语言模型

刘挺
哈工大信息检索研究室
2004年春

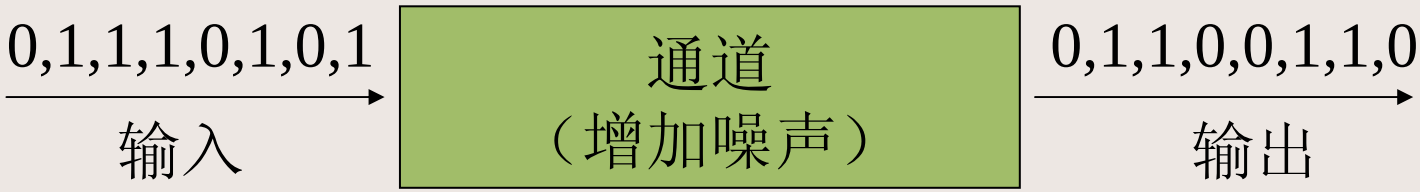
大纲

- 概述
- 参数估计
- 基本平滑算法
- 其它平滑算法

概述

噪声通道模型

- 原型



- 模型：出错的概率
- 举例：p(0|1)=0.3, p(1|1)=0.7, p(1|0)=0.4, p(0|0)=0.6
- 任务是：
  - 已知带有噪声的输出
  - 想知道输入是什么（也称为：Decoding)

噪声通道的应用

- OCR
  - 文本→打印(引入噪声), 扫描→图像
- 手写识别
  - 文本→神经肌肉(引入噪声), 扫描→图像
- 语音识别
  - 文本→朗读(引入噪声) →声学波形
- 机器翻译
  - 目标语言→翻译(引入噪声) →源语言
- 其它：词性标注
  - 词性序列→选择词形→文本

噪声通道：黄金规则

- 适用于OCR，手写识别，语音识别，机器翻译，词性标注等各个问题
- 贝叶斯公式：$P(A|B)=P(B|A)P(A)/P(B)$
- $A_{best}=\text{argmax}_A\ P(B|A)P(A)$
- P(B|A)是声学/图像/翻译等模型
  - 在不同领域用不同的术语来描述
- P(A)是语言模型

什么是语言模型(Language Model)

- 语言模型是用来计算一个句子的概率的概率模型
  - 例如：$P(w_1,w_2,\ldots,w_n)$
- 语言模型的用途
  - 决定哪一个词序列的可能性更大
  - 已知若干个词，预测下一个词
- 应用
  - 语音识别
  - 机器翻译
  - 上下文敏感的拼写检查

应用于语音识别

- 有的词序列听起来很像，但并不都是正确的句子
  - 例子1：
    - I went to a party. √
    - Eye went two a bar tea.
  - 例子2：
    - 你现在在干什么？ √
    - 你西安载感什么？

应用于机器翻译

- 给定一个汉语句子
  - 例如：王刚出现在电视上。
  - 英文译文：
    - Wang Gang appeared in TV.
    - In Wang Gang appeared TV.
    - Wang Gang appeared on TV. √

应用于拼写检查

- 举例
  - 汉语
    - 我自己知道 √
    - 我自已知道
  - 英语
    - Wang Gang appeared on TV. √
    - Wang Gang appeared of TV.

参数估计

完美的语言模型

- 对于词序列$W=w_1,w_2,\ldots,w_n$
- 如何计算$p(W)$?
- 根据链式规则：
  $p(W)=p(w_1)p(w_2|w_1)\ldots p(w_n|w_1,\ldots,w_{n-1})$
- 即使对于很小的n，上面的理想公式也很难计算，因为参数太多

马尔科夫链

- 有限的记忆能力
  - 不考虑太“老”的历史
  - 只记住前k个词$w_1,\ldots,w_k$
  - 称为k阶马尔科夫近似
- $p(W)=\prod_{i=1\ldots d}p(w_i|w_{i-k},\ldots,w_{i-1})$, $d=|W|$

