確率分布に基づいた言語埋め込みモデル

Distribution based word embedding models

数学専攻　大竹 梓月
OHTAKE, Shizuki

1　はじめに

近年，人工知能の発展に伴い，その要素技術の 1 つである機械学習が多くの分野で活用されている．機械学習の目的の 1 つは，データからパターンを抽出し，それらのパターンを用いて予測を行うことである．学習を行う際の入力データの特徴は，学習パターンの性質や品質に大きな影響を及ぼすため，特徴を反映している入力データが重要な役割を果たす．入力データが既に実数値として表現されている場合は問題はないが，画像やテキスト，音声データのように単純に表現することができないものもある．これらの構造化されていない入力データを，特徴をもつ低次元のベクトルに落とし込むことを特徴埋め込みといい，また特にテキストデータにおけるこの低次元に埋め込まれたベクトルを単語の分散表現と呼ぶ．

自然言語処理の分野では，行列分解によって各単語の分散表現を獲得する潜在意味解析という手法から始まり，近年では Schwenk [2] が提案したニューラルネットワークを用いて単語をベクトルに埋め込むモデルであるニューラルネットワーク言語モデル (Neural Network Language Model, NNLM) [2] など数多くの研究が行われている．単語を分散表現として表すことにより，単語同士の類似度を測ったり，単語の加算や減算を行ったりすることができるようになる．一方で，単語の意味の広がりを捉えることができず，単語同士の意味の重なりや包含関係をこれらのモデルでは考慮することができないという欠点がある．

以上の課題を解決するため近年提案されているモデルが word2gauss [3] である．このモデルでは各単語を分散表現として埋め込むのではなく，埋め込み空間上に確率分布として埋め込むことを目的としている．各単語を確率分布として埋め込むことにより，その平均ベクトルを NNLM などで得られる分散表現として見ることができ，また分散を単語の意味の広がりとして捉えることができる．

本研究では，単語を分散表現として埋め込む 12 個のモデルと確率分布として埋め込む 3 個のモデルについて整理し，さらに word2gauss モデル [3] と word2gm [1] モデルの目的関数を変更したモデルについて検討する．

2　単語分散表現の埋め込み

単語の分散表現を学習するための手法は多く提案されており，本論文では 12 個のモデルについて整理した．ここでは，ニューラルネットワークを用いた手法である NNLM [2] を説明する．

2.1　NNLM: Neural Network Language Model

NNLM では，着目したある単語をその前に出現する $n-1$ 個の単語から予測することを基本的な考え方としており，その予測を入力層，隠れ層，出力層の 3 層のニューラルネットワークを用いて行う．通常のニューラルネットワークと異なる点は投影層の存在である．投影層は，入力層の $n-1$ 個のベクトルに重みを掛け，それを結合したものを出力する層である．

入力層のベクトルを $\boldsymbol{w}(j-n+1) = (w_1(j-n+1),\ldots,w_I(j-n+1))^{\mathrm{T}},\ldots,\boldsymbol{w}(j-1) = (w_1(j-1),...,w_I(j-1))^{\mathrm{T}}$，投影層のベクトルを $\boldsymbol{c} = (c_1,\ldots,c_{(n-1)P})^{\mathrm{T}}$，隠れ層の入力ベクトルを $\boldsymbol{d} = (d_1,\ldots,d_J)^{\mathrm{T}}$，

隠れ層の出力ベクトルを $\boldsymbol{h} = (h_1, \ldots, h_J)^{\mathrm{T}}$，出力層の入力ベクトルを $\boldsymbol{o} = (o_1, \ldots, o_I)^{\mathrm{T}}$，出力層の出力ベクトルを $\boldsymbol{y} = (y_1, \ldots, y_I)^{\mathrm{T}}$ とする．ここで入力ベクトルは局所表現，すなわち 1 単語に 1 次元を割り当てるベクトル表現であり，出力層の出力ベクトルは全ての単語に対する出現確率を表す．入力層と投影層間の重みを $\boldsymbol{C}(P \times I$ 行列)，投影層から隠れ層への重みを $\boldsymbol{M}(J \times L$ 行列)，隠れ層と出力層間の重みを $\boldsymbol{V}(I \times J$ 行列) と表記する．すなわち，それぞれの重み行列は

