语言模型

语言模型（LM, Language Model）

一个句子s可以由任何字符构成，其概率$p(s)$大小不同，且是未知的。

- s_1=我准备去散步 s_2=我去散步准备 那么：$p(s_1) > p(s_2)$

对于句子空间S，D表示任意可能句子的概率分布。那么估计句子空间S的概率分布D的过程即**语言建模**，D被称作S的**语言模型**LM。

$$\sum_{s \in S} p(s) = 1$$

基本概念

语句 $s = w_1\ w_2\ \ldots\ w_m$ 的先验概率：

$$p(s) = p(w_1) \times p(w_2|w_1) \times p(w_3|w_1 w_2) \times \ldots \times p(w_m|w_1 \ldots w_{m\text{-}1})$$

$$= \prod_{i=1}^{m} p(w_i \mid w_1 \cdots w_{i-1})$$

语言模型

当 i=1 时，$p(w_1|w_0) = p(w_1)$。

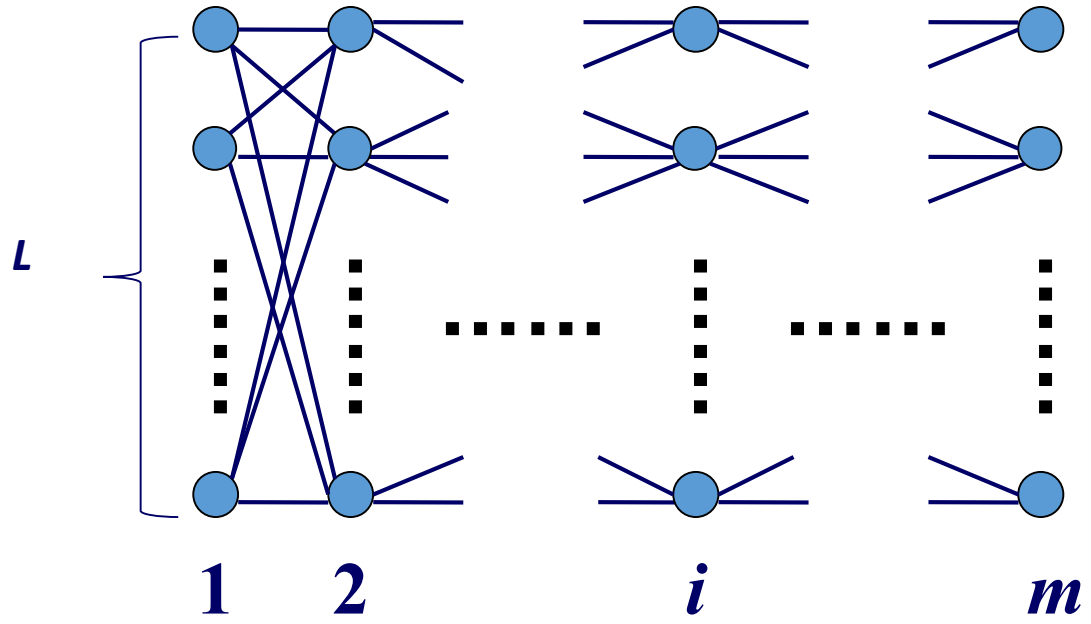
说明：

(1) w_i 可以是字、词、短语或词类等等，称为统计基元。通常以“词”代之。

(2) w_i 的概率由 $w_1, \ldots, w_{i\text{-}1}$ 决定，由特定的一组$w_1, \ldots, w_{i\text{-}1}$ 构成的一个序列，称为 w_i 的历史（history）。

基本概念

问题：随着历史基元数量的增加，不同的“历史”(路径)按指数级增长。对于第 i ($i>1$) 个统计基元，历史基元的个数为 i-1，如果共有 L 个不同的基元，如词汇表，理论上每一个单词都有可能出现在1到 i-1的每一个位置上，那么，i 基元就有 L^{i-1} 种不同的历史情况。必须考虑在所有的 L^{i-1} 种不同历史情况下产生第 i 个基元的概率。那么，模型中有L^m 个自由参数 $p(w_m|w_1 \ldots w_{m-1})$。



如果 L=5000, m = 3, 自由参数的数目为 1250 亿!

