A Generalized Principal Component Analysis for Word Embedding

UPPSALA UNIVERSITET

Ali Basirat

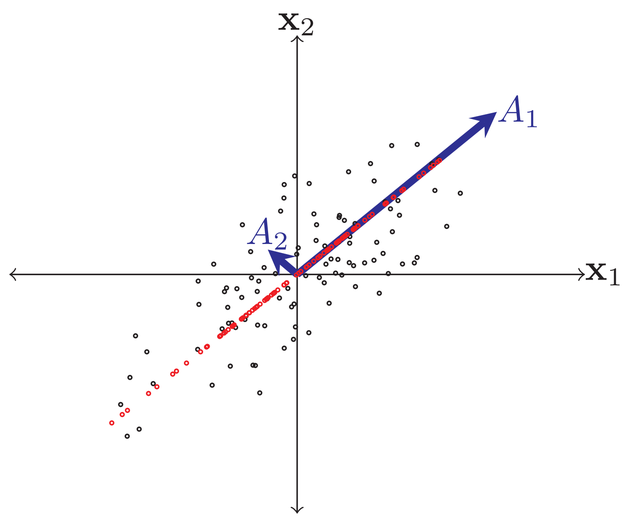
Introduction

- **Word Embeddings**
  - Representing words as elements of a vector space
  - Providing a continuous representation of words
  - Useful for machine learning

- **Principal Component Analysis (PCA)**
  - A method to study the structure of a data matrix
  - Looks for latent variables $\mathbf{Y}$ that retain most of variations in $\mathbf{X}$

$$\mathbf{Y} = A^T(\mathbf{X} - \mathbf{E}[\mathbf{X}])$$

$A = [A_1 \dots A_k]$ has k top eigenvectors of $\text{cov}(\mathbf{X} - \mathbf{E}[\mathbf{X}])$



- **PCA for Word Embedding**
  - $\mathbf{X}$ is a random vector of context units (e.g., word forms)

$$\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_m)^T$$

  - $\mathbf{x}_i$ is the frequency of context unit i with a word

Research Questions

- **What are the limitations of PCA for word embedding?**
  - The distribution of $\mathbf{X}$ can be far from the normal distribution
    - PCA works better with normally distributed data

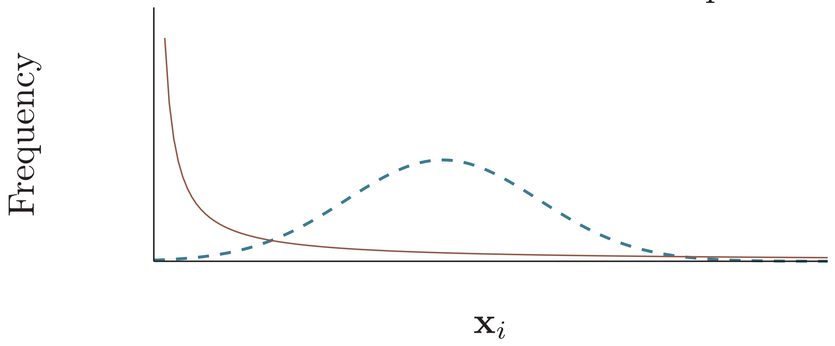
$$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