N元语言模型

- n-1阶马尔科夫近似称为n元语言模型(LM, Language Model)
  - $p(W)=\prod_{i=1\ldots d}p(w_i|w_{i-n+1},\ldots,w_{i-1})$, $d=|W|$
- n越大，需要估计的参数越多，假设词汇量为20,000

模型	需要的参数数量
0阶(一元Unigram)	20,000
1阶(二元bigram)	20,000*19,999 = 400 million
2阶(三元trigram)	$20,000^2$*19,999 = 8 trillion
3阶(四元four-gram)	$20,000^3$*19,999 = $1.6*10^{17}$

语言模型的讨论

- n多大？
  - 理论上讲，越大越好
  - 经验值：3，trigram用的最多
    - four-gram需要太多的参数，很难估计了
- 目前一般直接计算词形，不进行语言学处理，如形态还原等
- 可靠性(Reliability)和可区别性(Discrimination)成反比，需要折中
  - n越大，区别力越大；n越小，可靠性越高

可靠性和区别性

- 可靠性(Reliability)和可区别性(discrimination)
- 为了有效地推导一个特征，我们希望通过模型的其它特征来预测它，把这些特征分成等价类便于我们预测新的数据。
- 分类特征越多，对未知分布的目标特征的预测就更精确，即有较好的可区别性，但是这样对每一个分类其实例就较少，统计的数据就不可靠，所以在划分等价类时要在可靠性和可区别性之间找一个折衷点。

长度问题

- $\forall n; \sum_{w \in \Omega n} p(w)=1 \Rightarrow$
- $\sum_{n=1\ldots\infty} \sum_{w \in \Omega n} p(w) >> 1 (\rightarrow\infty)$
- 我们试图对所有的词序列建立模型
  - 对于固定长度的任务，没有问题，n一旦固定，累计和为1
    - 比如Tagging, 手写识别等
  - 对于变长的任务，需要对比较短的巨资进行折扣
- 一般模型
  - 对于长度为n的词序列
    - $P'(w)=\lambda_n p(w), \sum_{n=1\ldots\infty} \lambda_n=1$
    - $\sum_{n=1\ldots\infty} \sum_{w \in \Omega n} p'(w)=1$
    - 从数据中估计λ_n

参数估计

- 参数：用来计算p(w|h)的数值
- 从数据中得到
- 数据准备
  - 去掉格式符号
  - 定义词的边界
  - 定义句子边界(插入<s>和</s>等记号)
  - 字母的大小写（保留、忽略或者智能识别）
  - 数字（保留、替换为<num>等）

最大似然估计

- 最大似然估计MLE
  - 是对训练数据的最佳估计
- 从训练数据T中获得Trigrams
  - 统计T中三个词连续出现的次数$C_3(w_{i-2},w_{i-1},w_i)$
  - 统计T中两个词连续出现的次数$C_2(w_{i-2},w_{i-1})$
- $p_{MLE}(w_i|w_{i-2},w_{i-1}) = C_3(w_{i-2},w_{i-1},w_i) / C_2(w_{i-2},w_{i-1})$

MLE不适合用于NLP

- MLE选择的参数使训练语料具有最高的概率，它没有浪费任何概率在于没有出现的现象中
- 但是MLE通常是不适合NLP的统计语言推导的，因为数据稀疏，如果一个为0，会向下传播…
- 一个例子说明数据稀疏：从IBM Laser Patent Text语料中1.5 Million 的词进行训练，在同一语料中的测试文本中，新出现23%的trigram tokens.

