Poincare Embedding を用いた単語の埋め込みベクトルの獲得

橋本 隼人[1]　森 信介[2]
[1] 京都大学 情報学研究科　[2] 京都大学 学術情報メディアセンター
[1]hashimoto.hayato.73e@st.kyoto-u.ac.jp　[2]forest@i.kyoto-u.ac.jp

1 序論

単語のベクトル埋め込み表現の獲得は、深層学習を用いた自然言語処理を行うために不可欠である。Mikolov ら [2] は、注目する単語の前後の単語を予測するタスクである Skip-Gram をネガティブサンプリングを用いて学習することで、教師なしで有用な分散表現が学習できることを示した。この分散表現は、人手評価でつくられた単語の類似度である WordSim-353 スコアとの類似度により評価され、高い相関を持つことが示された。また、この手法によって学習された表現は、加算により合成可能である性質があると考えられており、 例えば、“Vietnam” の埋め込みベクトルと “capital” の埋め込みベクトルを加算し、その結果に最も近い埋め込みベクトルは “Hanoi” のものとなる、といった結果が示されている。

一方で、単語の同士の意味的関連は、合成する・合成されるといった関係だけでなく、上位語・下位語といった関係もある。Nickel ら [3] は Poincare Embedding を提案し、WordNet によって定義される上位語・下位語の関連性をこれを用いた埋め込みベクトルによって提示元で効率よく表現できることを示した。Poincare Embedding は、双曲空間のモデルである Poincare 円盤に対応する Riemann 計量を用いてベクトル間の距離・微分を定義するものであり、Poincare 円盤で定義される距離は、円盤の周縁部に近いほど Euclid 距離よりも大きくなるように定義されている。

本研究では、Poincare Embedding にいくらかの改変を加えることで、教師なしで階層的関係を考慮した分散表現を獲得できることを示す。

2 提案手法

Nickel らは、Poincare Embedding による距離をそのまま softmax によって確率としていた。しかし、この手法をそのまま Mikolov らの方法に適用すると、精度よく学習することができない。Nickel らは、精度の高いものから順に、Poincare 円盤での距離、Euclid 距離、内積であることを示したが、われわれの予備実験では、Skip-Gram や Continuous Bag of Words (CBoW) では、内積を用いた方が Euclid 距離を用いるほうが精度が高いことが分かった。したがって、Poincare Embedding を Skip-Gram や CBoW に適用するためには、Poincare 円盤での内積を使う必要があると推察される。

通常の内積は、$\boldsymbol{x} \cdot \boldsymbol{y} = ||\boldsymbol{x}||\,||\boldsymbol{y}|| \cos\theta$ と定義される。これは、Euclid 距離 d_E を用いて以下のように書ける。$\|\boldsymbol{x}\|\,\|\boldsymbol{y}\|(1-(d_E(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}))^2/2)$ ここで $\hat{\boldsymbol{x}} = \boldsymbol{x}/||\boldsymbol{x}||$, $\hat{\boldsymbol{y}} = \boldsymbol{y}/||\boldsymbol{y}||$ である。Poincare 円盤上のベクトル $\boldsymbol{x_p}, \boldsymbol{y_p}$ を $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}$ に相当するものと考え、距離を Poincare 円盤での距離 d_p に置き換え、さらに新しいパラメータ r_x, r_y を導入すると、これらに対する内積 $(\cdot_p)$ を $(r_x, \boldsymbol{x_p}) \cdot_p (r_y, \boldsymbol{y_p}) = r_x r_y (1 - d_p(\boldsymbol{x_p}, \boldsymbol{y_p})/2)$ と定義できる。ここで

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

である。ところで、Euclid 空間での単位円上の 2 点間の距離は [0, 2] に限定されるが、Poincare 円盤上での距離には上限がないため、距離を除算するためのパラメータ σ を導入し、これを学習により定めるものとする。これらをすべて含めた Negative Sampling による Loss 関数は、以下のようになる。

$$\begin{aligned} Loss = &-\log\left(\hat{r_T} r_{\hat{H}} \exp(-d_p(\boldsymbol{x_T}, \boldsymbol{x_H})/\sigma^2)\right) \\ &+\log\left(\hat{r_T} r_{\hat{H}} \exp(-d_p(\boldsymbol{x_T}, \boldsymbol{x_H})/\sigma^2)\right. \\ &\left.+\sum_{i=1}^{S} r_{\hat{Ni}} r_{\hat{H}} \exp(-d_p(\boldsymbol{x_{Ni}}, \boldsymbol{x_H})/\sigma^2)\right) \end{aligned}$$

ここで $\hat{r_x} = \sqrt{e}\exp(r_x)$ であり、$\boldsymbol{x_T}$ は正解単語のベクトル、$\boldsymbol{x_H}$ は予測ベクトル、$\boldsymbol{x_{Ni}}$ はネガティブサンプルされた単語のベクトル、S はネガティブサンプル数である。