基本概念

◆问题解决方法

设法减少历史基元的个数，将 $w_1\ w_2 \ldots w_{i-1}$ 映射到等价类 $S(w_1\ w_2\ \ldots\ w_{i-1})$，使等价类的数目远远小于原来不同历史基元的数目。则有：

$$p(w_i \mid w_1,\cdots,w_{i-1}) = p(w_i \mid S(w_1,\cdots,w_{i-1}))$$

… (5-2)

基本概念

◆如何划分等价类

将两个历史映射到同一个等价类，当且仅当这两个历史中的最近 ***n*-1** 个基元相同，即：

$$H_1: w_1\, w_2 \ldots \ldots \underbrace{w_{i-n+1}\, w_{i-n+2} \ldots w_{i-1}}_{n-1}\, w_i \ldots\ldots$$

$$H_2: \quad v_1\, v_2 \ldots \ldots \underbrace{v_{k-n+1}\, v_{k-n+2} \ldots v_{k-1}}_{n-1}\, v_k \ldots\ldots$$

$$S(w_1, w_2, \cdots, w_i) = S(v_1, v_2 \cdots, v_k)$$
$$iff \quad H_1:(w_{i-n+1}, \cdots, w_i) = H_2:(v_{k-n+1}, \cdots, v_k) \qquad \ldots (5\text{-}3)$$

基本概念

这种情况下的语言模型称为 *n* 元文法(*n*-gram)模型。

通常地，

- 当 n=1 时，即出现在第 i 位上的基元 w_i 独立于历史。一元文法也被写为 uni-gram 或 monogram；
- 当 n=2 时, 2-gram (bi-gram) 被称为1阶马尔可夫链；
- 当 n=3 时, 3-gram(tri-gram)被称为2阶马尔可夫链，依次类推。

基本概念

为了保证条件概率在 $i=1$ 时有意义，同时为了保证句子内所有字符串的概率和为 1，即 $\sum_s p(s)=1$，可以在句子首尾两端增加两个标志: <BOS> $w_1\ w_2\ \ldots\ w_m$ <EOS>。不失一般性，对于$n>2$ 的 n-gram，$p(s)$ 可以分解为：

$$p(s)=\prod_{i=1}^{m+1} p(w_i \mid w_{i-n+1}^{i-1})$$

其中，w_i^j 表示词序列 $w_i \ldots w_j$，$w_{i\text{-}n+1}$ 从 w_0 开始，w_0 为 <BOS>，w_{m+1} 为 <EOS>。

基本概念

◆举例：

给定句子：John read a book

增加标记：<BOS> John read a book <EOS>

Unigram：<BOS>, John, read, a, book, <EOS>

Bigram：(<BOS>John), (John read), (read a), (a book), (book <EOS>)

Trigram：(<BOS>John read), (John read a), (read a book), (a book <EOS>)

基本概念

<BOS> John read a book <EOS>

基于2元文法的概率为：

$$p(\text{John read a book}) = p(\text{John}|\text{<BOS>}) \times p(\text{read}|\text{John}) \times p(\text{a}|\text{read}) \times$$

$$p(\text{book}|\text{a}) \times p(\text{<EOS>}|\text{book})$$

基本概念

◆应用-1：音字转换问题

给定拼音串：ta shi yan jiu sheng wu de

可能的汉字串：**踏实研究生物的**

他实验救生物的

他使烟酒生物的

他是研究生物的

…… ……

基本概念

$$\widehat{CString} = \arg\max_{CString} p(CString \mid Pinyin)$$

$$= \arg\max_{CString} \frac{p(Pinyin \mid CString) \times p(CString)}{p(Pingyin)}$$

$$= \arg\max_{CString} p(Pinyin \mid CString) \times p(CString)$$

$$= \arg\max_{CString} p(CString)$$

基本概念

CString = {踏实研究生物的，他实验救生物的，他是研究生物的，他使烟酒生雾的，… …}

如果使用 2-gram：

$p(\boldsymbol{CString}_1)$ =p(踏实|<BOS>)×p(研究|踏实)×p(生物|研究)×p(的|生物)×p(<EOS>|的)

