1. Why Glove?

Before Glove was introduced in 2014, there are two main classes of methods to find word embeddings.

- global matrix factorization: latent semantic analysis (LSA)
- local context window methods: skip-gram, CBOW

`Global matrix factorization` methods are count-based and rely on matrix factorization (e.g. LSA, HAL). While these methods effectively leverage global statistical information, they are primarily used to capture
word similarities and do poorly on tasks such as word analogy, indicating a sub-optimal vector space structure.

`Local context window` methods learn word embeddings by making predictions in local context windows. These models demonstrate the capacity to capture complex linguistic patterns such as analogy beyond word similarity, but fail to make use of the global co-occurrence statistics.

`Glove` takes the advantage of both global statistical information and local context information of words.

2. Loss Function

First of all, we create a word-word co-occurrence matrix and count how many times each two words occur together based on some window size. Then the co-occurrence of two words w_i and w_j can be denoted as X_{ij}.

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

The occurrence of word w_i in the full text can be represented as $X_i = \sum_k X_{ik}$ (summing over all words in the full text). For two words w_i and w_j, the probability of w_j occurring provided with w_i occurs is $P_{ij} = P(w_j|w_i) = \frac{P(w_i,w_j)}{P(w_i)} = \frac{X_{ij}}{X_i}$. Based on this concept, we have the below table:

	x = solid	x = gas	x = water	x = fashion
$P(x\|\text{ice})$	1.9 x 10^{-4}	6.6 x 10^{-5}	3.0 x 10^{-3}	1.7 x 10^{-5}
$P(x\|\text{steam})$	2.2 x 10^{-5}	7.8 x 10^{-4}	2.2 x 10^{-3}	1.8 x 10^{-5}
$\frac{P(x\|\text{ice})}{P(x\|\text{steam})}$	8.9	8.5 x 10^{-2}	1.36	0.96

The first two rows of the above table do not give us much information. However, if you look at the third row, you can see that there is something interesting when we start to compare the conditional probabilities of different words.

- When $x = solid$, $\frac{P(solid|ice)}{P(solid|steam)} >> 1$. We know that ice is closer to solid than steam to solid.
- When $x = gas$, $\frac{P(gas|ice)}{P(gas|steam)} << 1$. We know that ice is farther to gas than steam to gas.
- When $x = water$, $\frac{P(water|ice)}{P(water|steam)} \approx 1$. We know that ice and steam are equally close to water.
- When $x = fashion$, $\frac{P(fashion|ice)}{P(fashion|steam)} \approx 1$. We know that ice and steam are equally close to fashion.

The above example shows that we can do something interesting with 3 words involved: $w_i, w_j, \tilde{w}_k$.

$$F(v_i, v_j, \tilde{v}_k) = \frac{P_{ik}}{P_{jk}}$$

- v_i: vector of central target word w_i
- v_j: vector of central target word w_j
- $\tilde{v}_k$: vector of context word $\tilde{w}_k$
- $\frac{P_{ik}}{P_{jk}}$ can be obtained from the text: ratio of probability of words $w_i \& w_k$ over probability of words $w_j \& w_k$

We can also define our loss function here. Basically the goal is the train a function F so that it represents the value of $\frac{P_{ik}}{P_{jk}}$.

$$J = \sum_{i,j,k} (F(v_i, v_j, \tilde{v}_k) - \frac{P_{i,k}}{P_{j,k}})^2$$

The value of $\frac{P_{ik}}{P_{jk}}$ can be obtained from the text. However, we have no idea of the form of function $F(v_i, v_j, \tilde{v}_k)$ that satisfies the equation $F(v_i, v_j, \tilde{v}_k) \approx \frac{P_{ik}}{P_{jk}}$. There are many possibilities of the function. One possible form is

$$F(v_i - v_j, \tilde{v}_k) \approx \frac{P_{ik}}{P_{jk}}$$

The left hand side of the above equation is a vector while the right hand side is a scalar. Let's convert the left hand side of the equation to a new form so that it's also a scalar (applying dot product of two vectors)

$$F((v_i - v_j)^T \tilde{v}_k) \approx \frac{P_{ik}}{P_{jk}}$$

We wish the co-occurrence matrix X_{ik} and X_{jk} are symmetric matrix where $X_{ik} = X_{ki}$ and $X_{jk} = X_{kj}$ (number of times word i appearing in the context of word k is same as number of times word k appearing in the context of word i).

