Skip-gram Model with Negative Sampling のオンライン逐次学習

鍛治伸裕　小林隼人
ヤフー株式会社

{nkaji, hakobaya}@yahoo-corp.jp

1　はじめに

近年，単語分散表現の学習に関する研究が大きな成果を挙げている [4, 7]．しかしながら，既存の分散表現学習法はバッチ学習に基づくものが主流であるため，マイクロブログや検索クエリなど日々増加し続けるデータから効率的に学習を行うことが難しい．訓練データを大規模化することは分散表現の品質向上に直結するため [2]，新しい訓練データが入手可能となった場合には，古い訓練データと新しい訓練データを合わせたより大きな訓練データから，分散表現を学習しなおすことが望ましいと考えられる．しかし，そうするためには，バッチ学習にもとづく既存の分散表現学習法は，訓練データが追加されるたびにモデルを一から再学習しなくてはならず非効率である．これに加え，再学習を行うためには古い訓練データを全て保持しておかなくてはならないため，その余分な記憶領域が必要となってしまう．

本論文では，代表的な分散表現学習法である skip-gram model with negative sampling (SGNS)[4] を逐次更新可能なオンラインアルゴリズムに拡張し，その性能について理論的な解析を行う．さらに，提案するオンライン学習法が，従来の SGNS と同等精度の分散表現を学習可能であることを実証的に示す．

2　SGNS

始めに本研究の基礎となる SGNS について簡単に説明を行う [4]．

Skip-gram model は，単語列 $w_1, w_2, \ldots, w_n$ を訓練データとし，以下の目的関数を最小化することによって分散表現の学習を行う．

$$\mathcal{L}_{\mathrm{SG}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{\substack{|j|\le c\\ j\neq 0}} \log p(w_{i+j} \mid w_i),$$

ここで $p(w_{i+j} \mid w_i)$ は，単語 w_i の分散表現 $\mathbf{t}_{w_i}$ と文脈語 w_{i+j} の分散表現 $\mathbf{c}_{w_{i+j}}$ を用いて

$$p(w_{i+j} \mid w_i) = \frac{\exp(\mathbf{t}_{w_i} \cdot \mathbf{c}_{w_{i+j}})}{\sum_{w\in\mathcal{W}} \exp(\mathbf{t}_{w_i} \cdot \mathbf{c}_w)}$$

と表現される．ただし $\mathcal{W}$ は語彙集合である．

SGNS は上記の損失関数を以下のような形で近似することにより，分散表現の学習を高速化させる．

$$\mathcal{L}_{\mathrm{SGNS}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{\substack{|j|\le c\\ j\neq 0}} \psi^{+}_{w_i,w_{i+j}} + k\mathbb{E}_{v\sim q(v)}[\psi^{-}_{w_i,v}]$$

ただし，$\psi^{+}_{w,v} = \log\sigma(\mathbf{t}_w\cdot\mathbf{c}_v)$，$\psi^{-}_{w,v} = \log\sigma(-\mathbf{t}_w\cdot\mathbf{c}_v)$，$\sigma(x)$ はシグモイド関数である．$q(v)$ は雑音分布と呼ばれる単語分布であり，$f(v)$ を訓練データ中の単語 v の頻度，α を $0 < \alpha \le 1$ を満たすパラメータとすると，$q(v) \propto f(v)^{\alpha}$ で与えられる．

SGNS は確率的勾配法によって学習を行う．すなわち，単語 w_i，文脈語 w_{i+j}，雑音分布 $q(v)$ からサンプリングした k 個の（擬似的な）負例語 $(v_1, v_2 \ldots v_k)$ を用いて $\mathcal{L}_{SGNS}$ の勾配を近似的に求める．そして勾配降下法によって各々の分散表現の更新を行う．

確率的勾配法は，一般的にはオンライン学習の範疇に分類することができるが，SGNS の場合はそうではない．これは雑音分布が訓練データ中の単語頻度 $f(v)$ に依存しているためである．このため，勾配計算を行うためには，訓練データ全体から雑音分布を事前に計算しておく必要があり，オンライン処理による逐次的なモデル更新は難しい．

3　オンライン SGNS

本研究では，SGNS をオンライン学習可能な形に拡張した，オンライン SGNS を提案する（Algorithm 1）．オンライン SGNS では，雑音分布を前計算するので

はなく，単語を読み込むたびに雑音分布を逐次更新しながら，勾配計算に必要な負例語のサンプリングを行う．以下では，提案するオンライン SGNS との区別を明確にするため，Mikolov らによる本来の SGNS[4] のことはバッチ SGNS と呼ぶ．