$p(\boldsymbol{CString}_2)$ =p(他|<BOS>)×p(实验|他)×p(救|实验)×p(生物|救)×p(的|生物)×p(<EOS>|的)

……

基本概念

如果汉字的总数为：N

- 一元语法：1）样本空间为 N
  2）只选择使用频率最高的汉字
- 2元语法：1）样本空间为 N^2
  2）效果比一元语法明显提高
- 估计对汉字而言四元语法效果会好一些
- 智能狂拼、微软拼音输入法基于 ***n*-gram.**

基本概念

◆应用-2：汉语分词问题

给定汉字串：**他是研究生物的。**

可能的汉字串：

1) 他|是|研究生|物|的

2) 他|是|研究|生物|的

基本概念

$$\widehat{Seg} = \arg\max_{Seg} p(Seg \mid Text)$$

$$= \arg\max_{Seg} \frac{p(Text \mid Seg) \times p(Seg)}{p(Text)}$$

$$= \arg\max_{Seg} p(Text \mid Seg) \times p(Seg)$$

$$= \arg\max_{Seg} p(Seg)$$

训练集是标注好的，所以必须后验转先验，即利用Seg来计算生成Text的概率

举例

Text = 门把手弄坏了

Seg1 = 门/ 把/ 手/ 弄/ 坏/ 了

Seg2 = 门把手/ 弄/ 坏/ 了

P(Seg1|Text) =? P(Seg2|Text) =?

P(Seg1|Text) < P(Seg2|Text)

P(Text|Seg1) =? P(Text|Seg2) =?

P(Text|Seg1) = P(Text|Seg2) =1即不管哪个Seg生成的都是Text

$$\begin{aligned}\widehat{Seg} &= \arg\max_{Seg} p(Seg \mid Text) \\ &= \arg\max_{Seg} p(Text \mid Seg) \times p(Seg) \\ &= \arg\max_{Seg} p(Seg)\end{aligned}$$

基本概念

如果采用2元文法：

p(Seg1) =p(他|<BOS>)×p(是|他)×p(研究生|是)× p(物|研究生)×p(的|物)×p(的|<EOS>)

p(Seg2) = p(他|<BOS>)×p(是|他)×p(研究|是)× p(生物|研究)×p(的|生物)× p(的|<EOS>)

问题：如何获得 n 元语法模型？

参数估计

参数估计

◆两个重要概念：

- 训练语料(*training data*)：用于建立模型，确定模型参数的已知语料。
- 最大似然估计(*maximum likelihood Evaluation*, MLE)：用相对频率计算概率的方法。

参数估计

对于 ***n*-gram**，参数 $p(w_i \mid w_{i-n+1}^{i-1})$ 可由最大似然估计求得：

$$p(w_i \mid w_{i-n+1}^{i-1}) = f(w_i \mid w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i})}{\sum_{w_i} c(w_{i-n+1}^{i})}$$

其中，$\sum_{w_i} c(w_{i-n+1}^{i})$ 是历史串 w_{i-n+1}^{i-1} 在给定语料中出现的次数，即 $c(w_{i-n+1}^{i-1})$ 不管 w_i 是什么。

$f(w_i \mid w_{i-n+1}^{i-1})$ 是在给定 w_{i-n+1}^{i-1} 的条件下w_i出现的相对频度，分子为 w_{i-n+1}^{i-1}与 w_i 同现的次数。

参数估计

例如，给定训练语料：

“*John read Moby Dick*”，

“*Mary read a different book*”，

“*She read a book by Cher*”

