極座標を用いた階層構造埋め込み

岩本蘭 小比田涼介 和地瞭良
慶應義塾大学 日本アイ・ビー・エム株式会社 東京基礎研究所
r.iwamoto@keio.jp kohi@ibm.com akifumi.wachi@ibm.com

1 はじめに

実世界の階層構造を表現することは自然言語処理の多くのタスクにとって重要である [1, 2]．近年単語の階層構造を埋め込んだ分散表現の研究が活発に行われ [3, 4]，その中でも双曲空間を用いた分散表現が話題になった [5, 6, 7]．双曲空間は体積が原点からの距離に応じて指数関数的に広がるため低次元で階層構造を表現するのに適している．しかし双曲埋め込みを用いる際には応用タスクのモデルも双曲空間で動作するように変更する必要があり [8]，ユークリッド空間で学習された多くの手法を活用することが難しい [9]．そこで我々は汎用性の高いユークリッド空間で分散表現を学習する．

本稿では極座標を用いて単語を図 1 のように低次元のユークリッド空間に埋め込む Polar Embedding について述べる．Polar Embedding では単語の抽象度を半径 (原点からの距離) で，類似度を角度で表現する．極座標の特徴を活かしながら角度を最適化するために Welsch 損失 [10] と Stein Variational Gradient Descent (SVGD) [11] を用いる．WordNet の link prediction タスクで性能評価を行い，Polar Embedding は低次元ユークリッド空間で学習された既存の分散表現より高精度を達成し，双曲空間の分散表現と同等の性能を達成した．

2 Polar Embedding

Polar Embedding では極座標の半径と角度を用いて単語を表現する．本節ではまず極座標での単語ベクトルの定義について述べ，その後角度の最適化について述べる．

ユークリッドノルムを $\|\cdot\|$，半径を $r_{\max} \in \mathbb{R}$ とする．単語 w は n 次元の開球 $\mathbb{W}^n = \{\mathbf{w} \in \mathbb{R}^n \mid \|\mathbf{w}\| < r_{\max}\}$ 内の極座標ベクトル $\mathbf{w} = (r, \theta, \varphi^1, \varphi^2, ..., \varphi^{n-2})$ で表現される．ここで $r \in (0, r_{\max}), \theta \in [0, 2\pi), \varphi^k \in (0, \pi), k = 1, 2, ..., n-2$ である．例えば，3 次元極座標での角度は図 2 のように表される．

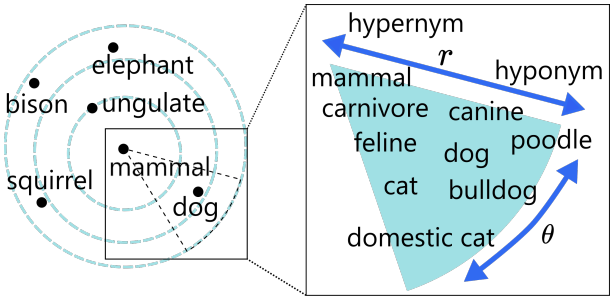


図 1: Polar Embedding の概要．半径を用いて単語の抽象度を，角度を用いて類似度を表現する．

2.1 半径

半径 r は単語の抽象度を表す．一般的には具体的な単語のほうが抽象的な単語よりも多いため，具体的な単語を空間の中心から離れた場所に配置することで空間を効率的に使う．例えば，“bulldog” や “wooden chair” より “mammal” や “furniture” のほうが抽象度が高いため r を小さく設定する．半径 r は頻度情報などを用いて定めることができる．本論文では r は学習せず，WordNet 内のエッジや，その単語と上位下位ペアとなる単語の多さから求める．

2.2 角度

角度 (θ, φ^k) は単語の類似度を表す．既存の分散表現と同様に，類似した意味を持つ語同士は近く，そうでない語は遠くなるように角度を学習する．極座標は図 3 のように角度の範囲が制限されているため，$r = 1$ とすると，θ は円周上で，φ^k は半円上で最適化される．

2 つの単語 w_i, w_j があるとき，θ は図 3 の左図のように $[0, 2\pi)$ の円周上を動く．θ_{w_i} と θ_{w_j} の距離は円周上の短いほうの弧とし，以下のように定義する．

$$d(\theta_{w_i}, \theta_{w_j}) = \min\left(2\pi - |\theta_{w_i} - \theta_{w_j}|, |\theta_{w_i} - \theta_{w_j}|\right) \quad (1)$$