举例1

- p(z|xy)=?
- 假设训练语料为：
  … xya …; … xyd …; … xyd …

  xyz没有出现过
- 我们能够说：
  – p(a|xy)=1/3, p(d|xy)=2/3, p(z|xy)=0/3吗？
- 不能，因为xyz可能是一个常见的组合，但在现有的训练集中不应有的缺失了

分析

- 被除数越小，越不可靠
  - 1/3可能太高, 100/300可能是对的
- 除数越小，越不可靠
  - 1/300可能太高，100/30000可能是对的

字符语言模型

- 使用单独的字符而不是词
- 使用相同的公式和方法
- 可以考虑使用4-gram，5-gram，因为数据比较充足
- 对交叉语言的比较很有用
- 基于字和基于词的交叉熵的换算关系
  - $H_S(p_c) = H_S(p_w)$ / 句子S中的平均词长

举例2

- 训练数据:
  - <s0> <s> He can buy you the can of soda </s>
  - Unigram: (8 words in vocabulary)
    - **p_1(He) = p_1(buy) = p_1 (you) = p_1 (the) = p_1(of) = p_1(soda)= .125, p_1(can) = .25**
  - Bigram:
    - **p_2(He|<s>) = 1, p_2(can|He) = 1, p_2(buy|can) = .5, p_2(of|can) = .5, p_2(you |buy) = 1,...**
  - Trigram:
    - **p_3(He|<s0>,<s>) = 1, p_3(can|<s>,He) = 1, p_3(buy|He,can) = 1, p_3(of|the,can)= 1, ..., p_3(</s>|of,soda) = 1.**
  - Entropy: $H(p_1) = 2.75$, $H(p_2) = 1$, $H(p_3) = 0$

交叉熵

- 交叉熵
  - S = <s0> <s> It was the greatest buy of all </s>
- $H_S(p_1) = H_S(p_2) = H_S(p_3) = \infty$，原因是：
  - 所有的unigrams除了p_1(the), p_1(buy), and p_1(of) 都是0
  - 所有bigram的概率都是 0.
  - 所有trigram的概率都是 0.
- 我们希望使每个概率都是非零的

零概率问题

- 原始的Trigram模型估计
  - 一定会有很多概率为0的情况
    - 因为参数空间太大，trigram:8T，而数据只有1G
  - 哪些参数真的应该是0呢？
    - 理想情况是：最低频的trigram也应该出现几次，以便把它的概率和其它trigram的概率区别开来
    - 但是理想情况不会发生，到底需要多少数据，我们不知道
  - 我们必须去除概率为0的情况
    - 包括：p(w|h)=0，或者p(h)=0

为什么我们需要非零的概率？

- 避免无穷大的交叉熵
  - 当测试数据中出现了一个在训练数据中没有出现过的事件，就会发生$H(p)=\infty$的情况
- 使系统更健壮
  - 低频的估计值
    - 更加细腻(detailed)，但相对来说很少出现
  - 高频的估计值
    - 更可靠但是不够细腻

基本平滑算法

避免零概率：数据平滑

- p’(w) ≈p(w), 但p’(w)≠0
- 对一些p(w)>0,生成p’(w)<p(w)

$$\sum_{w \in discounted} (p(w) - p'(w)) = D$$

- 分配D给所有概率为0的w: p’(w)>p(w)=0
  - 可能对于概率值较低的词也作调整
- 可能有些w: p’(w)=p(w)
- 务必确保 $\sum_{w \in \Omega} p'(w) = 1$
- 有许多数据平滑的方法

折扣discounting

- 回退Back-off
  - 如果n-gram的值为零，则用n-1 gram来计算
- 平滑Smoothing
  - 将MLE方法与其它方向相混合，保证没有0概率的值

