《数学之美》读书笔记

语意模型--马尔可夫假设

- 每个词的出现概率只和前一个词有关，根据大数定律，两者共同出现的概率等于语料库中两者一起出现的次数除以语料库大小。
- n阶马尔可夫假设(n元模型）：出现概率与前面n-1个词有关
- 解决语料库中零概率问题的方法：古德－图灵估计
  - 假定r较小时统计可能不可靠，因此出现r次的词不应有那么高的权重
  - 原本的计算概率方式：
    - $N = \sum_{r=1}^{\infty} r N_r$
    - 其中出现r次的词有Nr个,N为语料库大小
  - 现在的方式：
    - $d_r = (r+1) \cdot N_{r+1} / N_r$
    - 其中d_r是估计的概率
- 局限：
  - 复杂度大
  - 难找到上下文关联

隐含马尔科夫模型

- 马尔科夫链
  - 一个状态本该和以前所有状态有关，但是马尔科夫链中只考虑上**一个**状态
  - 实质是一个有穷状态机，随机选择一个状态开始，按自身规则随机转移到下一个状态
  - 组成成分：
    - 状态
    - 状态转移概率
    - 结果
    - 产生结果概率（独立输出假设）
- 隐含马尔科夫模型：缺少上述成分中的一个
  - 三个基本问题
    1. 给定模型，计算产生某个特定输出的概率: forward-backward算法

2. 给定模型和输出序列，找到最可能的状态序列: 维特比算法
3. 给定足够观测数据，估计模型的各个参数（如下）

- 需要求解
  1. $P(o_t|s_t) = \frac{P(o_t,s_t)}{P(s_t)}$--------（1）
  2. $P(s_t|s_{t-1}) = \frac{P(s_{t-1},s_t)}{P(s_{t-1})}$--------（2）
     其中s_t为第t个状态，o_t为第t个输出
- 有监督：如果知道s_t出现次数，每次经过这个状态时产生的输出，那么$P(o_t|s_t) = \frac{\#(o_t,s_t)}{\#(s_t)}$
- 无监督：使用鲍姆-韦尔奇算法：
  1. 找到一组能够产生输出序列O的参数，这个初始模型记为M_{θ_0}
  2. 为了找到更好的模型，我们假定已经解决了问题1，2，可以算出**这个模型产生O的概率**$P(O|M_{\theta_0})$,以及**这个模型产生O的所有途径及其概率**
  3. 这些可能路径可以看作***标记的训练数据***，根据（1）（2）式迭代至收敛

信息的度量

- 信息熵

那么如何量化信息的度量呢？来看一个例子。2010 年举行了世界杯足球赛，大家都很关心谁会是冠军。假如我错过了看世界杯，赛后我问一个知道比赛结果的观众“哪支球队是冠军”？他不愿意直接告诉我，而让我猜，并且我每猜一次，他要收一元钱才肯告诉我是否猜对了，那么我需要付给他多少钱才能知道谁是冠军呢？我可以把球队编上号，从 1 到 32，然后提问：“冠军的球队在 1-16 号中吗？”假如他告诉我猜对了，我会接着问：“冠军在 1-8 号中吗？”假如他告诉我猜错了，我自然知道冠军队在 9-16 号中。这样只需要五次，我就能知道哪支球队是冠军。所以，谁是世界杯冠军这条消息的信息量只值 5 块钱。

当然，香农不是用钱，而是用“比特”（Bit）这个概念来度量信息量。一个比特是一位二进制数，计算机中的一个字节是 8 比特。在上面的例子中，这条消息的信息量是 5 比特。（如果有朝一日有 64 支球队进入决赛阶段的比赛，那么“谁是世界杯冠军”的信息量就是 6 比特，因为要多猜一次。）读者可能已经发现，信息量的比特数和所有可能情况的对数函数 log 有关[1]。（log32 = 5，log64 = 6）

有些读者此时可能会发现实际上可能不需要猜五次就能猜出谁是冠军，因为像西班牙、巴西、德国、意大利这样的球队得冠军的可能性比日本、南非、韩国等球队大得多。因此，第一次猜测时不需要把 32 支球队等分成两个组，而可以把少数几支最可能的球队分成一组，把其他队分成另一组。然后猜冠军球队是否在那几支热门队中。重复这样的过程，根据夺冠概率对剩下的候选球队分组，直至找到冠军队。这样，也许三次或四次就猜出结果。因此，当每支球队夺冠的可能性（概率）不等时，“谁是世界杯冠军”的信息量比 5 比特少。香农指出，它的准确信息量应该是

$$H = -(p_1 \cdot \log p_1 + p_2 \cdot \log p_2 + \cdots + p_{32} \cdot \log p_{32}) \qquad (6.1)$$

其中，$p_1, p_2, \cdots, p_{32}$ 分别是这 32 支球队夺冠的概率。香农把它称为“信息熵”（Entropy），一般用符号 H 表示，单位是比特。有兴趣的读者可以推算一下当 32 支球队夺冠概率相同时，对应的信息熵等于 5 比特。有数