角度 $\varphi^k \in (0, \pi)$ は図 3 の右図のように半円上を動き，$\varphi^k_{w_i}$ と $\varphi^k_{w_j}$ の距離は次式で表す．

$$d(\varphi^k_{w_i}, \varphi^k_{w_j}) = |\varphi^k_{w_i} - \varphi^k_{w_j}|, \forall k \in \{1, n-2\} \quad (2)$$

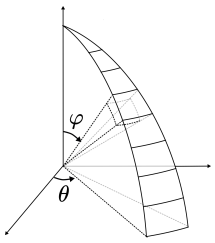


図 2: 3 次元極座標での角度の定義

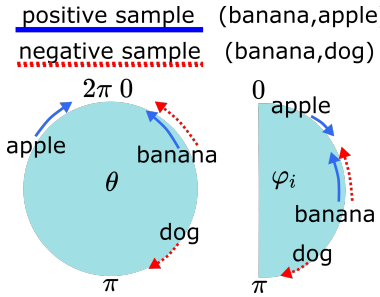


図 3: 極座標での角度の学習

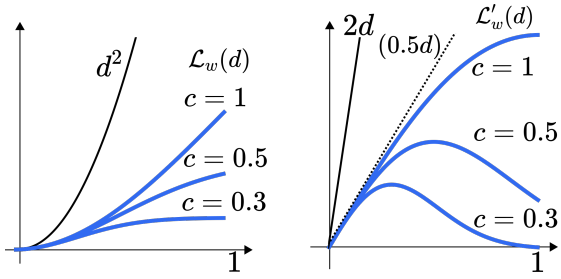


図 4: Welsch 損失 (青) と二乗誤差関数 (黒) の概形 (左) と勾配 (右)

意味が似ている単語ペアの角度はそれぞれの次元において，上で定義した距離を小さくする方向に最適化される．極座標での角度の最適化には少し工夫を加える必要がある．低次元ユークリッド空間の超球面，つまり小さな空間で角度を最適化する場合，単語が空間内の一部分に集中しすぎないよう補正を加え，限られた空間を効率的に使いたい．それを実現するための最適化手法を紹介する．

2.3 角度の最適化

本節では角度の最適化について述べる．デカルト座標と異なり，極座標の角度は次元ごとに値域が制限されている．一般的に用いられる二乗誤差関数 $y = x^2$ は，図 4 の黒い線で示すように x が大きいほど y が大きくなり，すでに距離が十分離れている単語に対しても単語を遠ざける力が強い．そのため，negative sampling の際に単語が互いに遠ざかりすぎて角度が値域の端にたまってしまう，すなわち単語が極に集中する現象が生じる．低次元空間を有効に使うためには，空間内に単語がばらけて分布するようにしたい．単語をばらけたままにしながら最適化を行うために，図 4 の青い線で示す Welsch 損失 [10] と，単語の分布を超球面上での一様分布にする Stein Variational Gradient Descent (SVGD [11]) を用いる．

Welsch 損失 Welsch 損失 $\mathscr{L}_w(d)$ は図 4 の青線のように角度の勾配に制限があり，d が大きいと勾配は小さくなる．そのため単語同士が似ていない，つまりすでに十分角度の差が大きい時には更新の値が小さくなる仕組みになっている．

Welsch 損失は次のように定義される．

$$\mathscr{L}_w(d) = \frac{c^2}{2}\left[1 - \exp\left(-\frac{d^2}{2c^2}\right)\right] \tag{3}$$

d は式 1 で示した 2 つの単語のある次元の角度の差，c はパラメータである．勾配は次式で表される．

$$\frac{\partial \mathscr{L}_w(d)}{\partial d} = \frac{d}{2}\exp\left(-\frac{d^2}{2c^2}\right) \tag{4}$$

SVGD 低次元のユークリッド空間をより効率的に使う，つまり単語の類似関係を表現しつつ ($r = 1$ としたときに) 超球面上に単語をばらつかせるために SVGD を用いる．

SVGD では再生核ヒルベルト空間 $\mathscr{H}^d$ で理想分布 p と現在の分布 q の Kernelized Stein Discrepancy (KSD) $S(\cdot,\cdot)$ を次式のように定義し，KSD を最小化することによって KL divergence を最小化する．

$$\mathcal{S}(q,p) = \max_{\phi\in\mathscr{H}^d}\left\{\mathbb{E}_{x\sim q}\left[\mathscr{A}_p\phi(x)\right]\right\} \tag{5}$$

