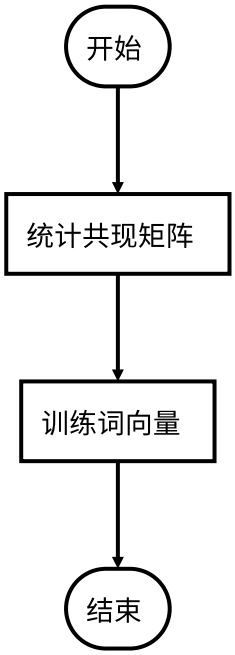
理解GloVe模型

概述

- 模型目标：进行词的向量化表示，使得向量之间尽可能多地蕴含语义和语法的信息。
- 输入：语料库
- 输出：词向量
- 方法概述：首先基于语料库构建词的共现矩阵，然后基于共现矩阵和GloVe模型学习词向量。



统计共现矩阵

设共现矩阵为XX，其元素为$X_{i,j}$Xi,j。
$X_{i,j}$Xi,j的意义为：在整个语料库中，单词*i*和单词*j*共同出现在一个窗口中的次数。
举个栗子：
设有语料库：

> i love you but you love him i am sad

这个小小的语料库只有1个句子，涉及到7个单词：i、love、you、but、him、am、sad。
如果我们采用一个窗口宽度为5（左右长度都为2）的统计窗口，那么就有以下窗口内容：

窗口标号	中心词	窗口内容
0	i	i love you
1	love	i love you but

窗口标号	中心词	窗口内容
2	you	i love you but you
3	but	love you but you love
4	you	you but you love him
5	love	but you love him i
6	him	you love him i am
7	i	love him i am sad
8	am	him i am sad
9	sad	i am sad

窗口0、1长度小于5是因为中心词左侧内容少于2个，同理窗口8、9长度也小于5。
以窗口5为例说明如何构造共现矩阵：
中心词为love，语境词为but、you、him、i；则执行：

$$X_{love,but} += 1$$
Xlove,but+=1

$$X_{love,you} += 1$$
Xlove,you+=1

$$X_{love,him} += 1$$
Xlove,him+=1

$$X_{love,i} += 1$$
Xlove,i+=1

使用窗口将整个语料库遍历一遍，即可得到共现矩阵XX。

使用GloVe模型训练词向量

模型公式

先看模型，代价函数长这个样子：

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

J=∑i,jNf(Xi,j)(viTvj+bi+bj−log(Xi,j))2

$v_i$vi，$v_j$vj是单词ii和单词jj的词向量，$b_i$bi，$b_j$bj是两个标量（作者定义的偏差项），ff是权重函数（具体函数公式及功能下一节介绍），NN是词汇表的大小（共现矩阵维度为$N * N$N＊N）。
可以看到，GloVe模型没有使用神经网络的方法。

模型怎么来的

那么作者为什么这么构造模型呢？首先定义几个符号：

$$X_i = \sum_{j=1}^{N} X_{i,j}$$
Xi=∑j=1NXi,j

其实就是矩阵单词ii那一行的和；

$$P_{i,k} = \frac{X_{i,k}}{X_i}$$
Pi,k=Xi,kXi

条件概率，表示单词kk出现在单词ii语境中的概率；

$$ratio_{i,j,k} = \frac{P_{i,k}}{P_{j,k}}$$
ratioi,j,k=Pi,kPj,k

两个条件概率的比率。
作者的灵感是这样的：
作者发现，$ratio_{i,j,k}$ratioi,j,k这个指标是有规律的，规律统计在下表：

$ratio_{i,j,k}$ratioi,j,k**的值**	单词j, k单词j,k**相关**	单词j, k单词j,k**不相关**
单词i, k单词i,k相关	趋近1	很大
单词i, k单词i,k不相关	很小	趋近1

很简单的规律，但是有用。
思想：假设我们已经得到了词向量，如果我们用词向量$v_i$vi、$v_j$vj、$v_k$vk通过某种函数计算$ratio_{i,j,k}$ratioi,j,k，能够同样得到这样的规律的话，就意味着我们词向量与共现矩阵具有很好的一致性，也就说明我们的词向量中蕴含了共现矩阵中所蕴含的信息。
设用词向量$v_i$vi、$v_j$vj、$v_k$vk计算$ratio_{i,j,k}$ratioi,j,k的函数为$g(v_i, v_j, v_k)$g(vi,vj,vk)（我们先不去管具体的函数形式），那么应该有：

$$\frac{P_{i,k}}{P_{j,k}} = ratio_{i,j,k} = g(v_i, v_j, v_k)$$
Pi,kPj,k=ratioi,j,k=g(vi,vj,vk)

即：

$$\frac{P_{i,k}}{P_{j,k}} = g(v_i, v_j, v_k)$$

Pi,kPj,k=g(vi,vj,vk)