$$\boldsymbol{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1I} \\ c_{21} & c_{22} & \cdots & c_{2I} \\ \vdots & \vdots & \ddots & \vdots \\ c_{P1} & c_{P2} & \cdots & c_{PI} \end{pmatrix}, \quad \boldsymbol{M} = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1L} \\ m_{21} & m_{22} & \cdots & m_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ m_{J1} & m_{J2} & \cdots & m_{JL} \end{pmatrix}, \quad \boldsymbol{V} = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1J} \\ v_{21} & v_{22} & \cdots & v_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ v_{I1} & v_{I2} & \cdots & v_{IJ} \end{pmatrix}$$

のように表される．また，隠れ層と出力層の活性化関数をそれぞれ f，g とする．

まず投影層では，ある $n-1$ 個の入力ベクトルが与えられた際に，それらの入力ベクトルを 1 つのベクトルに表現し直す操作をする．はじめに $n-1$ 個の各入力ベクトル $\boldsymbol{w}(k)$ に対し，投影行列 $\boldsymbol{C}$ を掛け

$$\boldsymbol{c}(k) = \boldsymbol{C}\boldsymbol{w}(k) \tag{2.1}$$

を得る．(2.1) 式で計算された $n-1$ 個の P 次元ベクトル $\boldsymbol{c}(k)(k = 1, \ldots, n-1)$ をそのまま縦に並べ

$$\boldsymbol{c} = (\boldsymbol{c}(1), \boldsymbol{c}(2), \ldots, \boldsymbol{c}(n-1))^{\mathrm{T}} \tag{2.2}$$

を獲得する．この $(n-1)P$ 次元のベクトル $\boldsymbol{c}$ が投影層ベクトルである．以降，$(n-1)P$ を L として扱う．

続いて，隠れ層の入力 $\boldsymbol{d}$ と出力 $\boldsymbol{h}$ は投影層ベクトル $\boldsymbol{c}$ と活性化関数 f を用いて

$$\boldsymbol{d} = \boldsymbol{M}\boldsymbol{c} + \boldsymbol{b}, \quad \boldsymbol{h} = f(\boldsymbol{d}) \tag{2.3}$$

と計算される．ここで $\boldsymbol{b}$ はバイアスと呼ばれる定数ベクトルであり，回帰分析における切片項と同様の役割である．

最後に，出力層の入力 $\boldsymbol{o}$ と出力 $\boldsymbol{y}$ は隠れ層の出力 $\boldsymbol{h}$ と活性化関数 g から

$$\boldsymbol{o} = \boldsymbol{V}\boldsymbol{h} + \boldsymbol{k}, \quad \boldsymbol{y} = g(\boldsymbol{o}) \tag{2.4}$$

と計算される．ここでの $\boldsymbol{k}$ もバイアスを表す．ここで用いた活性化関数 f，g は，それぞれ

$$f(z) = \tanh(z), \quad g(z_m) = \frac{\exp(z_m)}{\sum_k \exp(z_k)} \tag{2.5}$$

である．

以上のような順伝播計算により，最終的な出力層，つまり全ての単語に対する出現確率を並べたベクトルが得られる．学習では出力層と正解データとの交差エントロピー誤差による目的関数からパラメータに関する勾配を計算し，誤差逆伝播法を用いて勾配を更新する．このフレームワークでは獲得したい単語分散表現は $\boldsymbol{C}$ の重み行列にあり，その各列が各単語の分散表現に対応している．

3 単語表現の確率分布への埋め込み

潜在意味解析や NNLM では，単語の意味の広がりや単語同士の包含関係を見ることができないという欠点があった．そこで，ガウス分布に単語を埋め込む word2gauss モデル [3] や混合ガウス分布へ埋め込む word2gm モデル [1] が提案されている．3.1 節，3.2 節でそれぞれのモデルについて説明し，3.3 節ではその 2 つのモデルを拡張し，目的関数を変更したモデルについて検討する．