ここで $\mathscr{A}_p\phi(x) = \phi(x)\nabla_x \log p(x) + \nabla_x\phi(x)$ であり，式 5 の最適解は次のように得られる．

$$\phi^*_p(x') = \mathbb{E}_{x\sim q}\left[\,\kappa(x,x')\nabla_x \log p(x') + \nabla_x\kappa(x,x')\,\right] \tag{6}$$

$\kappa : \mathfrak{X}\times\mathfrak{X} \to \mathbb{R}$ は一定の条件を満たす正定値カーネルであり，例えば RBF カーネル $\kappa(x,x') = \exp\left(-\gamma\,||x-x'||^2\right)$ が当てはまる．SVGD は Welsch loss を用いた訓練の試行回数ごとに行われる．超球面上に単語が一様に分布するときのそれぞれの次元の角度分布 (理想分布) を数式で表すことは難しいので，Gaussian Mixture model (GMM) で近似した分布を理想分布として用いる．

3 実験

我々は WordNet の mammal/noun subtree [12] を用いて分散表現を学習した．Polar Embedding の定性的/定量的評価と，角度の最適化に用いた Welsch 損失と SVGD の効果について検証した．

3.1 データセット

mammal/noun subtree は WordNet のツリーから単語ペアを抽出したデータセットで，単語ペア (w_i, w_j) の間にエッジが存在するとき，w_i は w_j の上位語である．mammal subtree には 6540 ペア，noun subtree には 743300 ペアが含まれる．双曲空間の分散表現の定量評価として多く用いられる link prediction [5, 13]

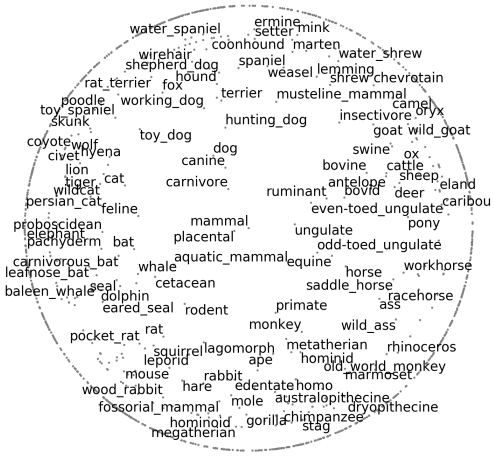


図 5: WordNet mammal subtree で学習した Polar embedding

表 1: WordNet noun subtree での link prediction タスクの結果. (E) と (H) はユークリッド空間と双曲空間を表す.

モデル (空間)	5 次元			10 次元		
	訓練に使用したエッジの割合					
	10%	25%	50%	10%	25%	50%
Polar r^g (E)	**78.5%**	**79.9%**	**81.8%**	**82.2%**	81.6%	82.3%
Polar r^e (E)	75.8%	77.2%	78.6%	78.5%	79.2%	80.1%
Simple (E)	71.3%	73.8%	72.8%	75.4%	78.4%	78.1%
Order (E)	70.2%	75.9%	81.7%	69.7%	79.4%	**84.1%**
Cone (E)	69.7%	75.0%	77.4%	81.5%	**84.5%**	81.6%
Disk (E)	38.9%	42.5%	45.1%	54.0%	65.8%	72.0%
Poincare (H)	70.2%	78.2%	83.6%	71.4%	82.0%	85.3%
Cone (H)	80.1%	86.0%	92.8%	85.9%	91.0%	94.5%
Disk (H)	69.1%	81.3%	83.1%	79.7%	90.5%	94.2%

は，noun subtree の単語ペアの間にエッジが存在するかを予測することで，分散表現が単語間の階層構造をどの程度埋め込んでいるかを検証するタスクである．

まず mammal subtree で 2 次元の Polar Embedding を学習し定性的評価を行い，その後 noun subtree を用いて link prediction を行った．

3.2 実験設定

ここでは Polar Embedding の実験設定について述べる．Polar Embedding では半径 (抽象度) と角度 (類似度) は別々のパラメータで表されるため，半径 r を WordNet のエッジの深さから求め，角度をエッジでつながった単語ペアの集合から学習した．

半径の設定 WordNet の mammal/noun subtree のデータセット全体からエッジの深さを推定する r^g と，訓練データの単語ペアのみからエッジの深さを推定する r^e を作成した．単語 w_i の半径 r^e は，エッジの数 e_i を用いて $r_i^e = 1 - z(\log(e_i + 1))$ と定義し，r^g は階層中の子の数 l_i とツリーの最大高さ h_i を用いて $r_i^g = 1 - z(h_i + \log(l_i + 1))$ と定義する．z は正規化関数であり，r^e, r^g とも 0, 1 の範囲をとる．