即二者应该尽可能地接近；
很容易想到用二者的差方来作为代价函数：

$$J = \sum_{i,j,k}^{N} (\frac{P_{i,k}}{P_{j,k}} - g(v_i, v_j, v_k))^2$$

J=∑i,j,kN(Pi,kPj,k−g(vi,vj,vk))2

但是仔细一看，模型中包含3个单词，这就意味着要在$N * N * N$N＊N＊N的复杂度上进行计算，太复杂了，最好能再简单点。
现在我们来仔细思考$g(v_i, v_j, v_k)$g(vi,vj,vk)，或许它能帮上忙；
作者的脑洞是这样的：
1. 要考虑单词ii和单词jj之间的关系，那$g(v_i, v_j, v_k)$g(vi,vj,vk)中大概要有这么一项吧：$v_i - v_j$vi−vj；嗯，合理，在线性空间中考察两个向量的相似性，不失线性地考察，那么$v_i - v_j$vi−vj大概是个合理的选择；
2. $ratio_{i,j,k}$ratioi,j,k是个标量，那么$g(v_i, v_j, v_k)$g(vi,vj,vk)最后应该是个标量啊，虽然其输入都是向量，那内积应该是合理的选择，于是应该有这么一项吧：$(v_i - v_j)^T v_k$(vi−vj)Tvk。
3. 然后作者又往$(v_i - v_j)^T v_k$(vi−vj)Tvk的外面套了一层指数运算$exp()$exp()，得到最终的
$g(v_i, v_j, v_k) = exp((v_i - v_j)^T v_k)$g(vi,vj,vk)=exp((vi−vj)Tvk)；
最关键的第3步，为什么套了一层$exp()$exp()？
套上之后，我们的目标是让以下公式尽可能地成立：

$$\frac{P_{i,k}}{P_{j,k}} = g(v_i, v_j, v_k)$$

Pi,kPj,k=g(vi,vj,vk)

即：

$$\frac{P_{i,k}}{P_{j,k}} = exp((v_i - v_j)^T v_k)$$

Pi,kPj,k=exp((vi−vj)Tvk)

即：

$$\frac{P_{i,k}}{P_{j,k}} = exp(v_i^T v_k - v_j^T v_k)$$

Pi,kPj,k=exp(viTvk−vjTvk)

即：

$$\frac{P_{i,k}}{P_{j,k}} = \frac{exp(v_i^T v_k)}{exp(v_j^T v_k)}$$

Pi,kPj,k=exp(viTvk)exp(vjTvk)

然后就发现找到简化方法了：只需要让上式分子对应相等，分母对应相等，即：

$$P_{i,k} = exp(v_i^T v_k)并且P_{j,k} = exp(v_j^T v_k)$$
Pi,k=exp(viTvk)并且Pj,k=exp(vjTvk)

然而分子分母形式相同，就可以把两者统一考虑了，即：

$$P_{i,j} = exp(v_i^T v_j)$$
Pi,j=exp(viTvj)

本来我们追求：

$$\frac{P_{i,k}}{P_{j,k}} = g(v_i, v_j, v_k)$$
Pi,kPj,k=g(vi,vj,vk)

现在只需要追求：

$$P_{i,j} = exp(v_i^T v_j)$$
Pi,j=exp(viTvj)

两边取个对数：

$$log(P_{i,j}) = v_i^T v_j$$
log(Pi,j)=viTvj

那么代价函数就可以简化为：

$$J = \sum_{i,j}^{N} (log(P_{i,j}) - v_i^T v_j)^2$$
J=∑i,jN(log(Pi,j)−viTvj)2

现在只需要在$N * N$N＊N的复杂度上进行计算，而不是$N * N * N$N＊N＊N，现在关于为什么第3步中，外面套一层$exp()$exp()就清楚了，正是因为套了一层$exp()$exp()，才使得差形式变成商形式，进而等式两边分子分母对应相等，进而简化模型。
然而，出了点问题。
仔细看这两个式子：

$$log(P_{i,j}) = v_i^T v_j和log(P_{j,i}) = v_j^T v_i$$
log(Pi,j)=viTvj和log(Pj,i)=vjTvi

$log(P_{i,j})$log(Pi,j)不等于$log(P_{j,i})$log(Pj,i)但是$v_i^T v_j$viTvj等于$v_j^T v_i$vjTvi；即等式左侧不具有对称性，但是右侧具有对称性。
数学上出了问题。
补救一下好了。
现将代价函数中的条件概率展开：

$$log(P_{i,j}) = v_i^T v_j$$
log(Pi,j)=viTvj

即为:

$$log(X_{i,j}) - log(X_i) = v_i^T v_j$$

log(Xi,j)−log(Xi)=viTvj

将其变为：

$$log(X_{i,j}) = v_i^T v_j + b_i + b_j$$

log(Xi,j)=viTvj+bi+bj

即添了一个偏差项$b_j$bj，并将$log(X_i)$log(Xi)吸收到偏差项$b_i$bi中。
于是代价函数就变成了：

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

J=∑i,jN(viTvj+bi+bj−log(Xi,j))2

然后基于出现频率越高的词对儿权重应该越大的原则，在代价函数中添加权重项，于是代价函数进一步完善：

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

J=∑i,jNf(Xi,j)(viTvj+bi+bj−log(Xi,j))2

具体权重函数应该是怎么样的呢？
首先应该是非减的，其次当词频过高时，权重不应过分增大，作者通过实验确定权重函数为：

$$f(x) = \begin{cases} (x/xmax)^{0.75}, & \text{if } x < xmax \\ 1, & \text{if } x >= xmax \end{cases}$$

f(x)={(x/xmax)0.75,if x<xmax1,if x>=xmax

到此，整个模型就介绍完了。