加1平滑

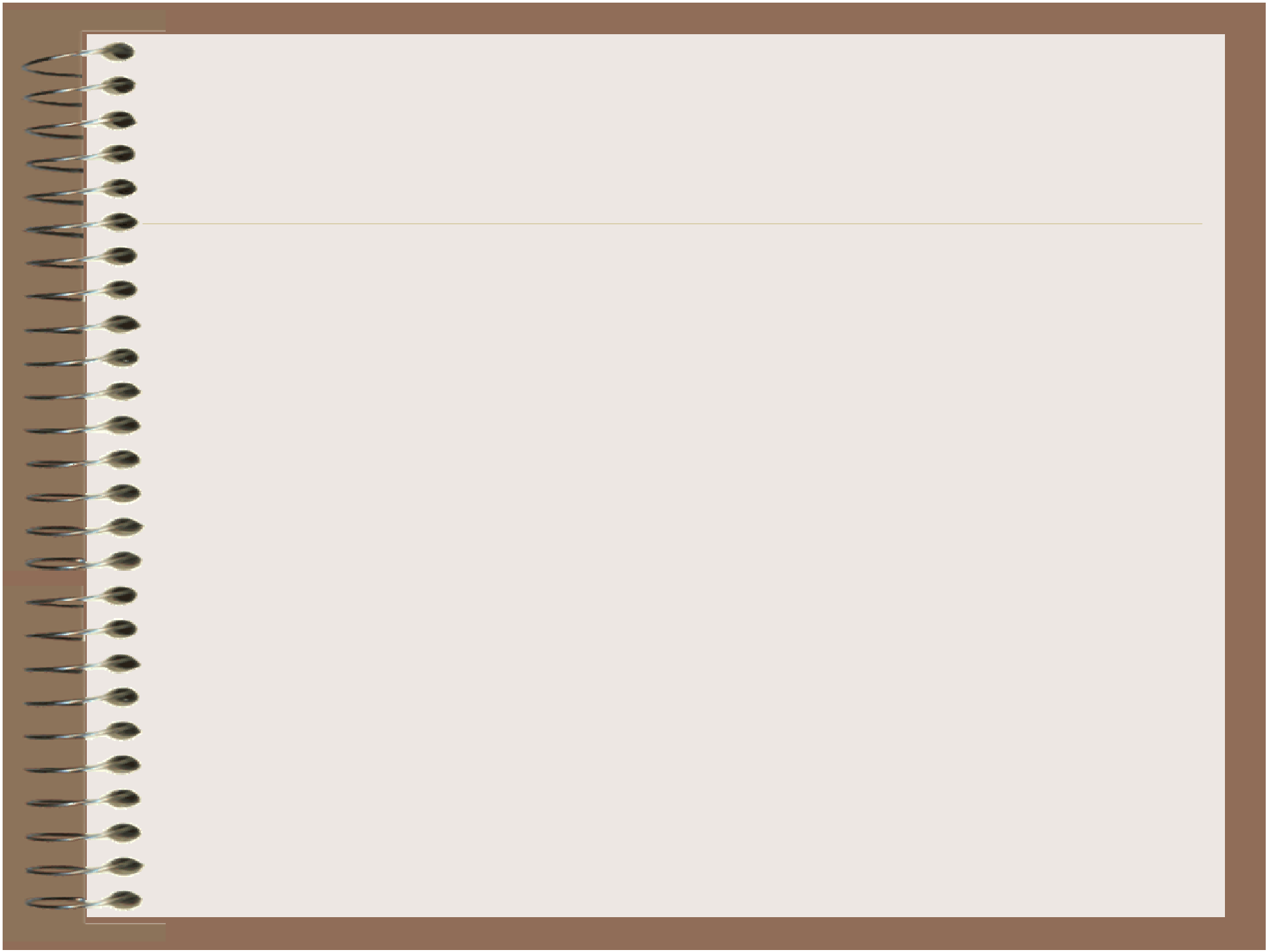
- 最简单，但不是真的能用
  - T:训练数据,V:词表,w: 词

    预测 p’(w|h)=(c(h,w)+1)/(c(h)+|V|)