直感的に表現すると，r^g のほうがより多くの情報から半径を作成していて，ツリーの根の近くにあり多くの子を持つ単語の抽象度が高い．それに対して r^e では多くの単語とつながっている単語の抽象度が高い．実用的には r は頻度情報など単語の抽象度を表す他の尺度でも代用することができる．

角度の学習 角度を学習する際にはエッジでつながった語同士は positive sample, そうでない語のペアは negative sample とした．mammal subtree では 2 次元 (角度の次元は 1)，noun subtree では既存研究に倣い 5 次元と 10 次元 (角度の次元は 4 または 9) の分散表現を学習した．

評価関数 ある単語ペアが上位下位関係にあるかどうかには，単語の抽象度と類似度の両方が関係すると仮定する．そこで，まず次のように類似度を評価する関数 s_a と単語の抽象度 (半径) の差を評価する関数 s_r の 2 つを用意した．

$$\begin{aligned} s_a(w_i, w_j) &= \frac{f(w_i) \cdot f(w_j)}{\|f(w_i)\| \|f(w_j)\|} \\ s_r(w_i, w_j) &= |r_i - r_j| \end{aligned} \tag{7}$$

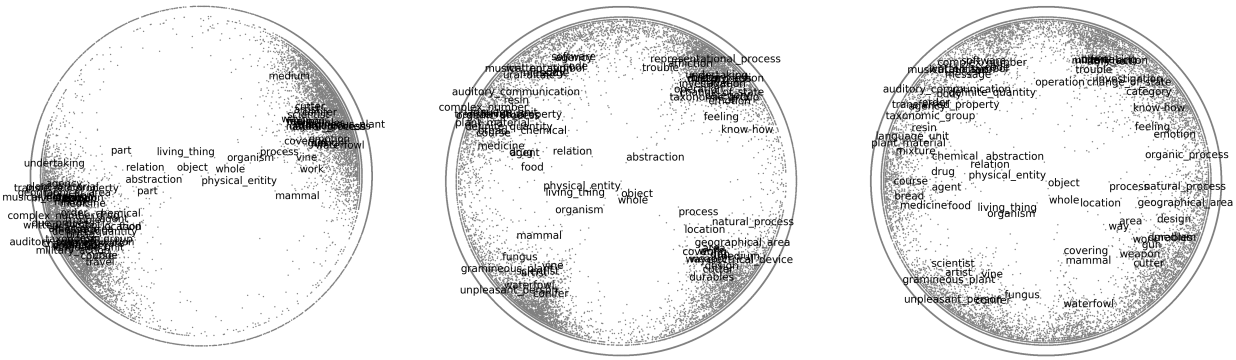
f は極座標からデカルト座標への変換 [14] を表す．ある単語ペアの間にエッジが存在するとき，2 つの単語 w_i と w_j のなす角 (類似度) が一定以上小さく，2 つの単語の半径の差 (抽象度の差) が一定以上大きいと仮定する．エッジが存在するかどうかの分類は s_a と s_r を組み合わせ，validation データを用いて決定した閾値 τ を用いて次式のように判定した．

$$s(w_i, w_j) = \begin{cases} 1 & \text{if } s_a(w_i, w_j) > 1 - \tau s_r(w_i, w_j)^2 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

4 結果と分析

4.1 Mammal Subtree

WordNet mammal subtree で 2 次元の Polar Embedding を学習した結果を図 5 に示す．単語は角度の偏りなく分布し，様々な階層構造を見つけることができる．例えば，“cat” と “dog” は “carnivore” の下位語であり，



(a) 二乗誤差　(b) Welsch 損失　(c) Welsch 損失 + SVGD

図 6: WordNet noun subtree で学習した 5 次元分散表現の 2 次元部分

“carnivore” から見た扇形の中に存在する．さらに “dog” の下位語である “hunting dog” や “terrier” との階層関係や “lion” と “wildcat” が “cat” の下位語であることが読み取れる．種の階層性も埋め込まれており，例えば “aquatic mammal” → “cetacean” → “seal”, “dolphin” という関係性や “primate” → “monkey” → “gorilla”, “ape” といった関係性が読み取れる．抽象度の差を考慮せず，類似度のみを考慮したい場合は既存の分散表現と同様に cos 類似度で単語間の関係を評価すればよく，Polar Embedding では用途に応じて類似度と抽象度を組み合わせることができる．