3.1 word2gauss: Word Representations via Gaussian Embedding

Vilnis and McMallum [3] により提案された word2gauss モデル [3] は，NNLM などの埋め込みモデルの欠点であった意味の広がりを考慮したモデルである．word2gauss モデルでは単語をベクトルとして埋め込むのではなく，潜在的な関数，すなわち潜在空間における連続的な密度に埋め込むことを提案している．このように単語を無限次元の関数空間におけるガウス分布として直接埋め込むことにより，空間内の領域にマッピングできるので包含関係も捉えることが可能である．

このモデルの目標としては，各単語 w_i に対して埋め込み空間上の適切なガウス分布

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{3.1}$$

を割り当てることである．ここで，各単語のガウス分布の平均 $\boldsymbol{\mu}_i$ は NNLM における分散表現に相当するもので，共分散行列 $\boldsymbol{\Sigma}_i$ が埋め込み空間上での分散を表すものとなっている．似た文脈に出現する単語に対しては似た意味をもつようなパラメータ $\boldsymbol{\mu}_i$，$\boldsymbol{\Sigma}_i$ を獲得することが目的となる．

2 つのパラメータ $\boldsymbol{\mu}_i$，$\boldsymbol{\Sigma}_i$ の学習を行うために，パラメータ $\boldsymbol{\theta}$ のもとで単語 x と y の類似度を測るエネルギー関数 $E_{\boldsymbol{\theta}}(x, y)$ を導入して目的関数に用いる．注目している単語を w とし，w と同一の文脈に出現する文脈語を c_p，同一の文脈に出現しない単語を c_n としたときに

$$E_\theta(w, c_p) > E_\theta(w, c_n) \tag{3.2}$$

となるようなパラメータ $\boldsymbol{\theta}$，つまり $\boldsymbol{\mu}_i$ と $\boldsymbol{\Sigma}_i$ を学習によって獲得したい．エネルギーベースの学習の目的としては，観測された正の入出力ペアに負のペアよりも高いスコアをつけるようにエネルギー関数のパラメータを学習することである．そのため，目的関数として

$$L_m(w, c_p, c_n) = \max(0, m - E(w, c_p) + E(w, c_n)) \tag{3.3}$$

を用いる．これは max-margin ranking loss と呼ばれるもので，この L を最小化するように学習を行っていく．ここで m はマージンを表し，$E(w, c_p)$ を $E(w, c_n)$ よりも m 以上大きくとることを意図している．

エネルギー関数 E としては，2 つの分布間の距離を測る尺度である期待尤度や KL ダイバージェンスを用いることが提案されている．それぞれのエネルギー関数を用いて目的関数を計算し，各パラメータ $\boldsymbol{\mu}_i$，$\boldsymbol{\Sigma}_i$ で偏微分することにより勾配が求まり，勾配降下法などにより最適なパラメータが導出される．

3.2 word2gm

word2gauss モデルでは単語を 1 つの山の形でしか表現することができないので，この表現で学習された不確実性は複数の異なる意味をもつ単語，つまり多義語に対して過度に拡散してしまう可能性がある．そこで Athiwaratkun and Wilson [1] は複数の異なる意味をもつ単語の不確実性，及び解釈性の向上のために混合ガウス分布で各単語を表現する word2gm モデル [1] を提案した．

このモデルの目標としては，各単語 w に対して埋め込み空間上の適切な混合ガウス分布

$$\sum_{i=1}^{K} p_{w,i} \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{w,i}, \boldsymbol{\Sigma}_{w,i}) \tag{3.4}$$

を割り当てることである．ここで K は混合要素数であり，ここでは 1 つの単語に与える意味の数を表し，通常は 2 や 3 を指定する．また，$\sum_{i=1}^{K} p_{w,i} = 1$ である．word2gauss と同様に，エネルギー関数には期待尤度，損失関数としては max-margin ranking loss を用いることでパラメータの更新を行う．