    特别:非条件分布时p’(w)=(c(w)+1)/(|T|+|V|)
  - 问题：经常会|V|>c(h)，甚至|V|>>c(h)
- 举例：T: <s>what is it what is small? |T|=8
  - V={what,is,it,small,?,<s>,flying,birds,are,a,bird,.}, |V|=12
  - p(it)=0.125, p(what)=0.25, p(.)=0, p(what is it?)=$0.25^2*0.125^2 \approx 0.001$

    p(it is flying.)=$0.125*0.25*0^2=0$
  - p’(it)=0.1, p’(what)=0.15,p’(.)=0.05, p’(what is it?)=$0.15^2*0.1^2 \approx 0.0002$

    p’(it is flying.)=$0.1*0.15*0.05^2 \approx 0.00004$



Add one举例

Vocabulary Size (V) = 10,543

$$P_{+1}(\text{not}|\text{they,do}) = \frac{C(they, do, not)+1}{C(they, do)+10{,}543} = \frac{8}{22+10{,}543} = 0.000757$$

$$P_{+1}(\text{offer}|\text{they,do}) = \frac{C(they, do, like)+1}{C(they, do)+10{,}543} = \frac{1}{22+10{,}543} = 0.0000947$$

$$P_{+1}(\text{have}|\text{they,do}) = \frac{C(they, do, have)}{C(they, do)+10{,}543} = \frac{3}{22+10{,}543} = 0.000284$$

小于1平滑

- 加入λ系数

  -T:训练数据,V:词表,w: 词

  预测 $p'(w|h)=(c(h,w)+\lambda)/(c(h)+\lambda|V|)$, $\lambda<1$

  特别:非条件分布时$p'(w)=(c(w)+\lambda)/(|T|+\lambda|V|)$
- 举例：T: <s>what is it what is small? |T|=8
  - V={what,is,it,small,?,<s>,flying,birds,are,a,bird,.}, |V|=12
  - p(it)=0.125, p(what)=0.25, p(.)=0, p(what is it?)=$0.25^2 * 0.125^2 \approx 0.001$

    p(it is flying.)=$0.125*0.25*0^2=0$
  - 取 λ=0.1
  - p'(it)=0.12, p'(what)=0.23,p'(.)=0.01,
  - p'(what is it?)=$0.23^2*0.12^2 \approx 0.0007$

    p'(it is flying.)=$0.12*0.23*0.01^2 \approx 0.000003$

Good-Turing

- 适用于评估大规模的数据
  - 相似点：

    $p_r(w)=(c(w)+1)*N(c(w)+1)/(|T|*N(c(w)))$,

    其中：N(c)是数目为c的词的数量

    特别：对于训练集中没有出现的词，c(w)=0

    $p_r(w)=N(1)/(|T|*N(0))$
  - 有利于数量少的词语（<5-10, 但N(c)较大）
    - “调富济贫”
  - 归一化（可以得到 $\sum_w p'(w)=1$ ）

Good-Turing: 举例

- 例如：记住： $p_r(w)=(c(w)+1)*N(c(w)+1)/(|T|*N(c(w)))$
  - T: <s>what is it what is small? |T|=8
  - V={what,is,it,small,?,<s>,flying,birds,are,a,bird,.}, |V|=12

  $p(it)=0.125, p(what)=0.25, p(.)=0, p(what\ is\ it?)=0.25^2*0.125^2 \approx 0.001$

  $p(it\ is\ flying.)=0.125*0.25*0^2=0$
  - 重新计算（N(0)=6,N(1)=4,N(2)=2,N(i)=0 当 i>2）:

    $P_r(it)=(1+1)*N(1+1)/(8*N(1))=2*2/(8*4)=0.125$

    $P_r(what)=(2+1)*N(2+1)/(8*N(2))=3*0/(8*2)=0$: keep orig. p(what)

    $P_r(.)=(0+1)*N(0+1)/(8*N(0))=1*4/(8*6)\approx 0.083$
  - 归一化（除以 1.5= $\sum_{w\in|V|} p_r(w)$ ）并计算

    $p'(it) \approx 0.08,\ p'(what) \approx 0.17,\ p'(.) \approx 0.06$

    $p'(what\ is\ it?)=0175^2*0.08^2 \approx 0.0002$

    $p'(it\ is\ flying.) \approx 0.08*0.17*0.06^2 \approx 0.00004$

典型n-gram语言模型的平滑

- 采用 $\lambda = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$:

$$p'_\lambda(w_i \mid w_{i-2}, w_{i-1}) = \lambda_3 p_3(w_i \mid w_{i-2}, w_{i-1}) + \lambda_2 p_2(w_i \mid w_{i-1}) + \lambda_1 p_1(w_i) + \lambda_0 / |V|$$