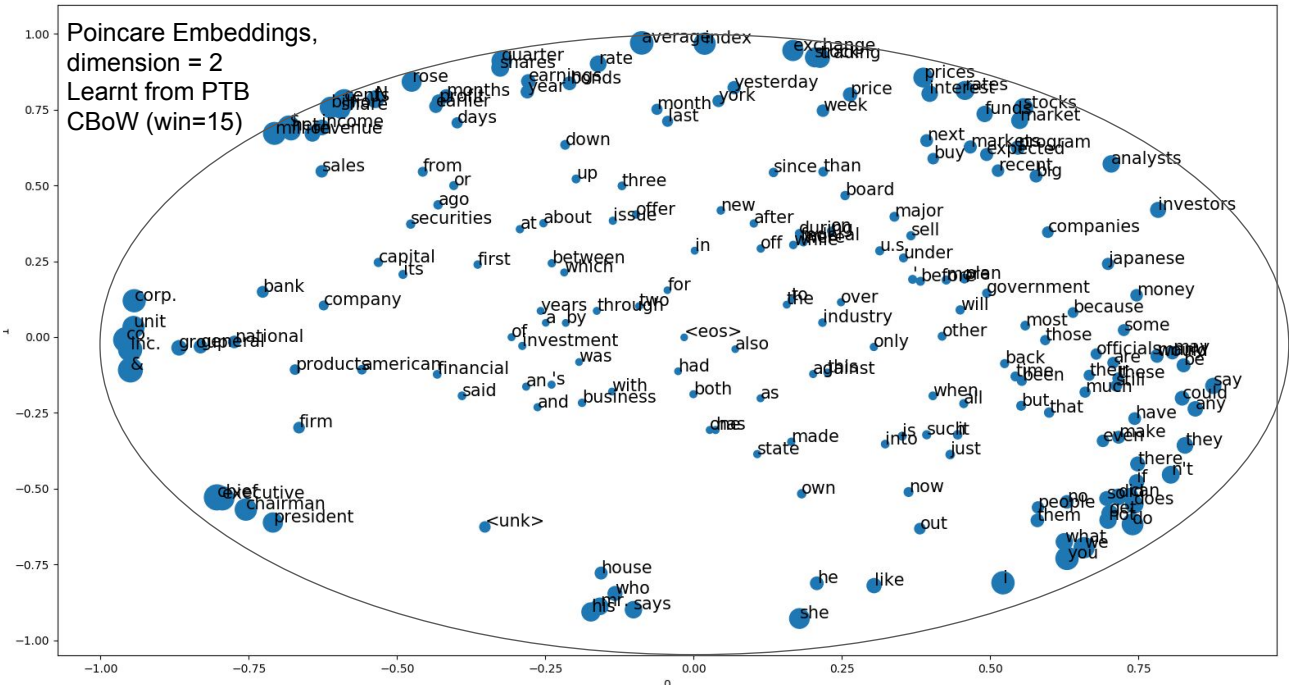

図 1　Penn Treebank から学習することによって得られた Poincare embedding による 2 次元ベクトル表現。高頻度語のみを示す。単語を表す丸の大きさは、$\hat{r_x}$ の大きさを示す。

3　実験

学習ルーチンは Python の深層学習ライブラリ Chainer を用いて実装した。後ろ向き計算は、Nickel らによる RSGD と同じ方法で微係数をリーマン計量によって修正し、Adam [1] により最適化した。

Penn Treebank を学習コーパスとして用いて CBoW により埋め込みベクトルを学習し、その学習結果を図示した。また、単語の予測確率のクロスエントロピーを Euclid 内積による場合と提案手法による場合とで比較した。提案手法の場合、r_x と $\boldsymbol{x_p}$ の両方で一つの単語を表すことから、n 次元 Euclid ベクトルと $n-1$ 次元 Poincare 円盤ベクトルとを比較することとする。Poincare Embedding の学習のうち、$n = 100$ での実験においては発散を防ぐため、Adam のパラメータを $\alpha = 0.0003$ に設定した。そのほかの実験では、$\alpha = 0.001$ とした。

実験はそれぞれのパラメータにつき 6 回行い平均値を算出した。結果を図 2 に示す。5 次元では Poincare Embedding によってエントロピーを小さくすることができている。しかし、100 次元では検証データセットのエントロピーが小さくならず、過学習が目立つ結果となっている。

4　考察

Penn Treebank は経済ニュースをもとにしているコーパスのため、それらに関連する単語が多く出現している。図示した結果を観察すると、周縁部に社名の一部とみられる “Corp.”, “Inc.” といった単語からなるクラスターや、会社の代表を表す “chairman”, “exective”, “president” といった単語からなるクラスター、“revenue”, “income” といった会社の業績の単語からなるクラスターなどがみられる。さらに、“bank”, “company”, “capital” といった単語がこれらの上位語に相当する位置に位置しており、会社の業績や株式指標、価格といった時間変化する数値を表すような単語の上位語に相当する位置に、“up”, “down”, “about”, “since”, “after”, “in” といった単語がみられる。また、属する文脈が最も不明であるといえる “<eos>” タグは中心に位置している。これらは上位語ではないが、これらの単語に共通している文脈をあらわす単語であると考えることができる。文脈に階層性があるとすると、この埋め込みベクトルはその情報を学習しているといえるだろう。