オンライン SGNS のアイデア自体は単純なものであるが，実際に効率的な学習を実現するためには，以下のような工夫が必要となる．

3.1 動的語彙

オンライン学習下では語彙集合を事前に決めておくことが難しい．そこで，データストリームからの頻出アイテム集合発見アルゴリズムである Misra-Gries アルゴリズム [5] を使って，頻出単語を近似的に列挙し，それらを動的に変化する語彙集合として用いる．

3.2 適応的ユニグラム表

Negative sampling にもとづく学習においては，勾配計算のたびに，雑音分布から負例語を k 個サンプリングする必要がある．そのため，効率的な学習を実現するためには，このサンプリング処理を高速に行う必要がある。

バッチ SGNS の場合，ユニグラム表と呼ばれるサイズ τ の配列 T ($|T| = \tau$) に単語 w をそれぞれ $q(w) \times \tau$ 個格納しておき，その配列要素をランダムサンプリングすることによって，高速な負例語のサンプリングが実現できる [3]．一方，オンライン SGNS の場合に同様の方法を使うためには，単語を読み込むたびに雑音分布が更新されるため，ユニグラム表 T も更新しなくてはならない．そこで，重み付きデータストリームからのサンプリングアルゴリズムである weighted reservoir sampling アルゴリズム [1] を用いる (Algorithm 2)．単語 w_i の重みを $F(w_i) = f(w_i)^\alpha - (f(w_i) - 1)^\alpha$ (3 行目) とすれば，ユニグラム表の逐次更新処理は，単語列からの重み付き復元サンプリングとみなすことができるため，weighted reservoir sampling の適用が可能になる．ただし，8 から 10 行目の反復処理はそのまま実装すると非効率なので，ランダムに選択した $\frac{\tau F(w_i)}{z}$ 個の配列要素に w_i を代入するという処理で近似する．

Algorithm 1 オンライン SGNS

```
1: for i = 1, ..., n do
2:     f(w_i) ← f(w_i) + 1
3:     q(w) ← f(w)^α / Σ_{w'∈W} f(w')^α  for all w ∈ W
4:     for j = −c, ..., −1, 1, ..., c do
5:         雑音分布 q(w) から負例語 v_1, ..., v_k をサンプリング
6:         確率的勾配法により t_{w_i}, c_{w_{i+j}}, c_{v_1}, ..., c_{v_k} を更新
7:     end for
8: end for
```

Algorithm 2 適応的ユニグラム表

```
1: for i = 1, ..., n do
2:     f(w_i) ← f(w_i) + 1
3:     F(w_i) ← f(w_i)^α − (f(w_i) − 1)^α
4:     z ← z + F(w_i)
5:     if |T| < τ then
6:         T に w_i を F(w_i) 個追加
7:     else
8:         for j = 1, ..., τ do
9:             F(w_i)/z の確率で T[j] に w_i を代入
10:        end for
11:    end if+
12: end for
```

4 理論的解析

オンライン SGNS は，各ステップにおいて，$-\psi^+_{w_i,w_{i+j}} - k\mathbb{E}_{v\sim q_i(v)}[\psi^-_{w_i,v}]$ の勾配計算を行っている．ただし $q_i(w)$ は，訓練データの先頭 i 単語から求めた雑音分布である ($q(v) = q_n(v)$ を満たす)．これは

$$\mathcal{L}_{\mathrm{O}}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{\substack{|j|\le c\\ j\neq 0}} \psi^+_{w_i,w_{i+j}} + k\mathbb{E}_{v\sim q_i(v)}[\psi^-_{w_i,v}]$$

の勾配のサンプル近似とみなせる．ただし θ はモデルのパラメータ（分散表現の集合）である．従ってオンライン SGNS は確率的勾配法によって損失関数 $\mathcal{L}_{\mathrm{O}}$ を最小化していると解釈できる．