- 归一化：

  $\lambda_i > 0$, $\sum_{i=0..n} \lambda_i = 1$ 就可以了（ $\lambda_0 = 1 - \sum_{i=1..n} \lambda_i$ ）（n=3）

- 极大似然估计
  - 固定p_3, p_2, p_1和$|V|$，根据训练数据确定参数
  - 再寻找一组$\{\lambda_i\}$使得交叉熵达到最小

    （使数据的概率达到最大）：

$$-(1/|D|)\sum_{i=1..|D|} \log_2(p'_\lambda(w_i \mid h_i))$$

Held-out Data

- 使用什么数据？
  - 试用训练数据T：但是我们总会得到 $\lambda_3=1$
    - 为什么？
    - 在向量 λ 上最小化$H_T(p'(\lambda))$, $p'(\lambda)=\lambda_3 p_{3T}+\lambda_2 p_{2T}+\lambda_1 p_{1T}+\lambda_0/|V|$
      - 记住$H_T(p'_\lambda)=H(p_{3T})+D(p_{3T}||p'_\lambda)$;(p_{3T} fixed→H(p3T)fixed,best)
      - 满足$D(p_{3T}||p'_\lambda)=0$的p'_λ可以使得$H_T(p'_\lambda)$达到最小
      - 解是p_{3T}（因为D(p||p)=0）
      - If $\lambda_3=1$,必有$p'_\lambda = 1*p_{3T}+ 0*p_{2T}+ 0*p_{1T}+ 0/|V|= p_{3T}$
- 所以：不要采用训练数据来估计 λ！
  - 必须留出部分训练数据（Held out data, H）
  - 剩下的数据为真实原始的训练数据(training data,T)
  - 测试数据 S（例如为了比较）：仍然是不同的数据！

公式

- 重复：在 λ 上最小化 $-(1/|H|)\sum_{i=1..|H|}\log_2(p'_\lambda(w_i|h_i))$

$$p'_\lambda(w_i|w_{i-2},w_{i-1})=\lambda_3 p_3(w_i|w_{i-2},w_{i-1})+\lambda_2 p_2(w_i|w_{i-1})+\lambda_1 p_1(w_i)+\lambda_0/|V|$$

- λ 的期望数值：j=0..3

$$c(\lambda_j)=\sum_{i=1..|H|}(\lambda_j p_j(w_i|h_i)/p'_\lambda(w_i|h_i))$$

- “Next λ”：j=0..3

$$\lambda_{j,next}=c(\lambda_j)/\sum_{k=0..3}(c(\lambda_k))$$

EM平滑算法

1、从某些 λ 开始，如 $\lambda_j>0$,所有$j\in 0..3$
2、计算每个 λ_j的期望数值
3、采用“Next λ”公式计算的 λ_j新集合
4、返回2，除非遇到终止条件

- 终止条件为： λ 收敛
  简单设定一个 ε，当step3中对每个j都有
  $|\lambda_j - \lambda_{j,next}| < \varepsilon$ 时终止

简单实例

- 原始分布（unigram，对均匀分布平滑）
  p(a)=0.25,p(b)=0.5,p(α)=1/64, α ∈{c..r} ,剩下的s,t,u,v,w,x,y,z均为0
- Heldout数据：baby; 采用一个λ的集合（ λ_1:unigram, λ_0:均匀分布）
- 开始于 λ_1=0.5；

