On the Dimensionality of Word Embedding

0. Reference

- On the Dimensionality of Word Embedding, Zi Yin et al.

1. Introduction

행렬 분해를 활용한 LSA 워드 임베딩, word2vec, glove 등 다양한 단어 임베딩 기법이 NLP 분야에서 나오고 있다. 단어 임베딩의 중요한 hyperparameter 중 하나는 벡터의 차원(dimension)이다. 벡터의 차원이 커질수록 overfitting될 위험이 있고 작을수록 단어의 관계를 잘 파악하지 못할 것이다. 또한 벡터의 차원이 크다는 뜻은 학습해야 할 파라미터가 많다는 뜻인데, 이는 computation 문제와도 직결되므로 적절한 벡터의 차원을 정하는 것이 중요하다.
word2vec을 제안한 Mikolov의 논문에서 벡터가 300차원으로 정해졌기 때문에 많은 사람들이 300차원 벡터를 사용한다. 또는 여러 차원의 임베딩 벡터를 구하고, 각 임베딩 벡터 세트에 대해서 relatedness, word analogy 테스트 등을 통해 벡터의 품질을 파악하는 방법도 있다. 하지만 이 방법은 여러 차원의 벡터를 학습시키는 것이 시간이 오래 걸리고 모든 차원의 벡터에 대해서 성능을 평가하는 것이 불가능하다는 단점이 있다.
이러한 문제점에 대한 해결 방안으로, 단어 임베딩의 차원을 결정하는 방법을 본 논문에서 소개한다.
본격적으로 들어가기 전에 notation을 정의하자.

- $\mathcal{V} = \{1, 2, \cdots, n\}$, 즉 n개의 단어에 대한 임베딩 벡터를 학습하는 것이 목표이다.
- d차원의 벡터, $v_i \in \mathbb{R}^d$가 최종 결과물이다.
- v_i를 쌓아 올린 $E \in \mathbb{R}^{n \times d}$ 임베딩 행렬을 얻을 수 있다. 여기서 한 i번째 row는 i번째 임베딩 벡터를 의미한다; $E_{i,\cdot} = v_i$

2. Preliminaries and Background Knowledge

2.1 Unitary Invariance of Word Embeddings

단어 임베딩의 initary invariance은 아래와 같이 정의된다.

- It states that two embeddings are essentially identical if one can be obtained from the other by performing a unitary operation, e.g., a rotation.
- $U^T = U^{-1}$을 만족하는 행렬을 unitary matrix라고 한다.
- matrix norm $\mathbb{C}^{m,n}$에 대한 $||\ ||$은 아래의 조건을 만족하면 unitary invariant라고 부른다.

$$||UAV|| = ||A||,\ A \in \mathbb{C}^{m,n}, U \in \mathbb{C}^{m,m}, \in \mathbb{C}^{n,n}$$

- 벡터에 대해서 unitary operation을 한다는 것은 벡터에 unitary matrix을 곱한다는 뜻이다; $v^{'} = vU, where\ U^{'}U = UU^{'} = I \cdot d.$

- unitary 변환은 벡터의 relative geometry을 보존시켜줘서 임베딩의 equivalence class을 정의한다. 후에 소개할 PIP loss는 unitary-invariant metric이다.

2.2 Word Embeddings from Explicit Matrix Factorization

많은 임베딩 알고리즘은 explicit matrix factorization을 사용한다. 예를 들어 LSA는 PMI, PPMI 등의 행렬에 truncated SVD를 수행하여 단어 임베딩을 얻는다.
Levy 등은 signal matrix M을 $M = UDV^T$로 분해하여 $E = U_{1:k}D^{\alpha}_{1:k,1:k},\ \alpha \in [0,1]$을 통해 단어 임베딩을 얻는 방법을 설명했다. 본 논문에서 α는 임베딩의 robustness을 결정하는 모수임을 발견했고 이에 대해서는 section 5에서 살펴본다.

2.3 Word Embeddings from Implicit Matrix Factorization

단어 임베딩에서 많이 사용되는 기법인 word2vec과 glove는 objective function을 최소화하는 방법으로 학습이 되었지만 이후 Levy 등에 의해서 이 둘이 implicitly matrix factorization을 수행하고 있음이 밝혀졌다.[1]

3. PIP Loss: Novel Unitary-invariant Loss Function for Embeddings

어떤 것의 '좋고 나쁨'에 대해서 어떤 '기준'을 정해서 이야기하기 마련이다. 본 논문에서는 두 임베딩의 dissimilarity metric 으로 Pariwise Inner Product (PIP) loss를 제안한다. 이는 unitary-invariance에서 자연스럽게 생각할 수 있다.

$$PIP(E) = EE^T$$

PIP 행렬의 (i,j)번째 원소는 i,j번째 단어 임베딩의 내적임을 알 수 있다; $PIP_{i,j} =< v_i, v_j >$
두 단어 임베딩 행렬을 E_1, E_2라고 하자. 두 단어 임베딩의 PIP loss는 아래와 같이 정의된다.