ここで，2 節で説明したバッチ SGNS の損失関数を $\mathcal{L}_B(\theta)$ と書き直し，オンライン SGNS の損失関数との差分を $\Delta\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{B}}(\theta) - \mathcal{L}_{\mathrm{O}}(\theta)$ とする．これは次のように書き換えられる．

$$\begin{aligned}
\Delta\mathcal{L}(\theta) &= \mathcal{L}_{\mathrm{B}}(\theta) - \mathcal{L}_{\mathrm{O}}(\theta)\\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{\substack{-c\le j\le c\\ j\neq 0}} k\sum_{v\in\mathcal{W}}(q_i(v) - q_n(v))\psi^-_{w_i,v}\\
&= \frac{2ck}{n}\sum_{i=1}^{n}\sum_{v\in\mathcal{W}}(q_i(v) - q_n(v))\psi^-_{w_i,v}\\
&= \frac{2ck}{n}\sum_{w\in\mathcal{W}}\sum_{v\in\mathcal{W}}\sum_{i=1}^{n}\delta_{w_i,w}(q_i(v) - q_n(v))\psi^-_{w,v}
\end{aligned}$$

以下では，雑音分布のパラメータを $\alpha = 1.0$ と仮定し，$\Delta\mathcal{L}(\theta)$ の確率的性質を分析することによって，バッチ SGNS とオンライン SGNS の最適解の関係性を理論的に明らかにする[1]．

[1] パラメータが $\alpha \neq 1.0$ となる場合でも，雑音分布 $q_i(v)$ を一次の Taylor 展開を使って近似することにより同様に議論できるが，紙面の都合により省略する．

$\Delta\mathcal{L}(\theta)$ の値は $q_i(v)$ に依存しているためこのままでは定量的に議論することが難しい．そこで訓練データ中の単語がユニグラムモデル $\mu = (\mu_1, \mu_2, \ldots, \mu_{|\mathcal{W}|})$ から生成されていると仮定する[2]．$q_i(v)$ はそもそも単語ユニグラムの頻度にもとづき定義されているので，理論解析のためにこのような仮定をおくことは妥当なことである．このように仮定のもと，$\delta_{w_i,w}$ を表す確率変数を $\mathrm{X}_{i,w}$ とすると，その期待値は，$\mathbb{E}[\mathrm{X}_{i,w}] = \mu_w$ となる．一方，共分散は，$i \neq j$ のとき $\mathbb{V}[\mathrm{X}_{i,w}\mathrm{X}_{j,v}] = 0$，それ以外のとき $\mathbb{V}[\mathrm{X}_{i,w}\mathrm{X}_{j,v}] = \rho_{w,v}$ となる．ただし

$$\rho_{w,v} = \begin{cases} \mu_w(1-\mu_w) & (w = v) \\ -\mu_w\mu_v & (w \neq v) \end{cases}$$

である．また，$q_i(w)$ を表す確率変数を $\mathrm{Y}_{i,w}$ とすると，$\mathrm{Y}_{i,w} = \frac{1}{i}\sum_{i'=1}^{i}\mathrm{X}_{i',w}$ となる．

このとき $\Delta\mathcal{L}(\theta)$ の一次モーメントと二次モーメントに関して，以下の 3 つの定理が得られる．

定理 1. $\Delta\mathcal{L}(\theta)$ の一次モーメントは

$$\mathbb{E}[\Delta\mathcal{L}(\theta)] = \frac{2ck(H_n - 1)}{n}\sum_{w\in\mathcal{W}}\sum_{v\in\mathcal{W}}\rho_{w,v}\psi^{-}_{w,v}$$

となる．ただし H_n は n 番目の調和数である．

証明. $\Delta\mathcal{L}(\theta)$ の一次モーメントは

$$\frac{2ck}{n}\sum_{w\in\mathcal{W}}\sum_{v\in\mathcal{W}}\sum_{i=1}^{n}\Big(\mathbb{E}[\mathrm{X}_{i,w}\mathrm{Y}_{i,v}] - \mathbb{E}[\mathrm{X}_{i,w}\mathrm{Y}_{n,v}]\Big)\psi^{-}_{w,v}$$

で表されるが，$i \leq j$ を満たす任意の i と j について

$$\mathbb{E}[\mathrm{X}_{i,w}\mathrm{Y}_{j,v}] = \frac{1}{j}\sum_{j'=1}^{j}\mathbb{E}[\mathrm{X}_{i,w}\mathrm{X}_{j',v}] = \mu_w\mu_v + \frac{1}{j}\rho_{w,v}$$

となるので，これを代入すれば定理が証明できる． □

定理 2. $\Delta\mathcal{L}(\theta)$ の一次モーメントのオーダは

$$\mathbb{E}[\Delta\mathcal{L}(\theta)] = \mathcal{O}\Big(\frac{\log(n)}{n}\Big)$$

