10.5 全局向量的词嵌入（GloVe）

让我们先回顾一下word2vec中的跳字模型。将跳字模型中使用softmax运算表达的条件概率$P(w_j \mid w_i)$记作q_{ij}，即

$$q_{ij} = \frac{\exp(\boldsymbol{u}_j^\top \boldsymbol{v}_i)}{\sum_{k \in \mathcal{V}} \exp(\boldsymbol{u}_k^\top \boldsymbol{v}_i)},$$

其中$\boldsymbol{v}_i$和$\boldsymbol{u}_i$分别是索引为i的词w_i作为中心词和背景词时的向量表示，$\mathcal{V} = \{0, 1, \ldots, |\mathcal{V}| - 1\}$为词典索引集。

对于词w_i，它在数据集中可能多次出现。我们将每一次以它作为中心词的所有背景词全部汇总并保留重复元素，记作多重集（multiset）$\mathcal{C}_i$。一个元素在多重集中的个数称为该元素的重数（multiplicity）。举例来说，假设词w_i在数据集中出现2次：文本序列中以这2个w_i作为中心词的背景窗口分别包含背景词索引$2, 1, 5, 2$和$2, 3, 2, 1$。那么多重集$\mathcal{C}_i = \{1, 1, 2, 2, 2, 2, 3, 5\}$，其中元素1的重数为2，元素2的重数为4，元素3和5的重数均为1。将多重集$\mathcal{C}_i$中元素j的重数记作x_{ij}：它表示了整个数据集中所有以w_i为中心词的背景窗口中词w_j的个数。那么，跳字模型的损失函数还可以用另一种方式表达：

$$-\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} x_{ij} \log\ q_{ij}.$$

我们将数据集中所有以词w_i为中心词的背景词的数量之和$|\mathcal{C}_i|$记为x_i，并将以w_i为中心词生成背景词w_j的条件概率x_{ij}/x_i记作p_{ij}。我们可以进一步改写跳字模型的损失函数为

$$-\sum_{i \in \mathcal{V}} x_i \sum_{j \in \mathcal{V}} p_{ij} \log\ q_{ij}.$$

上式中，$-\sum_{j \in \mathcal{V}} p_{ij} \log\ q_{ij}$计算的是以$w_i$为中心词的背景词条件概率分布$p_{ij}$和模型预测的条件概率分布$q_{ij}$的交叉熵，且损失函数使用所有以词$w_i$为中心词的背景词的数量之和来加权。最小化上式中的损失函数会令预测的条件概率分布尽可能接近真实的条件概率分布。

然而，作为常用损失函数的一种，交叉熵损失函数有时并不是好的选择。一方面，正如我们在10.2节（近似训练）中所提到的，令模型预测q_{ij}成为合法概率分布的代价是它在分母中基于整个词典的累加项。这很容易带来过大的计算开销。另一方面，词典中往往有大量生僻词，它们在数据集中出现的次数极少。而有关大量生僻词的条件概率分布在交叉熵损失函数中的最终预测往往并不准确。

10.5.1 GloVe模型

鉴于此，作为在word2vec之后提出的词嵌入模型，GloVe模型采用了平方损失，并基于该损失对跳字模型做了3点改动 [1]：

1. 使用非概率分布的变量$p'_{ij} = x_{ij}$和$q'_{ij} = \exp(\boldsymbol{u}_j^\top \boldsymbol{v}_i)$，并对它们取对数。因此，平方损失项是$(\log\, p'_{ij} - \log\, q'_{ij})^2 = (\boldsymbol{u}_j^\top \boldsymbol{v}_i - \log\, x_{ij})^2$。
2. 为每个词w_i增加两个为标量的模型参数：中心词偏差项b_i和背景词偏差项c_i。
3. 将每个损失项的权重替换成函数$h(x_{ij})$。权重函数$h(x)$是值域在$[0, 1]$的单调递增函数。

如此一来，GloVe模型的目标是最小化损失函数

$$\sum_{i\in\text{V}} \sum_{j\in\text{V}} h(x_{ij}) (\boldsymbol{u}_j^\top \boldsymbol{v}_i + b_i + c_j - \log\, x_{ij})^2 .$$

其中权重函数$h(x)$的一个建议选择是：当$x < c$时（如$c = 100$），令$h(x) = (x/c)^\alpha$（如$\alpha = 0.75$），反之令$h(x) = 1$。因为$h(0) = 0$，所以对于$x_{ij} = 0$的平方损失项可以直接忽略。当使用小批量随机梯度下降来训练时，每个时间步我们随机采样小批量非零x_{ij}，然后计算梯度来迭代模型参数。这些非零x_{ij}是预先基于整个数据集计算得到的，包含了数据集的全局统计信息。因此，GloVe模型的命名取“全局向量”（Global Vectors）之意。