$$||PIP(E_1) - PIP(E_2)|| = ||E_1E_1^T - E_2E_2^T|| = \sqrt{\sum_{i,j}(< v_i^{(1)}, v_j^{(1)} > - < v_i^{(2)}, v_j^{(2)} >)^2}$$

PIP 행렬의 i번째 행은 $v_iE^T = (< v_i, v_1 >, \cdots, < v_i, v_n >)$인데, 다른 모든 벡터 $v_1, \cdots, v_n$에 대해 anchor된 v_i의 상대적인 위치로 볼 수 있다.
PIP loss의 의미에 대해서 더 살펴보면 다음과 같다. PIP loss는 inner products간의 차이를 측정하므로 E_1, E_2간의 작은 PIP loss는 두 임베딩 벡터의 relatedness나 analogy의 차이가 작다는 뜻이다. 뿐만 아니라, PIP loss는 임베딩 차원을 이해하는 방법을 제시한다. 이를 section 4에서 살펴보자.

4. How Does Dimensionality Affect the Quality of Embedding?

이제 matrix factorization을 사용하는 임베딩 알고리즘의 품질을 PIP loss을 통해서 판별하는 방법에 대해 알아보자. d차원의 임베딩 벡터가 signal matrix M(예를 들어 PMI matrix)을 통해 도출되었

[1] Neural word embedding as implicit matrix factorization, Levy et Goldberg

다고 하자.

$$f_{\alpha,d}(M) \triangleq U_{\cdot,1:d} D^{\alpha}_{1:d,1:}, \; M = UDV^T$$

true M을 한다면 아주 쉽겠지만 그런 경우는 없으므로 M을 추정해야 한다; $\tilde{M}$(예를 들어 empirical PMI). 추정에 따른 오차를 Z라고 하면

$$\tilde{M} = M + Z$$

훈련된 임베딩은 이러한 noisy matrix을 분해한 것이다.

$$\hat{E} = f_{\alpha,k}(\tilde{M})$$

$\hat{E}$와 E가 '가깝다고' 확인하기 위해, PIP loss, $||EE^T - \hat{E}\hat{E}^T||$를 작게하고 싶을 것이다. 특히, 이 PIP loss는 선택하는 차원인 k에 영향을 받는다.

"*... A striking finding in empirical work on word embeddings is that there is a sweet spot for the dimensionality of word vectors: neither too small, nor too large*"

Arora [2016]는 위와 같이 말하면서 지나치게 크지도 않고 작지도 않은 적당한 벡터의 차원이 존재한다고 말했다. 본 논문에서는 이러한 현상을 bias-variance trade-off를 이용해서 보인다.

4.1 The Bias Variance Trade-off for a Special Case: $\alpha = 0$

다음 theorem은 $\alpha = 0$일 때, PIP loss가 bias term과 variance term으로 분해됨을 보여준다.

Theorem 1. Let $E \in \mathbb{R}^{n \times d}$ and $\hat{E} \in \mathbb{R}^{n \times k}$ be the oracle and trained embeddings, where $k \leq d$. Assume both hae orthonoral columns. Then the PIP loss has a bias-variance decomposition.

$$||PIP(E) - PIP(\hat{E})||^2 = d - k + 2||\hat{E}^T E^{\perp}||^2$$

$d-k$는 $k \leq d$의 singular values을 선택한 것에 따른 lost signal의 양이다. $||\hat{E}^T E^{\perp}||^2$은 k가 증가함에 따라서 같이 증가하는데, 큰 k는 noisy한 singular values나 vectors을 많이 포함한다는 뜻이고 그만큼 더 변동하기 때문이다. 따라서 PIP loss을 최소화하는 optimal dimensionality k^*는 0과 d사이에 있다.

4.2 The Bias Variance Trade-off for the Generic Case: $\alpha \in (0, 1]$

이제 더 이상 $E, \hat{E}$가 orthonormal한 경우가 아니다.

Theorem 2. Let $M = UDV^T, \tilde{M} = \tilde{U}\tilde{D}\tilde{V}^T$. Suppose $E = U_{\cdot,1:d} D^{\alpha}_{1:d,1:d}$ is the oracle embedding, and $\hat{E} = \tilde{U}_{\cdot,1:k} \tilde{D}^{\alpha}_{1:k,1:k}$ is the trained embedding, for some $k \leq d$. Let $D = diag(\lambda_i)$ and $\tilde{D} = diag(\tilde{\lambda}_i)$, then

$$\|\text{PIP}(E) - \text{PIP}(\hat{E})\| \leq \sqrt{\sum_{i=k+1}^{d} \lambda_i^{4\alpha}} + \sqrt{\sum_{i=1}^{k} (\lambda_i^{2\alpha} - \tilde{\lambda}_i^{2\alpha})^2} + \sqrt{2} \sum_{i=1}^{k} (\lambda_i^{2\alpha} - \lambda_{i+1}^{2\alpha}) \|\tilde{U}^T_{\cdot,1:i} U_{\cdot,i:n}\|$$

첫 번째 term은 k가 커질 수록 작아지는 term이므로 bias이다. embedding matrix E가 $E = U_{\cdot,1:d}D^{\alpha}_{1:d,1:d}$로 정의됐었다. 여기서 signal directions은 U에 의해, magnitudes는 D^{α}로 구성된다. 두 번째 term은 magnitudes에 대한 분산, 세 번째 term은 directions에 대한 분산이다.