  - 定义：$H(X) = -\sum_{x \in X} P(x) log P(x)$
  - 冗余度：文件大小（BIT）与文件所含信息熵的差值（汉语冗余度较小）
  - 香农第一定律：**任何编码的长度不小于他的信息熵**
- 信息的作用：消除不确定性
  - 在Y的条件下X的**条件熵**:

$$H(X|Y) = -\sum_{x \in X, y \in Y} P(x,y) log P(x|y) < H(X)$$

> 所以语义分析的二元模型优于一元模型

- 互信息：对**两个随机事件相关性**的度量

$$I(X;Y) = -\sum_{x \in X, y \in Y} log \frac{P(x,y)}{P(x) \cdot P(y)}$$

$$I(X;Y) == H(X) - H(X|Y)$$

- 相对熵（交叉熵，Kullback-Leibler Divergence，KL）:衡量两个取值为正数的函数的相似性

$$KL(f(x)||g(x)) = \sum_{x \in X} f(x) \cdot log \frac{f(x)}{g(x)}$$

- 结论
  1. 函数相差越大，相对熵越大
  2. 对于概率分布/概率密度函数，如果取值大于零，相对熵可以度量两个随机分布的差异

搜索引擎

- 关键词权重的度量：TF-IDF
  - Term Frequency/Inverse Document Frequency
  - 每个词的重要程度不一样，所以对每一个词设一个权重

- 逆文本频率指数
  - 思想：一个词在越多网页里出现，他的权重就越小
  - $IDF = log(\frac{D}{D_w})$
    - 其中D是全部网页数，D_w代表关键词出现在多少个网页里
    - 这里的log是根据交叉熵来的

新闻分类问题

- 表示新闻
  - 将词汇表中的每一个词进行TF-IDF编号，这个排列当作特征向量
  - 相似性：余弦距离
- 分类
  1. 未知类别向量：自底向上不断合并最相近的，再截出想要的种类数
  2. 已知类别：直接分
- tricks
  - 大数时的余弦计算
    - 计算余弦距离时把长度乘积存起来复用
    - 只计算非零部分
    - 删除虚词
  - 位置加权
    - 标题/文本/段首段尾

矩阵分解

- 奇异值分解（Singular Value Decomposition）
  - 减小计算量和储存量
  - e.g. : 1000000*500000的大矩阵A表示有500000个词，1000000篇新闻，他可以分解为1000000*100的X矩阵（X记录每篇新闻和某个的相关度），100*100的B矩阵（B记录语义类和文章种类的相关程度），100*500000的Y矩阵（Y记录文章和主题间的相关度)的积
- 会丢失信息，但是速度快，可以和余弦方法共同使用（先粗分，再精分）

信息指纹/密码

- e.g. MD5,SHA-1
- 主要用处: 判断集合是否相同
- 相似哈希：对一篇文章的词进行TF-IDF编码，每个词算一个信息指纹，对应位为0则减w，为1则加w，最后算出的对应位大于0则化为1，小于则化为0
- 公开密钥的产生
  - 假设明码为067097101115097114（caesar的ascii码)
  - 找两个大素数P，Q
    - N = P * Q
    - M = (P - 1) * (Q - 1)
  - 找一个和M互素的E
  - 找一个D，使（E*D）% M == 1
  - 完成！E是公钥，D是私钥，N是公开的
- 加密（Y是密码）

$$Y = X^E \ mod \ N$$

- 解密

$$X = Y^D \ mod \ N$$

最大熵

需要对一个随机事件的概率分布进行预测时，我们的预测应该满足所有**已知条件**，而**对未知情况不做任何主观假设**

$$P(d|x_1, x_2, ..., x_i) = \frac{1}{Z(x_1, x_2, ..., x_i)} \cdot e^{\sum \lambda_i \cdot (x_i, d)}$$

其中d是要预测的值，x_1, x_2是i个信息，Z是归一化因子，保证概率之和为1，λ是训练的参数

- 训练:GIS(Generalized Iterative Scaling,通用迭代算法)
  1. 假定第0次迭代为等概率分布
  2. 用第n次迭代的模型估计每种信息特征在训练数据中的分布，根据差值调整参数
  3. 重复2.至收敛
  - 是一个典型的EM算法（Expectation Maximization)
  - 改进的算法：IIS（Improved Iterative Scaling）

布隆过滤器

贝叶斯网络

条件随机场

维特比算法

> 图论，动态规划

$$d(S,x_{i,m})=min(\underbrace{d(S,x_{i-1,n})}_{last\ iteration}+\underbrace{d(x_{i,m},x_{i-1,n})}_{easy\ to\ solve})$$

起点S到第i步，第m个节点的最短距离等于min『对上一步中的每个结点，min『已有的最短距离+此节点到当前步中各个节点的距离』』

期望最大化算法（EM）

- 步骤
  1. 随机挑K个点（$c_1(0),c_2(0),...,c_K(0)$，K为要划分的类的个数），作为起始的中心
  2. 计算所有点到中心的距离，将点归到最近的中心代表的那一类
  3. 重新计算中心，最简单的方法就是计算平均值
  4. 重复2，3，至收敛

收敛的必然性

我们希望同一类各个点到中心的平均距离d小，不同类之间的评价距离D大，我们希望每次迭代d变小，D变大
假设第i类中有n_i个点，点到中心的平均距离是d_i，那么

$$d=(n_1\cdot d_1+n_2\cdot d_2+...+n_k\cdot d_k)/k$$

假定第i类到第j类中心的距离为D_{ij}，并且考虑根据类里点的个数来加权平均，那么

$$D = \sum_{i,j} \frac{D_{ij} n_i n_j}{n(n-1)}$$

显然，我们易得d，D都是递减的

局部最优/全剧最优

如果我们的优化函数是凸函数则一定有全局最优（e.g.熵函数，欧式距离，而余弦距离不是）

EOF