であり，$n \to \infty$ のとき 0 に収束する．

$$\lim_{n\to\infty}\mathbb{E}[\Delta\mathcal{L}(\theta)] = 0$$

証明. $\log(x)$ の上積分より $H_n = \mathcal{O}(\log(n))$ が得られる．従って定理 1 により証明終了． □

定理 3. $\Delta\mathcal{L}(\theta)$ の二次モーメントのオーダは

$$\mathbb{E}[\Delta\mathcal{L}(\theta)^2] = \mathcal{O}\Big(\frac{\log(n)}{n}\Big)$$

であり，$n \to \infty$ のとき 0 に収束する．

$$\lim_{n\to\infty}\mathbb{E}[\Delta\mathcal{L}(\theta)^2] = 0$$

[2]添え字の自然数は単語に対応しているものとする．

証明. 紙面の都合により省略． □

ここまでの議論から以下の補題を導くことができる．

補題 4. $\mathcal{L}_B(\theta)$，$\mathcal{L}_O(\theta)$ の最適解を各々 θ^*，$\hat{\theta}$ とすると，以下の式が成り立つ．

$$\lim_{n\to\infty}\mathbb{E}[\mathcal{L}_B(\hat{\theta}) - \mathcal{L}_B(\theta^*)] = 0 \tag{1}$$

$$\lim_{n\to\infty}\mathbb{V}[\mathcal{L}_B(\hat{\theta}) - \mathcal{L}_B(\theta^*)] = 0 \tag{2}$$

証明. 表記を簡単にするため以下では $l = \mathcal{L}_\mathrm{B}(\hat{\theta}) - \mathcal{L}_\mathrm{B}(\theta^*)$ とおく．すると θ^* の最適性より $0 \leq l$ が成り立つ．また $\hat{\theta}$ の最適性より

$$\begin{aligned} l &= \mathcal{L}_\mathrm{B}(\hat{\theta}) - \mathcal{L}_\mathrm{O}(\theta^*) + \mathcal{L}_\mathrm{O}(\theta^*) - \mathcal{L}_\mathrm{B}(\theta^*) \\ &\leq \mathcal{L}_\mathrm{B}(\hat{\theta}) - \mathcal{L}_\mathrm{O}(\hat{\theta}) + \mathcal{L}_\mathrm{O}(\theta^*) - \mathcal{L}_\mathrm{B}(\theta^*) \\ &= \Delta\mathcal{L}(\hat{\theta}) - \Delta\mathcal{L}(\theta^*) \end{aligned} \tag{3}$$

が成り立つ．このとき，定理 2 により，式 (3) は $n \to \infty$ のとき 0 に収束することが分かる．従って，はさみうちの原理から式 (1) が得られる．一方，

$$\begin{aligned} \mathbb{V}[l] &= \mathbb{E}[l^2] - \mathbb{E}[l]^2 \leq \mathbb{E}[l^2] \leq \mathbb{E}[(\Delta\mathcal{L}(\hat{\theta}) - \Delta\mathcal{L}(\theta^*))^2] \\ &\leq \mathbb{E}[(\Delta\mathcal{L}(\hat{\theta}) - \Delta\mathcal{L}(\theta^*))^2 + (\Delta\mathcal{L}(\hat{\theta}) + \Delta\mathcal{L}(\theta^*))^2] \\ &= 2\mathbb{E}[\Delta\mathcal{L}(\hat{\theta})^2] + 2\mathbb{E}[\Delta\mathcal{L}(\theta^*)^2] \end{aligned} \tag{4}$$

が成り立つが，定理 3 より式 (4) は $n \to \infty$ のとき 0 に収束することが分かる．従って，分散の非負性と、はさみうちの原理から式 (2) が得られる． □

ここまでの結果から，バッチ SGNS とオンライン SGNS の最適解に関する以下の定理が得られる．この定理は，(不正確であるが) 大雑把に言うと，訓練データを十分に大きくすることによって，オンライン SGNS の最適解をバッチ SGNS の最適解に限りなく近づけることができることを意味している．

定理 5. $\mathcal{L}_B(\hat{\theta})$ は $\mathcal{L}_B(\theta^*)$ に確率収束する*:*

$$\forall\epsilon > 0, \lim_{n\to\infty}\Pr\Big[|\mathcal{L}_B(\hat{\theta}) - \mathcal{L}_B(\theta^*)| \geq \epsilon\Big] = 0.$$

証明. チェビシェフの不等式から，任意の $\epsilon_1 > 0$ に対して以下の不等式が成り立つ．

$$\lim_{n\to\infty}\frac{\mathbb{V}[l]}{\epsilon_1^2} \geq \lim_{n\to\infty}\Pr\Big[|l - \mathbb{E}[l]| \geq \epsilon_1\Big]$$