Let's assume function F satisfies homomorphism. This means the output of the function stays unchanged if we exchange the indices between i and k or between j and k. This means function F possibly can be a function of exp and satisfies the form below

$$F((v_i - v_j)^T \tilde{v}_k) = \frac{F(v_i^T \tilde{v}_k)}{F(v_j^T \tilde{v}_k)} \approx \frac{P_{ik}}{P_{jk}}$$

Just let the numerators equal each other. We can get

$$F(v_i^T \tilde{v}_k) \approx P_{ik}$$

We also know

$$P_{ik} = \frac{X_{ik}}{X_i}$$

Then we can get

$$F(v_i^T \tilde{v}_k) \approx P_{ik} = \frac{X_{ik}}{X_i}$$

Plug $F = exp$ into the above equation

$$v_i^T \tilde{v}_k \approx log(P_{ik}) = log(X_{ik}) - log(X_i)$$

Let's take a look at the above equation. The component $log(X_i)$ makes the equation asymmetric. Since $log(X_i)$ is independent from the context word k, we can use two bias terms (constants) to absorb the value of it and make the above equation symmetric

$$v_i^T \tilde{v}_k + b_i + b_k \approx log(X_{ik})$$

$log(X_{ik})$ represents the co-occurrence matrix of words w_i and w_k in the text and can be obtained from the text.

The loss function is

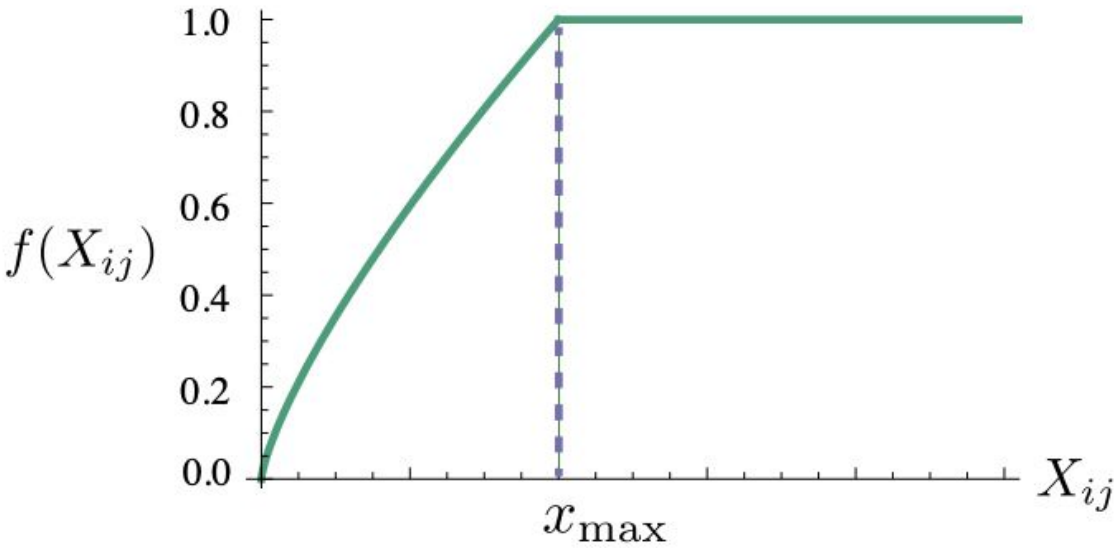
$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} (v_i^T v_j + b_i + b_j - log(X_{ij}))^2$$

A disadvantage of the above loss function is that the co-occurrences of all word pairs are the same. However, some low-occurring word pairs may be noise. Therefore, a weighing factor $f(X_{ij})$ is added to the loss function.

$$J = f(X_{ij}) \sum_{i=1}^{V} \sum_{j=1}^{V} (v_i^T v_j + b_i + b_j - log(X_{ij}))^2$$

where

$$f(x) = \begin{cases} (x/x_{\max})^\alpha, & \text{if } x < x_{\max} \\ 1, & \text{otherwise} \end{cases}$$



- The word pairs with frequent co-occurrences will have larger weight than those with rare co-occurrences. Therefore, $f(x)$ is non-decreasing.
- We don't want the value of $f(x)$ to be overweighed. It should stop when its value reaches a threshold.
- If two words never co-occur, i.e. $X_{ij} = 0$, they shouldn't be part of the computation of the loss function. Therefore, $f(0) = 0$.
- Based on experience, $\alpha = \frac{3}{4}$ and the value of x_{max} is dependent on the corpus.