4.3 The Bias Variance Trade-off Captures the Signal-to-Noise Ratio

이제 bias-variance trade-off가 'signal-to-noise ratio'을 반영함을 보여주는 main theorem을 살펴보자.

Theorem 3 (Main theorem). Suppose $\tilde{M} = M + Z$, where M is the signal matrix, symmetric with spectrum $\{\lambda_i\}_{i=1}^{d}$. Z is the estimation noise, symmetric with iid, zero mean, variance σ^2 entries. For any $0 \leq \alpha \leq 1$ and $k \leq d$, let the oracle and trained embeddings be

$$E = U_{\cdot,1:d}D^{\alpha}_{1:d,1:d},\ \hat{E} = \tilde{U}_{\cdot,1:k}\tilde{D}^{\alpha}_{1:k,1:k}$$

where $M = UDV^T,\ \tilde{M} = \tilde{U}\tilde{D}\tilde{V}^T$ are the SVDs of the clean and estimated signal matrices. Then,

1. When $\alpha = 0$,
$$\mathbb{E}[\|EE^T - \hat{E}\hat{E}^T\|] \leq \sqrt{d - k + 2\sigma^2 \sum_{r\leq k, s>d} (\lambda_r - \lambda_s)^{-2}}$$
2. When $0 < \alpha \leq 1$,
$$\mathbb{E}[\|EE^T - \hat{E}\hat{E}^T\|] \leq \sqrt{\sum_{i=k+1}^{d} \lambda_i^{4\alpha}} + 2\sqrt{2n}\alpha\sigma\sqrt{\sum_{i=1}^{k}\lambda_i^{4\alpha-2}} + \sqrt{2}\sum_{i=1}^{k}(\lambda_i^{2\alpha} - \lambda_{i+1}^{2\alpha})\sigma\sqrt{\sum_{r\leq i<s}(\lambda_r - \lambda_s)^{-2}}$$

첫 번째 term은 계속 봐왔던 lost signal power의 양이다. 작은 k를 선택하면 bias가 높아질 것이다. 그와는 반면에, k가 크다면두 번째, 세 번째 term이 커진다.

4.1, 4.2, 4.3을 살펴보면 모두 임베딩 간의 PIP loss를 bias, variance로 접근하며 k를 조정함에 따라서 이 둘이 서로 상충됨을 알 수 있다. 즉, ML에서 등장하는 bias variance trade-off 개념이 그대로 적용되는 것이다.

5. Two New Discoveries

section 5에서는 파라미터 α와 단어 임베딩의 관계를 통해 skip gram, glove 모델의 robustness을 밝히고 PIP loss를 차원 선택의 방법으로 사용하는 것에 대해 논의한다. 여기서는 후자에 대해 자세하게 살펴본다.

Theorem 3을 보면 추정해야할 모수는 λ, σ이다. 따라서 이를 적절하게 추정하면 Theorem 3의 approximation을 사용할 수 있다.

Noise Estimation

데이터를 랜덤하게 두 subsets으로 나누고 아래의 근사 행렬을 얻는다.

$$\tilde{M}_1 = M + Z_1,\ \tilde{M}_2 = M + Z_2$$

Z_1, Z_2는 분산이 $2\sigma^2$에서 얻은 independent copies이다. 이제 $\tilde{M_1} - \tilde{M_2} = Z_1 - Z_2$는 평균이 0이고 분산이 $4\sigma^2$인 random matrix이다. σ에 대한 estimator는 아래와 같이 sample standard deviation 이다.

$$\hat{\sigma} = \frac{1}{2\sqrt{nm}}||\tilde{M_1} - \tilde{M_2}||$$

Spectral Estimation

$$\hat{\lambda}_i = (\tilde{\lambda}_i - 2\sigma\sqrt{n})_+$$

$\tilde{\lambda}_i$는 i번째 empirical singular value이고 아마도 여기서 σ을 $\hat{\sigma}$로 추정해야 할 것 같다.

Monte Carlo Simulation

clean signal matrix인 $M = UDV$와 noisy signal matrix인 $\tilde{M} = M + Z$를 simulate한다. $M, \tilde{M}$를 factorize함으로써 $E, \hat{E}$를 얻을 수 있고 이 둘의 PIP loss을 계산한다. 논문에서는 Monte Carlo Simulation을 통해서 PIP loss을 근사한 방법을 사용했다.

이후 내용은 skip gram, word2vec, glove 모델의 적절한 벡터의 차원을 실험하는 내용이다.