3.3 L_2 ダイバージェンスを用いた word2gauss モデルと word2gm モデル

3.1 節，3.2 節のモデルに用いるエネルギー関数としては，word2gauss モデルでは期待尤度や KL ダイバージェンス，word2gm モデルでは期待尤度のみを使用していた．word2gm モデルにおいて KL ダイバージェンスを用いていないのは，エネルギー関数を閉じた形で表現できず，勾配計算によるパラメータ推定が困難となるためである．エネルギー関数としては 2 つの分布の近さを測るものを選択すれば良いため，他のダイバージェンスに置き換えることも可能である．ここでは，word2gm モデルにおいてもエネルギー関数を閉じた形として表現できるよう L_2 ダイバージェンスを用いた場合の word2gauss モデルと word2gm モデルについて検討する．

エネルギー関数は

$$E(f,g) = -D_{L_2}(f||g) = -\int_{\boldsymbol{x}\in\mathbb{R}^n} (f(\boldsymbol{x}) - g(\boldsymbol{x}))^2 d\boldsymbol{x} \tag{3.5}$$

で与えられ，$f(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$，$g(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ とすると，word2gauss モデルのエネルギー関数は

$$E(f,g) = -\mathcal{N}(\mathbf{0}; \mathbf{0}, 2\boldsymbol{\Sigma}_i) - \mathcal{N}(\mathbf{0}; \mathbf{0}, 2\boldsymbol{\Sigma}_j) + 2\mathcal{N}(\mathbf{0}; \boldsymbol{\mu}_i - \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j) \tag{3.6}$$

と求まる．同様に，$f(\boldsymbol{x}) = \sum_{i=1}^K p_i \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{f,i}, \boldsymbol{\Sigma}_{f,i})$，$g(\boldsymbol{x}) = \sum_{j=1}^K q_j \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{g,j}, \boldsymbol{\Sigma}_{g,j})$ とし，word2gm モデルのエネルギー関数を計算すると

$$\begin{aligned} E(f,g) = &-\sum_{i=1}^K \sum_{j=1}^K p_i p_j \mathcal{N}(\mathbf{0}; \boldsymbol{\mu}_{f,i} - \boldsymbol{\mu}_{f,j}, \boldsymbol{\Sigma}_{f,i} + \boldsymbol{\Sigma}_{f,j}) - \sum_{i=1}^K \sum_{j=1}^K q_i q_j \mathcal{N}(\mathbf{0}; \boldsymbol{\mu}_{g,i} - \boldsymbol{\mu}_{g,j}, \boldsymbol{\Sigma}_{g,i} + \boldsymbol{\Sigma}_{g,j}) \\ &+ 2\sum_{i=1}^K \sum_{j=1}^K p_i q_j \mathcal{N}(\mathbf{0}; \boldsymbol{\mu}_{f,i} - \boldsymbol{\mu}_{g,j}, \boldsymbol{\Sigma}_{f,i} + \boldsymbol{\Sigma}_{g,j}) \end{aligned} \tag{3.7}$$

と求まり, word2gm モデルにおいても閉じた形で表現できることがわかる．ここで, $\sum_{i=1}^K p_i = 1$, $\sum_{j=1}^K q_j = 1$ である．このエネルギー関数を用いて勾配を計算することで，最適なパラメータが導出される．

4 おわりに

本研究では，自然言語処理に焦点を当て単語の特徴埋め込みモデルを整理し，L_2 ダイバージェンスを用いたモデルを検討した．今後の研究課題として次の 2 つをあげる．1 つ目は，ガウス埋め込みのさらなる拡張である．本稿では単純にエネルギー関数の変更を行なったが，ガウス分布を他の分布，例えば楕円分布に置き換える方法も考えられる．2 つ目は，ダイバージェンスを変更したことによる影響や，L_2 ダイバージェンスを用いた場合の埋め込みの特徴がどのようなものであるかを調べることである．これら 2 点が今後の研究課題である．

参考文献

[1] Athiwaratkun, B. and Wilson, A. G. (2019) Multimodal word distributions, In *Proceeding of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, 1645-1656.

[2] Schwenk, H. (2007) Continuous space language models, In *Computer Speech and Language 21*, 492-518.

[3] Vilnis, L. and McCallum, A. (2015) Word representations via gaussian embedding, In *Published as a conference paper at ICLR 2015.*