4.2 Noun Subtree

WordNet noun subtree での link prediction の結果を表 1 に示す．性能評価として F1 スコアを用い，比較手法としてユークリッド空間の分散表現 (Order, Cone, Disk) とユークリッド距離を最小化する分散表現 (Simple) [13]，双曲空間の分散表現 (Poincaré, Cone, Disk) [4, 5, 13] を用いた．

5 次元の場合では Polar Embedding は r^g,r^e どちらの場合でも既存手法より優れた性能を発揮している．また，訓練に使用したエッジの割合が小さい時は，ユークリッド空間よりも双曲空間の分散表現とも同等の性能である．10 次元の時には他のユークリッド空間の分散表現と大きな差はないが同等の性能を達成している．これらの結果から，Polar Embedding は低次元 (5 次元) のユークリッド空間での性能は十分高いが，次元が増加した時の性能の伸び率の改善が今後の課題であることがわかる．

次に Welsch 損失と SVGD が角度の最適化に与えた影響を 5 次元の Polar Embedding を用いて分析した．まず r と θ の 2 次元のみを図 6 に示す．二乗誤差 (図 6a) のみで学習した際は単語が左右に寄っているが，Welsch 損失 (図 6b) を用いると単語の分布が二乗誤差よりはばらけているのがわかる．Welsch 損失+SVGD(図 6c) では円周上に一様に分布している．

表 2: Welsch 損失と SVGD の評価実験.

損失関数	SVGD	F1
Welsch 損失	あり	78.5%
Welsch 損失	なし	74.9%
二乗誤差	あり	69.1%
二乗誤差	なし	65.5%

定量評価を表 2 に示す．link prediction で訓練エッジ 10%, $r = r^g$ の設定で行った実験では，Welsch 損失を用いた場合二乗誤差よりも性能が高く，SVGD と Welsch 損失を両方用いたほうがさらに高い F1 スコアを達成している．これらの結果から，低次元のユークリッド空間を有効に使うことが性能の向上に寄与していると考えられる．

5 結論

本論文では単語を極座標で表現する Polar embedding を提案した．単語の抽象度を半径で，類似度を角度で表現することにより抽象度と類似度を分けて最適化し，極座標で角度をうまく最適化するために Welsch 損失と SVGD を用いた．単語が極周辺に集まりすぎる現象を防ぐことで低次元のユークリッド空間を有効に活用した提案法は，link prediction タスクで既存手法のユークリッド空間で作成された分散表現を上回る精度を達成した．今後の展望として教師無し学習や高次元への拡張，半径の学習が挙げられる．

参考文献

[1]Oktie Hassanzadeh, Debarun Bhattacharjya, Mark Feblowitz, Kavitha Srinivas, Michael Perrone, Shirin Sohrabi, and Michael Katz. Answering binary causal questions through large-scale text mining: An evaluation using cause-effect pairs from human experts. In *IJCAI*, 2019.

[2]Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *ACL*, page 4149–4158.

[3]Luke Vilnis and Andrew McCallum. Word Representations via Gaussian Embedding. In *ICLR*, 2015.

[4]Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-Embeddings of Images and Language. In *ICLR*, 2016.

[5]Maximillian Nickel and Douwe Kiela. Poincaré Embeddings for Learning Hierarchical Representations. In *NeurIPS*, pages 6338–6347. 2017.

[6]Bhuwan Dhingra, Christopher Shallue, Mohammad Norouzi, Andrew Dai, and George Dahl. Embedding Text in Hyperbolic Spaces. In *Proceedings of the Workshop on Graph-Based Methods for Natural Language Processing*, pages 59–69, 2018.

[7]Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. Poincare Glove: Hyperbolic Word Embeddings. In *ICLR*, 2019.

[8]Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *NeurIPS*, pages 5345–5355. 2018.

[9]Lun Du, Zhicong Lu, Yun Wang, Guojie Song, Yiming Wang, and Wei Chen. Galaxy Network Embedding: A Hierarchical Community Structure Preserving Approach. In *IJCAI*, pages 2079–2085, 2018.

[10]John E. Dennis and Roy E. Welsch. Techniques for nonlinear least squares and robust regression. *Communications in Statistics - Simulation and Computation*, 7(4):345–359, 1978.

[11]Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *ICML*, pages 276–284, 2016.

[12]George A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, 1995.

[13]Octavian.-E. Ganea, Gary. Becigneul, and Thomas. Hofmann. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. In *ICML*, pages 1646–1655, 2018.

[14]LE Blumenson. A derivation of n-dimensional spherical coordinates. *The American Mathematical Monthly*, 67(1):63–66, 1960.