つ．

$$|\mathbf{k}_{[w]}| = n \tag{7}$$

この更新方法は，ベクトルを部分的に更新する局所的な方法である．実験では $n = 3$ とした場合の結果を報告する

3.3　ベクトル表現の評価

Poincaré embeddings ベクトルを固定していることから，作成したベクトル表現の評価は，Poincaré embeddings の評価方法で行う．評価式を (8) 式に示す．

$$\text{score}(\text{is-a}(\boldsymbol{u}, \boldsymbol{v})) = -(1 + \alpha(\|\boldsymbol{v}\| - \|\boldsymbol{u}\|))d(\boldsymbol{u}, \boldsymbol{v}) \tag{8}$$

$\boldsymbol{v}$ は上位語の候補の単語ベクトル，$\boldsymbol{u}$ は下位語の候補の単語ベクトル，$d(\cdot)$ は (5) 式の双曲距離関数，α はハイパーパラメータである．上位下位判定は，評価データ中のスコアの平均を閾値として行う．

4　実験

実験は，まず，事前実験として Poincaré embeddings と Poincaré GloVe のそれぞれのベクトル分布の考察を行なった後に，両手法のベクトル表現を用いて提案手法の評価を行なった．

4.1　実験設定

提案手法の評価では，Wikipedia から作成した語彙数 208,881 の Poincaré GloVe ベクトルと，WordNet から作成した語彙数 31,222 の Poincaré embeddings ベクトルを用いて (6) 式（大域的更新）および (7) 式（局所的更新）の場合の最適化を行い，(8) 式で評価したときの Baroni2012，WBLESS データセットでの正解率を比較した[*1]．ベクトルの次元は 100，最適化時のエポック数は 300，学習率は大域的更新で 10^{-6}，局所的更新で 10^{-4} とした．また，評価時の α は 10 に設定した．

4.2　ベクトル分布の比較

Poincaré embeddings と Poincaré GloVe の各手法により作成されるベクトル分布の差異を説明する．2 次元双曲空間にベクトル表現した例を図 2と図 3に示す．Poincaré embeddings では空間の周縁部に多くのベクトルが表現されるのに対して，Poincaré GloVe では空

[*1] 本実験では，レトロフィッティングの性能を評価することを目的としている．他の手法では上位語の判定問題を単語ペアの分類問題や単語の写像問題として解いているが，本研究ではすべての単語が同じ空間上で上位下位関係を表すように配置されており，(8) 式による単語ベクトルのスコア計算のみで上位下位を判定する．

間の中心部に多くのベクトルが表現されており，ベクトル分布に大きな違いがある．この違いにより両手法では異なるベクトルの評価方法が用いられているが，本研究では Poincaré GloVe ベクトルを更新して Poincaré embeddings の空間に表現し，単一の評価方法による評価を行う．

4.3 提案手法の評価

目的関数を大域的に更新する (6) 式と局所的に更新する (7) 式のそれぞれを用いてベクトルを更新し，得られたベクトル表現を Baroni2012 および WBLESS 評価データセットを用いて評価した結果を表 1に示す．

結果は，いずれの評価データセットにおいても局所的更新の場合の方が高い正解率であった．これは，局所的に更新を行う方が少ない制約のもと自由にパラメータを更新できるためではないかと考えられる．

5 おわりに

本研究では大規模な語彙の上位下位の表現のために，Poincaré embeddings と Poincaré GloVe のベクトル表現を考慮した目的関数を導入し，パラメータの更新方法を変えて評価を行なった．事前実験として両手法の評価や分布の比較を行い，ベクトル分布に大きな違いがあることを明らかにした．このベクトル表現の差異を踏まえた上で，局所的および大域的な更新を行う 2 通りの目的関数を最適化してベクトル表現を作成したところ，局所的更新の場合の方が高い精度であった．この理由は，局所的に更新を行う方が少ない制約のもと自由にパラメータを更新できるためではないかと考えられる．すべての単語を同一空間上に上位下位関係を内在するよう配置できるのは双曲空間の大きな利点である．今後は，異なる目的関数やデータセット・関係を対象に，提案手法の拡張を行い，より汎用なモデルを目指す．

表1 更新方法ごとの正解率の比較

	Baroni2012	WBLESS
収録単語ペア数	2,770	1,668
正解ペア数（大域的更新）	2,000	1,279
正解率（大域的更新）	0.722	0.767
正解ペア数（局所的更新）	2,102	1,329
正解率（局所的更新）	**0.759**	**0.797**

謝辞

本研究の一部が JSPS 科研費 17K00318 により支援されたことに深く感謝する。

参考文献

[1] Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 23–32, 2012.

[2] Gary Becigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *International Conference on Learning Representations*, 2019.

[3] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1606–1615, 2015.

[4] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB: The paraphrase database. pp. 758–764. Association for Computational Linguistics, 2013.

[5] George A. Miller. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, Vol. 38(11), pp. 39–41, 1995.

[6] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pp. 6338–6347. 2017.

[7] Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. Poincaré glove: Hyperbolic word embeddings. In *International Conference on Learning Representations*, 2019.

[8] Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 2249–2259, 2014.