根据 2 元文法求句子的概率？

参数估计

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

$$p(John \mid <BOS>) = \frac{c(<BOS> John)}{\sum_w c(<BOS> w)} = \frac{1}{3}$$

$$p(read \mid John) = \frac{c(John \quad read)}{\sum_w c(John \quad w)} = \frac{1}{1}$$

$$p(a \mid read) = \frac{c(read \quad a)}{\sum_w c(read \quad w)} = \frac{2}{3}$$

$$p(book \mid a) = \frac{c(a \quad book)}{\sum_w c(a \quad w)} = \frac{1}{2}$$

$$p(<EOS> \mid book) = \frac{c(book \quad <EOS>)}{\sum_w c(book \quad w)} = \frac{1}{2}$$

$$p(John\ read\ a\ book) = \frac{1}{3} \times 1 \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2} \approx 0.06$$

参数估计

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

$p(\textbf{\textit{Cher read a book}}) = ?$

$$= p(Cher|<BOS>) \times p(read|Cher) \times p(a|read) \times p(book|a) \times p(<EOS>|book)$$

$$p(Cher|<BOS>) = \frac{c(<BOS>\ Cher)}{\sum_w c(<BOS>\ w)} = \frac{0}{3}$$

$$p(read|Cher) = \frac{c(Cher\quad read)}{\sum_w c(Cher\quad w)} = \frac{0}{1}$$

于是，$p(Cher\ read\ a\ book) = 0$

参数估计

问题：

数据匮乏(稀疏) (*Sparse Data*) 引起零概率问题，如何解决？

数据平滑(data smoothing)

数据平滑

数据平滑

◆数据平滑的基本思想：

调整最大似然估计的概率值,使零概率增值，使非零概率下调，“劫富济贫”，消除零概率，改进模型的整体正确率。

◆基本目标：测试样本的语言模型困惑度越小越好。

◆基本约束：$\sum_{w_i} p(w_i \mid w_1, w_2, \cdots, w_{i-1}) = 1$

数据平滑

- **困惑度的定义：**

对于一个平滑的 ***n*-gram**，其概率为 $p(w_i \mid w_{i-n+1}^{i-1})$ ，可以计算句子的概率：

$$p(s) = \prod_{i=1}^{m+1} p(w_i \mid w_{i-n+1}^{i-1})$$

假定测试语料 T 由 l_T 个句子构成 $(t_1, \ldots t_{l_T})$ ，则整个测试集的概率为：

$$p(T) = \prod_{i=1}^{l_T} p(t_i)$$

数据平滑

模型 $p(w_i \mid w_{i-n+1}^{i-1})$ 对于测试语料的交叉熵：

$$H_p(T) = -\frac{1}{W_T}\log_2 p(T)$$

其中，W_T 是测试文本 T 的词数。

模型 p 的困惑度 $PP_p(T)$ 定义为：$PP_P(T) = 2^{H_p(T)}$

n-gram 对于英语文本的困惑度范围一般为 50 ~ 1000，对应于交叉熵范围为 6 ~ 10 bits/word。

数据平滑

◆数据平滑方法

(1) 加1法(Additive smoothing)

基本思想：每一种情况出现的次数加1。

例如，对于 *uni*-gram，设 w_1, w_2, w_3 三个词， 概率分别为：1/3, 0, 2/3，加1后情况？

2/6, 1/6, 3/6

数据平滑

对于2-gram 有：

$$p(w_i \mid w_{i-1}) = \frac{1 + c(w_{i-1}w_i)}{\sum_{w_i}[1 + c(w_{i-1}w_i)]}$$

$$= \frac{1 + c(w_{i-1}w_i)}{|V| + \sum_{w_i} c(w_{i-1}w_i)}$$

其中，V 为被考虑语料的词汇量（全部可能的基元数）。

数据平滑

在前面 3 个句子的例子中,

$$p(Cher\ read\ a\ book) =$$
$$p(Cher|<BOS>) \times p(read|Cher) \times p(a|read) \times p(book|a) \times p(<EOS>|book)$$

<BOS>John read Moby Dick<EOS>
<BOS>Mary read a different book<EOS>
<BOS>She read a book by Cher<EOS>