  $p'_\lambda(b)=0.5*0.5+0.5/26=0.27$

  $p'_\lambda(a)=0.5*0.25+0.5/26=0.14$

  $p'_\lambda(y)=0.5*0+0.5/26=0.02$

  $c(\lambda_1)=0.5*0.5/0.27+0.5*0.25/0.14+0.5*0.5/0.27+0.5*0/0.2=2.72$

  $c(\lambda_0)=0.5*0.04/0.27+0.5*0.04/0.14+0.5*0.04/0.27+0.5*0.04/0.02=1.28$

  归一化： $\lambda_{1,next}=0.68$, $\lambda_{0,next}=0.32$
- 返回step2(重新计算p'_λ，然后 $c(\lambda_1)$，…）。结束结束条件为当前的λ与前一组λ相差很小（比如，<0.01）

其它平滑算法

线性插值

- 在简单线性插值中，权值仅仅是一个数，我们可以定一个更通用的和powerful的模型，其系数值是历史的函数，通用线性插值的形式是：

and

$$P_{li}(w \mid h) = \sum_{i=1}^{k} \lambda_i(h) P_i(w \mid h)$$

$$0 \le \lambda_i(h) \le 1 \qquad \sum_i \lambda_i(h) = 1$$

线性插值

- “桶”平滑
  - 根据历史频率信息，采用几个 λ 向量代替一个
    - 例如：h=(micrograms,per) 我们可以得到

      λ (h)=(0.999,0.0009,0.00009,0.00001) (因为“cubic”是唯一可以紧随的词语)
  - 事实上：不存在一个单独集合对应每个历史数据，但是可以对应近似的历史数据:

    λ (b(h)),其中b: $V^2 \rightarrow N$ (对应三元模型)

    b 根据可靠性来区分历史数据（频率信息）

“桶”平滑算法

- 利用held out数据确定“桶”函数
  - 预先设定需要的数量，如1000个桶
  - 计算1个桶中的历史频率总数 $f_{max}(b)$
  - 根据最高频bigram逐渐填满所有桶，使得频率总和不超过$f_{max}(b)$（可能会略微超过1000个桶）
- 根据桶数平分held out数据
- 对各个桶及其数据应用以上算法

绝对折扣
Absolute discounting

- 对所有的非0 MLE频率减一个较小的常数，把获得的频率分不到不出现的事件上。

$$C(w_1 \ldots w_n) = r$$

$$P_{abs}(w_1 \ldots w_n) = \begin{cases} (r-\delta)/N & if \quad r > 0 \\ \dfrac{(B-N_0)\delta}{N_0 N} & otherwise \end{cases}$$

线性折扣
linear discounting

- 非0的乘一个系数，把获得的概率分不到不出现的事件上

$$C(w_1 \ldots w_n) = r$$

$$P(w_1 \ldots w_n) = \begin{cases} (1-\alpha) r / N & if \quad r > 0 \\ \alpha / N_0 & otherwise \end{cases}$$

Katz's 回退方法 Back-off

- Katz的n-gram回退模型通过前面短一些的历史对当前n-gram进行估计。估计式如下：

$$P_{bo}(w_i \mid w_{i-n+1}..w_{i-1}) = \begin{cases} (1-d_{w_{i-n+1}..w_{i-1}}) \dfrac{C(w_{i-n+1}..w_i)}{C(w_{i-n+1}..w_{i-1})} & if \;\; C(w_{i-n+1}..w_i) > k \\ \alpha_{w_{i-n+1}..w_{i-1}} P_{bo}(w_i \mid w_{i-n+2}..w_{i-1}) & otherwise \end{cases}$$

划分等价类的其它方法

- Stemming
  - Computer
  - Computing
  - Compute
  - ……
- 语义类Class-based LM
  - 时间类
  - 人名类
  - 地点类
  - 机构名类
  - ……

对语言模型进行评价

- 最佳方法:
  - 将语言模型用于某一个应用中，例如拼写检查、机器翻译等
- 混乱度(Perplexity):
  - 给测试集赋予较高的概率值的语言模型较好

$$Perplexity = \left[\prod_{i=1}^{n} P\left(w_i \mid w_{1:i-1}\right) \right]^{-\frac{1}{n}}$$