需要强调的是，如果词w_i出现在词w_j的背景窗口里，那么词w_j也会出现在词w_i的背景窗口里。也就是说，$x_{ij} = x_{ji}$。不同于word2vec中拟合的是非对称的条件概率p_{ij}，GloVe模型拟合的是对称的$\log\, x_{ij}$。因此，任意词的中心词向量和背景词向量在GloVe模型中是等价的。但由于初始化值的不同，同一个词最终学习到的两组词向量可能不同。当学习得到所有词向量以后，GloVe模型使用中心词向量与背景词向量之和作为该词的最终词向量。

10.5.2 从条件概率比值理解GloVe模型

我们还可以从另外一个角度来理解GloVe模型。沿用本节前面的符号，$P(w_j \mid w_i)$表示数据集中以w_i为中心词生成背景词w_j的条件概率，并记作p_{ij}。作为源于某大型语料库的真实例子，以下列举了两组分别以“ice”（冰）和“steam”（蒸汽）为中心词的条件概率以及它们之间的比值 [1]：

w_k=	“solid”	“gas”	“w
$p_1 = P(w_k \mid$ “ice” $)$	0.00019	0.000066	0
$p_2 = P(w_k \mid$ “steam” $)$	0.000022	0.00078	0.
p_1/p_2	8.9	0.085	1

我们可以观察到以下现象。

- 对于与“ice”相关而与“steam”不相关的词w_k，如w_k =“solid”（固体），我们期望条件概率比值较大，如上表最后一行中的值8.9；

- 对于与“ice”不相关而与“steam”相关的词w_k，如w_k=“gas”（气体），我们期望条件概率比值较小，如上表最后一行中的值0.085；
- 对于与“ice”和“steam”都相关的词w_k，如w_k=“water”（水），我们期望条件概率比值接近1，如上表最后一行中的值1.36；
- 对于与“ice”和“steam”都不相关的词w_k，如w_k=“fashion”（时尚），我们期望条件概率比值接近1，如上表最后一行中的值0.96。

由此可见，条件概率比值能比较直观地表达词与词之间的关系。我们可以构造一个词向量函数使它能有效拟合条件概率比值。我们知道，任意一个这样的比值需要3个词w_i、w_j和w_k。以w_i作为中心词的条件概率比值为p_{ij}/p_{ik}。我们可以找一个函数，它使用词向量来拟合这个条件概率比值

$$f(\boldsymbol{u}_j, \boldsymbol{u}_k, \boldsymbol{v}_i) \approx \frac{p_{ij}}{p_{ik}}.$$

这里函数f可能的设计并不唯一，我们只需考虑一种较为合理的可能性。注意到条件概率比值是一个标量，我们可以将f限制为一个标量函数：$f(\boldsymbol{u}_j, \boldsymbol{u}_k, \boldsymbol{v}_i) = f((\boldsymbol{u}_j - \boldsymbol{u}_k)^\top \boldsymbol{v}_i)$。交换索引$j$和$k$后可以看到函数$f$应该满足$f(x)f(-x) = 1$，因此一种可能是$f(x) = \exp(x)$，于是

$$f(\boldsymbol{u}_j, \boldsymbol{u}_k, \boldsymbol{v}_i) = \frac{\exp(\boldsymbol{u}_j^\top \boldsymbol{v}_i)}{\exp(\boldsymbol{u}_k^\top \boldsymbol{v}_i)} \approx \frac{p_{ij}}{p_{ik}}.$$

满足最右边约等号的一种可能是$\exp(\boldsymbol{u}_j^\top \boldsymbol{v}_i) \approx \alpha p_{ij}$，这里$\alpha$是一个常数。考虑到$p_{ij} = x_{ij}/x_i$，取对数后$\boldsymbol{u}_j^\top \boldsymbol{v}_i \approx \log\ \alpha + \log\ x_{ij} - \log\ x_i$。我们使用额外的偏差项来拟合$-\log\ \alpha + \log\ x_i$，例如，中心词偏差项$b_i$和背景词偏差项$c_j$：

$$\boldsymbol{u}_j^\top \boldsymbol{v}_i + b_i + c_j \approx \log(x_{ij}).$$

对上式左右两边取平方误差并加权，我们可以得到GloVe模型的损失函数。

小结

- 在有些情况下，交叉熵损失函数有劣势。GloVe模型采用了平方损失，并通过词向量拟合预先基于整个数据集计算得到的全局统计信息。
- 任意词的中心词向量和背景词向量在GloVe模型中是等价的。