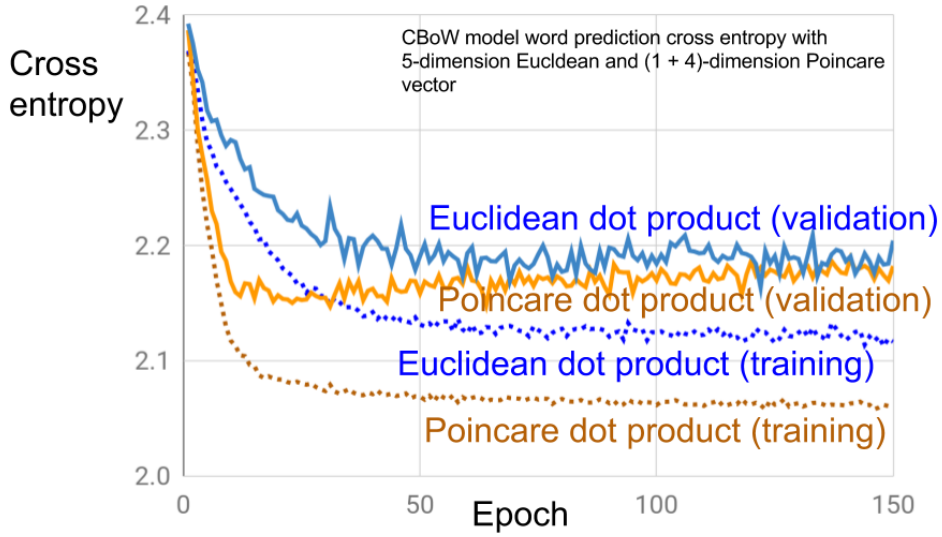

図 2　5 次元の埋め込みベクトルによる、CBoW での 1 単語の予測確率のクロスエントロピーの比較。“Euclidean dot product” は Euclid 内積の結果を示し、“Poincare dot product” は Poincare 内積の結果を示す。

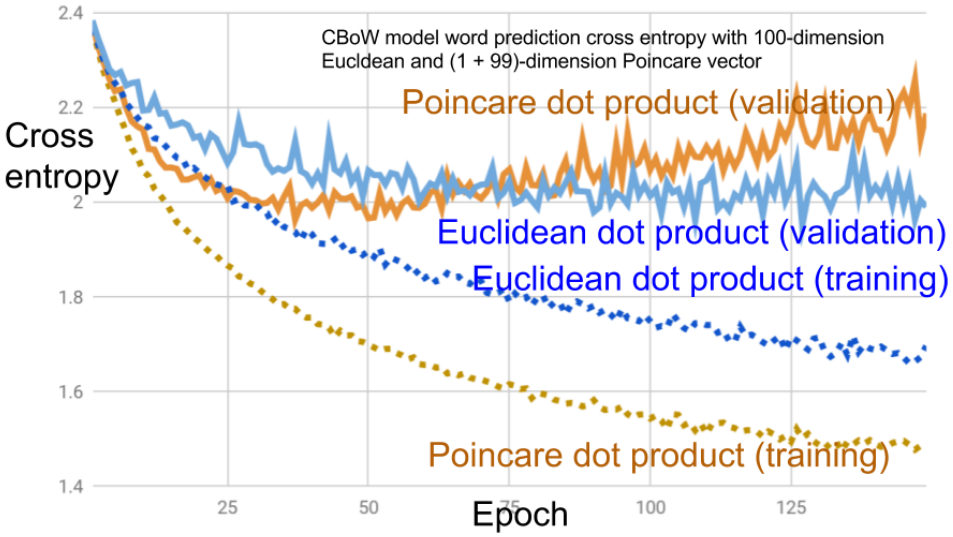

図 3　100 次元の埋め込みベクトルによる、CBoW での 1 単語の予測確率のクロスエントロピーの比較。

一方で、次元数が大きいときは、精度を向上させないようである。これは、Nickel らの報告において次元数が大きい場合には、Euclid 距離によるモデル (translational) と Poincare 円盤距離によるモデル (Poincare) との精度差が小さくなることと同じ理由であると考えられる。

5　結論

Poincare 円盤に内積に相当する量を導入することで、Poincare Embedding においてもテキストから埋め込みベクトルを学習することができる。これらは、低次元においては階層的関係の情報が標準的な Euclid モデルよりよく抽出されているとみることができるだろう。

参考文献

[1] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S

Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[3] Maximilian Nickel and Douwe Kiela. Poincare embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 2017.