原来:

$p(Cher|<BOS>) = 0/3$
$p(read|Cher) = 0/1$
$p(a|read) = 2/3$
$p(book|a) = 1/2$
$p(<EOS>|book) = 1/2$

数据平滑

词汇量：$|V|=11$

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

平滑以后：

$p(Cher|<BOS>) = (0+1)/(11+3) = 1/14$

$p(read|Cher) = (0+1)/(11+1) = 1/12$

$p(a|read) = (1+2)/(11+3) = 3/14$

$p(book|a) = (1+1)/(11+2) = 2/13$

$p(<EOS>|book) = (1+1)/(11+2) = 2/13$

$$p(Cher\ read\ a\ book) = \frac{1}{14}\times\frac{1}{12}\times\frac{3}{14}\times\frac{2}{13}\times\frac{2}{13}\approx 0.00003$$

数据平滑

<BOS>John read Moby Dick<EOS>
<BOS>Mary read a different book<EOS>
<BOS>She read a book by Cher<EOS>

同理，对于句子*John read a book* 数据平滑后：

$p(John|<BOS>) = 2/14, \quad p(read|John) = 2/12,$

$p(a|read) = 3/14, \quad p(book|a) = 2/13, \quad p(<EOS>|book) = 2/13$

于是，$p(John\ read\ a\ book) =$
$p(John|<BOS>) \times p(read|John) \times p(a|read) \times p(book|a) \times p(<EOS>|book)$

$$= \frac{2}{14} \times \frac{2}{12} \times \frac{3}{14} \times \frac{2}{13} \times \frac{2}{13} \approx 0.0001$$

数据平滑

(2) 减值法/折扣法(Discounting)

基本思想：修改训练样本中事件的实际计数，使样本中(实际出现的)不同事件的概率之和小于1，剩余的概率量分配给未见概率。

数据平滑

①Good-Turing 估计

I. J. Good 于1953 年引用 Turing 的方法来估计概率分布。

假设 N 是原来训练样本数据的大小， n_r 是在样本中正好出现 r 次的事件的数目(此处事件为 n-gram)，即出现 1 次的n-gram有 n_1个，出现 2 次的 n-gram 有n_2个, ……，出现 r 次的有 n_r 个。

数据平滑

那么，$N=\sum_{r=1}^{\infty}n_r r=\sum_{r=0}^{\infty}(r+1)n_{r+1}$

设：原先出现*r*次的*n*-gram在平滑后出现*r**次

则 $N=\sum_{r=0}^{\infty}n_r r^*$ 则总数不变 $\sum_{r=0}^{\infty}n_r r^*=\sum_{r=0}^{\infty}(r+1)n_{r+1}$

所以，$r^*=(r+1)\frac{n_{r+1}}{n_r}$

那么，Good-Turing 估计在样本中出现 *r* 次的事件的平滑后的概率为：

$$p_r=\frac{r^*}{N}$$

数据平滑

实际应用中，一般直接用 n_{r+1} 代替 $E(n_{r+1})$，n_r 代替 $E(n_r)$。这样，原训练样本中所有事件的概率之和为：

$$\sum_{r>0} n_r \times p_r = 1 - \frac{n_1}{N} < 1$$

因此，有 $\frac{n_1}{N}$ 的剩余的概率量就可以均分给所有的未见事件 $(r = 0)$。

Good-Turing 估计适用于大词汇集产生的符合多项式分布的大量的观察数据。

数据平滑

举例说明：假设有如下英语文本，估计 2-gram概率：

<BOS>John read Moby Dick<EOS>
<BOS>Mary read a different book<EOS>
<BOS>She read a book by Cher<EOS>
……

