6 Skip-Gram与CBOW的数学推导

6.1 基本的假设

他们的任务都是学习两个权重矩阵，词向量矩阵C，和输出权重矩阵θ,通过两个矩阵可以表示词w上下文中有词u的概率，由于对单个词来说，这是一个二项分布（w的上下文有或者没有u），所以用sigmoid归一化：

$$P(u \in context(w)|w;C,\theta) = sigmoid(\theta_u \cdot C_w^T)$$

其中C_w是词w对应的词向量，也就是矩阵C中的一行，θ_u是输出矩阵的一行，它也同样与一个词u对应，因为某个词向量与输出矩阵的某一行做点积实际上得到了输出向量中对词u的概率。

6.2 CBOW

CBOW通过确定的w的上下文$context(w)$来预测$w=u$的概率，对每一个属于w上下文的词k，按照上面的假设给出u在它的上下文范围内的概率，且对每个属于$context(w)$的词k的预测都能增强或削弱这种概率，不妨假设对所有k的预测结果地位相等(实际上假设更靠近w的k预测结果权重更大能得到更好的效果)，即对所有预测求均值作为最终的归一化预测：

$$\begin{aligned}
&P(w=u|context(w);C,\theta)\\
&=\frac{1}{|context(w)|}\cdot\sum_{k\in context(w)}P(u\in context(k)|k;C,\theta)\\
&=\frac{1}{|context(w)|}\cdot sigmoid(\sum_{k\in context(w)}\theta_u\cdot C_k^T)\\
&=sigmoid(\theta_u\cdot\frac{1}{|context(w)|}\cdot\sum_{k\in context(w)}C_k^T)
\end{aligned}$$

可以看见，这相当于把上下文所有词向量相加作为输入。记$\frac{1}{|context(w)|}\cdot\sum_{k\in context(w)}C_k^T=X_w^T$,那么：

$$p(u|context(w);C,\theta)=sigmoid(\theta_u\cdot X_w^T)$$

若采用NCE损失,在词库上以任意分布(Noise Contrast Estimation要求分布律覆盖所有训练样本，但在实际中这个限定条件很弱)做负采样采样，记所有采样以及正确标记的集合为N_w，对N_w中的任意词u,用l_u^w做类别标记，1为正类，0为负类,只有$u=w$时$l_u^w=1$,希望正类的似然最大而负类似然最小，那么对单个样本所有采样的似然：

$$\begin{aligned}
&g(w;C,\theta)\\
&=\prod_{u\in N_w}\{l_u^w\cdot p(u|context(w);C,\theta)+(1-l_u^w)(1-p(u|context(w);C,\theta))\}\\
&=\prod_{u\in N_w}\{l_u^w\cdot sigmoid(\theta_u\cdot X_w^T)+(1-l_u^w)(1-sigmoid(\theta_u\cdot X_w^T))\}
\end{aligned}$$

关于NCE损失的详细推导可以参考(https://zhuanlan.zhihu.com/p/27327191),对于整个训练集上的样本NCE损失应该写成：

$$\begin{aligned}
&G(C,\theta)=\prod_{w\in V}g(w;C,\theta)\\
&=\prod_{w\in V}\prod_{u\in N_w}\{l_u^w\cdot sigmoid(\theta_u\cdot X_w^T)+(1-l_u^w)(1-sigmoid(\theta_u\cdot X_w^T))\}
\end{aligned}$$

取对数：

$$\begin{aligned}
&L = \ln G(C,\theta) \\
&= \sum_{w\in V} \ln g(w;C,\theta) \\
&= \sum_{w\in V}\sum_{u\in N} \ln\{l_u^w \cdot sigmoid(\theta_u \cdot X_w^T) + (1-l_u^w)(1-sigmoid(\theta_u \cdot X_w^T))\} \\
&= \sum_{w\in V}\sum_{u\in N} \{l_u^w \ln sigmoid(\theta_u \cdot X_w^T) + (1-l_u^w)\ln(1-sigmoid(\theta_u \cdot X_w^T))\} \\
&= \sum_{w\in V}\sum_{u\in N} \{l_u^w \cdot \theta_u \cdot X_w^T - \theta_u \cdot X_w^T - \ln(1+e^{-\theta_u \cdot X_w^T})\}
\end{aligned}$$

L就是优化的目标，当然，若采用随机梯度下降，V是从训练集中随机选取的一批样本，对θ求梯度：

$$\begin{aligned}
&\frac{\partial L}{\partial \theta} = \sum_{w\in V}\sum_{u\in N_w} \frac{\partial\{l_u^w \cdot \theta_u \cdot X_w^T - \theta_u \cdot X_w^T - \ln(1+e^{-\theta_u \cdot X_w^T})\}}{\partial \theta_u} \\
&= \sum_{w\in V}\sum_{u\in N_w} \{l_u^w \cdot X_w - X_w + (1-sigmoid(\theta_u \cdot X_w^T)\cdot X_w)\} \\
&= \sum_{w\in V}\sum_{u\in N_w} \{l_u^w - sigmoid(\theta_u \cdot X_w^T)\} X_w
\end{aligned}$$

由于θ_u与X_w在L中地位是等价的，交换两者可以得出对偶式：

$$\frac{\partial L}{\partial X} = \sum_{w\in V}\sum_{u\in N_w} \{l_u^w - sigmoid(\theta_u \cdot X_w^T)\}\theta_u$$

在学习率α下，采用随机梯度下降更新权重以及词向量的算法如下：

```
C,θ初始化
while(!终止条件) do
{
    随机从V中选取样本集batch;
    for(w ∈ batch) do
    {
        抽样得到N_w
        for(u ∈ N_w) do
        {
            g = l_u^w − sigmoid(θ_u · X_w^T)
            ΔC_w += g · θ_u
            Δθ_u = g · X_w
            θ_u^* = θ_u + α · Δθ_u
        }
        C_w^* = C_w + α · ΔC_w
    }
}
```

注意对每个$u \in N_w$,θ_u会累加地影响对X_w的梯度，因此在更新θ_u前要先记录这种影响，这等效于先计算出所有向量的更新量，再对所有向量更新。

6.3 Skip-Gram

Skip-Gram确定了当前词w而预测上下文的似然,对每一个属于上下文的词u，希望由上面概率计算出的似然最大

$$\prod_{u\in context(w)} P(u \in context(w)|w;C,\theta) = \prod_{u\in context(w)} sigmoid(\theta_u \cdot C_w^T)$$

同样考虑NCE损失，对每个w取若干负采样，与其真实上下文构成集合N_w,用l_u^w做标记，与CBWO不同的是，由于w的上下文可能不止一个词，因此有多个可能的u使$l_u^w = 1$,其余负采样所得的u均使$l_u^w = 0$,于是有：

$$g(w;C,\theta) = \prod_{u \in N_w} \{l_u^w \cdot sigmoid(\theta_u \cdot C_w^T) + (1 - l_u^w) \cdot (1 - sigmoid(\theta_u \cdot C_w^T))\}$$

这形如CBOW的似然，后面的推导与更新算法几乎与CBOW的推导一样，唯一的区别在于C_w和X_w，即Skip-Gram输入单个词向量，而CBOW输入上下文所有词向量的均值。