    - Elements of $\mathbf{X}$ follow a mixture of Binomial distribution

$$p(\mathbf{x}_i) = \sum_{j=1}^{n} p_j f_{ij}$$

$$f_{ij} \sim \text{B}(n(v_j), p(\mathbf{x}_i|v_j))$$

    - The mixture distribution is closer to the Zipfian distribution

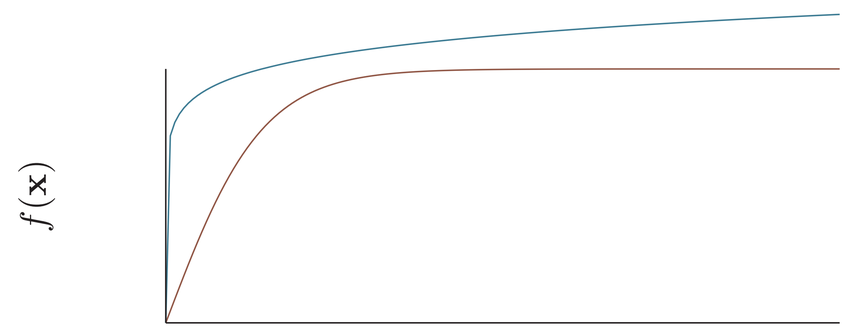


  - A huge amount of memory is needed to decompose $\text{cov}(\mathbf{X} - \mathbf{E}[\mathbf{X}])$
    - $\mathbf{X}$ is sparse
    - $\mathcal{X} = \mathbf{X} - \mathbf{E}[\mathbf{X}]$ is dense
    - The memory complexity of $\mathcal{X}$ can be $O(n^k)$ with $k \geq 2$

- **How to mitigate the limitations of PCA for word embedding?**
  - A transformation to normalize the distribution of $\mathbf{X}$
  - An efficient algorithm to decompose $\mathcal{X} = \mathbf{X} - \mathbf{E}[\mathbf{X}]$

Solutions

- **Transformation**
  - Any monotonically increasing concave function (e.g., $\sqrt[n]{\mathbf{x}}$)



  - f compresses data along the top eigenvectors of $\text{cov}(\mathbf{X})$
  - f expands data along the remaining eigenvectors of $\text{cov}(\mathbf{X})$
  - Optimization the parameters of the function

$$\hat{\theta} = \arg\max_{\theta} H(f(\mathbf{X}; \theta))$$

  - Simulated annealing for estimating $\hat{\theta}$

- **Randomized SVD to decomposes $\mathcal{X} = \mathbf{X} - \mathbf{E}[\mathbf{X}]$.**

procedure CENTRED-SVD(X, E, k, K)
  Draw an $n \times K$ standard Gaussian matrix Ω
  Form the sample matrix $X_1 \leftarrow X\Omega$
  Compute the economy-size QR factorization $X_1 = Q_1 R_1$
  Compute $QR = Q_1 R_1 - E\mathbf{1}_n^T$ using the QR-update algorithm
  Form $Y \leftarrow Q^T X - Q^T E\mathbf{1}_n^T$
  Compute the singular value decomposition of $Y = U_1 \Sigma V^T$
  $U \leftarrow QU_1$
  return (U, Σ, V)
end procedure

Experiments and Results

- **Parameters**
  - $f(\mathbf{X}) = \mathbf{X}^\theta$ with $\theta \in (0, 1)$
  - In CENTRED-SVD: $k = 50$ and $K = 2k$

- **Experiments**
  - Trained on a corpus with 11 billion words
  - 1 million lowercased words with frequency ≤ 50
  - Tested on different tasks
  - Compared with other word embedding methods

- **Results**

	Sim. Corr.	POS	NER	UAS	LAS
`RI`	0.2	95.4	93.6	90.5	88.2
`HPCA`	0.2	96.3	96.2	90.7	88.6
`CBOW`	0.5	96.3	96.2	**92.1**	**90.1**
`SGRAM`	**0.6**	**96.4**	97.2	**92.1**	**90.0**
`GloVe`	0.5	95.2	**97.4**	91.9	89.9
`PWE`	0.5	96.0	96.4	91.9	89.9

A comparison between principal word embeddings (`PWE`) and other sets of word embeddings. `CBOW` and `SGRAM` are two models of `word2vec`. Test data: POS tagging on sections 19–21 of WSJ - NER on the test set of CoNLL-2003 shared task, parsing (UAS and LAS) on Section 23 of WSJ.

	`RI`	`HPCA`	`GloVe`	`PWE`	`word2vec`
Scanning	7200	7200	7200	7200	36000
DimRed	**180**	480	8040	900	
Sum	**7380**	7680	15240	8100	36000

The amount of time (seconds) required by each of the word embedding methods to perform *scanning* and dimensionality reduction *DimRed*. `PWE` refers to the principal word embedding method.