从文本中统计出不同 2-gram 出现的次数：

<BOS> John	**15**
<BOS> Mary	**10**
……	
read Moby	**5**
……	

数据平滑

假设要估计以 read 开始的 2-gram 概率，列出以read开始的所有 2-gram，并转化为频率信息：

$$r^* = (r+1)\frac{n_{r+1}}{n_r}$$

r	n_r	r^*
1	2053	0.446
2	458	1.25
3	191	2.24
4	107	3.22
5	69	4.17
6	48	5.25
7	36	保持原来的计数7

$$r^* = (1+1)\frac{458}{2053}$$

因为$n_{r+1} = 0$

数据平滑

得到 r^* 后，概率如下：

$$p_r = \frac{r^*}{N}$$

其中，N 为以 read 开始的2-gram的总数(样本空间)，即read出现的次数。

那么，以 read开始，没有出现过的 2-gram的概率总和为：

$$p_0 = \frac{n_1}{N}$$

以 read作为开始，没有出现过的 2-gram的个数等于：

$$n_0 = |V_T| - \sum_{r>0} n_r$$

其中，$|V_T|$ 为语料的词汇量。

数据平滑

那么，没有出现过的那些 以 read 为开始的2-gram的概率平均为：$\frac{p_0}{n_0}$

注意：$\sum_{r=0}^{7} p_r \neq 1$

因此，需要归一化处理：

$$\hat{p}_r = \frac{p_r}{\sum_r p_r}$$

r	n_r	r^*
1	2053	0.446
2	458	1.25
3	191	2.24
4	107	3.22
5	69	4.17
6	48	5.25
7	36	—

数据平滑

②Back-off (后备/后退)方法

S. M. Katz 于 1987 年提出，所以又称 Katz 后退法。

基本思想：当某一事件在样本中出现的频率大于阈值K (通常取 K 为0或1)时，运用最大似然估计的减值法来估计其概率，否则，使用低阶的，即 (n-1)gram 的概率替代 n-gram 概率，而这种替代需受归一化因子α 的作用。

另一种理解: 对于每个计数 $r > 0$ 的n元文法的出现次数减值，把因减值而节省下来的剩余概率根据低阶的 (n-1)gram 分配给未见事件。

数据平滑

③绝对减值法 (Absolute discounting)

Hermann Ney 和 U. Essen 1993年提出。

基本思想：从每个计数 r 中减去同样的量，剩余的概率量由未见事件均分。

设 R 为所有可能事件的数目(当事件为 n-gram 时，如果统计基元为词，且词汇集的大小为 L，则 $R=L^n$)。

数据平滑

那么，样本出现了 r 次的事件的概率可以由如下公式估计：

$$p_r = \begin{cases} \dfrac{r-b}{N} & \text{当 } r>0 \\ \dfrac{b(R-n_0)}{Nn_0} & \text{当 } r=0 \end{cases}$$

其中，n_0 为样本中未出现的事件的数目。b 为减去的常量，$b \le 1$。

$b(R-n_0)/N$ 是由于减值而产生的剩余概率量。

b 为自由参数，可以通过留存数据(heldout data)法求得 b 的上限为：

$$b \le \frac{n_1}{n_1+2n_2} < 1$$

数据平滑

④线性减值法 (Linear discounting)

基本思想：从每个计数 r 中减去与该计数成正比的量(减值函数为线性的)，剩余概率量 α 被n_0个未见事件均分。

$$p_r = \begin{cases} (1-\alpha)\dfrac{N_r}{N} & \text{当 } r > 0 \\ \dfrac{\alpha}{n_0} & \text{当 } r = 0 \end{cases}$$

自由参数 α 的优化值为：$\dfrac{n_1}{N}$

绝对减值法产生的n-gram 通常优于线性减值法。

数据平滑

◆ 四种减值法的比较

- Good-Turing 法：对非0事件按公式削减出现的次数，节留出来的概率均分给0概率事件。
- Katz 后退法：对非0事件按Good-Turing法计算减值，节留出来的概率按低阶分布分给0概率事件。
- 绝对减值法：对非0事件无条件削减某一固定的出现次数值，节留出来的概率均分给0概率事件。
- 线性减值法：对非0事件根据出现次数按比例削减次数值，节留出来的概率均分给0概率事件。