ここで，式 (1) は「任意の $\epsilon_2 > 0$ に対して，$n' \leq n$ ならば $|\mathbb{E}[l]| < \epsilon_2$ を満たす n' が存在する」ことを意味していることに注意する．このことを利用すると，上記の不等式から以下が導かれる．

$$\lim_{n\to\infty}\frac{\mathbb{V}[l]}{\epsilon_1^2} \geq \lim_{n\to\infty}\Pr\Big[|l| \geq \epsilon_1 + \epsilon_2\Big] \geq 0$$

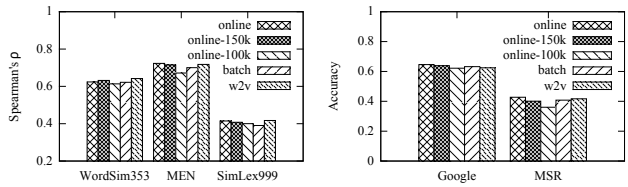

図 1: Word Similarity タスク（左図）と Analogy タスク（右図）の結果.

このとき，ϵ_1 と ϵ_2 の任意性により $\epsilon_1 + \epsilon_2$ を ϵ と置き換えてもよい．さらに，式 (2) より，$\lim_{n\to\infty} \frac{\mathbb{V}[l]}{\epsilon_1^2} = 0$ となる．これらのことから，はさみうちの原理によって定理が証明できる． □

5 実験

オンライン SGNS とバッチ SGNS が学習した分散表現の比較を行った．訓練データには English Gigaword Corpus [6] を用いた．評価用データには，過去の研究 [2] でも用いられている，Word Similarity タスクのベンチマークを 3 種類（WordSim353，MEN，SimLex999），Analogy タスクのベンチマークを 2 種類（Google データと MSR データ) 用いた．

図 1 に全 5 種類のベンチマークでの評価結果を示す．比較対象のバッチ SGNS は，独自実装による結果（batch）と Mikolov によって公開されているソフトウェア [4] による結果（w2v）の 2 種類を用いた．オンライン SGNS(online) と batch は 3 節で述べた動的語彙を用い，その最大サイズは $240k$ とした．w2v は頻度 100 以上の単語を語彙集合とした．この結果それぞれの語彙数は 220,389，246,134 と，ほぼ同程度になっている．また，動的語彙の有効性を検証するため，語彙の最大数を $150k$，$100k$ としたオンライン SGNS も比較に含めた (online-150k と online-100k）．

この図から，いずれのベンチマークにおいても，online は batch および w2v と同等の性能を達成していることが確認できる．このことから，オンライン SGNS は，学習される分散表現の品質を犠牲にすることなく，オンライン学習による逐次的なモデル更新を実現できていることが確認された．また，online-150k と online-100k の精度は，online と比べて一部のデータセットでは低下しているものの，全体として大きな劣化は見られないことから，動的語彙が語彙サイズを効果的に制御できていることが分かる。

6 おわりに

本論文はオンライン SGNS を提案し，その効率的な実装方法および理論解析の結果を示した．そして，オンライン SGNS が，オンライン処理による逐次的モデル更新を実現すると同時に，従来のバッチ学習法と同等の品質の分散表現を学習可能であることを実証的に示した．今後の課題としては，オンライン SGNS を使って，ウェブ検索クエリのような大規模かつ日々更新されるデータを簡潔に表現して利活用するための研究を行いたい．また，GloVe [7] など，SGNS 以外の分散表現学習法のオンラインアルゴリズム化も興味深い課題と考えている．

参考文献

[1] P.S. Efraimidis. Weighted random sampling over data streams. arXiv:1012.0256, 2015.

[2] O. Levy, Y. Goldberg, and I. Dagan. Improving distributional similarity with lessons learned from word embeddings. *TACL*, Vol. 3, pp. 211–225, 2015.

[3] T. Mikolov. word2vec. https://code.google.com/archive/p/word2vec, 2013.

[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in NIPS*, pp. 3111–3119, 2013.

[5] J. Misra and D. Gries. Finding repeated elements. *Science of Computer Programming*, Vol. 2, No. 2, pp. 143–152, 1982.

[6] C. Napoles, M. Gormley, and B. Van Durme. Annotated english gigaword ldc2012t21, 2012.

[7] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pp. 1532–1543, 2014.