

中文信息熵计算

陈煜磊 ZY2103502

1 问题概述

熵是一个热力学概念，用来描述系统的混乱程度。1948年，Shannon 将这个概念引入信息论中，用它来表示消息中包含信息的平均量¹。本文参考了 Brown 等人对英语信息熵上界进行估计的工作²，利用金庸小说作为语料库，建立基于词的二元语言模型，计算得中文的信息熵。

2 原理

2.1 信息熵

在一个有限字母表的平稳随机过程 $X = \{\dots X_{-2}, X_{-1}, X_0, X_1, X_2 \dots\}$ 中，令 P 为 X 的概率分布， E_P 是在 P 下的期望。定义 X 的熵：

$$H(X) \equiv H(P) \equiv -E_P \log P(X_0 | X_{-1}, X_{-2}, \dots) \quad (1)$$

$H(P)$ 也可以表示为

$$H(P) = \lim_{n \rightarrow \infty} -E_P \log P(X_0 | X_{-1}, X_{-2}, \dots, X_{-n}) = \lim_{n \rightarrow \infty} -\frac{1}{n} E_P \log P(X_1 X_2 \dots X_n) \quad (2)$$

当整个过程各态遍历时，则由 Shannon-McMillan-Breiman 定理³可得

$$H(P) = \lim_{n \rightarrow \infty} -\frac{1}{n} \log P(X_1 X_2 \dots X_n) \quad (3)$$

也就是说，当以符合分布 P 的方式抽取足够长的文本，就可以对 $H(P)$ 进行估计。即使 P 未知，也可对其进行近似，得到 $H(P)$ 的上界。定义 P 的模型 M ，则 P 的交叉熵为：

$$H(P, M) \equiv -E_P \log M(X_0 | X_{-1}, X_{-2}, \dots) \quad (4)$$

$H(P, M)$ 也可以表示为

$$H(P, M) = \lim_{n \rightarrow \infty} -E_P \log M(X_0 | X_{-1}, X_{-2}, \dots, X_{-n}) = \lim_{n \rightarrow \infty} -\frac{1}{n} E_P \log M(X_1 X_2 \dots X_n) \quad (5)$$

同理，若 P 是遍历的，有

$$H(P, M) = \lim_{n \rightarrow \infty} -\frac{1}{n} \log M(X_1 X_2 \dots X_n) \quad (6)$$

Cover 等已证明⁴，对任意唯一可解码的编码方案，有

$$E_p l(X_1 X_2 \dots X_n) \geq -E_p \log P(X_1 X_2 \dots X_n) \quad (7)$$

其中 $l(X_1 X_2 \dots X_n)$ 为编码字符串所用的位数。由 (2) 和 (7) 可得 $H(P)$ 是从 P 中提取文本的每符号平均编码位数的下界，即：

$$H(P) \leq \lim_{n \rightarrow \infty} \frac{1}{n} E_p l(X_1 X_2 \dots X_n) \quad (8)$$

$$H(P) \leq \lim_{n \rightarrow \infty} -\frac{1}{n} \log M(X_1 X_2 \dots X_n) \quad (8)$$

另一方面，使用模型 M 的算术编码方案 $l_M(x_1 x_2 \dots x_n) = \lceil -\log M(x_1 x_2 \dots x_n) + 1 \rceil$ ，其中 $\lceil r \rceil$ 表示不小于 r 的最小整数。由于交叉熵 $H(P, M)$ 是 $H(P)$ 的上界，所以 $H(P, M)$ 即由模型 M 对 P 提取文本的每符号编码位数。

$$H(P, M) = \lim_{n \rightarrow \infty} \frac{1}{n} l_M(X_1 X_2 \dots X_n) \quad (9)$$

我们将中文视为包含语料库中所有词的随机过程，由上文可以估计中文信息熵的上界：

1. 构建语言模型 M 。
2. 收集足够长测试样本，由 $H(\text{Chinese}) \leq -\frac{1}{n} \log M(\text{test sample})$ 估计中文信息熵的上界，其中 n 为测试样本中词的个数。

注意这里的语言模型 M 必须在未知测试样本的情况下构建，使用测试样本的知识会使得交叉熵显著降低。

• 2.2 语言模型

自然语言处理中的语言模型是一个计算句子（单词序列）的概率或序列中下一个单词的概率的模型。而 N-Gram 是一种经典的统计语言模型，它假设当前词只和它前面的 $n-1$ 个词有关，与更前面的词无关。

我们假设中文是关于词的 1 阶马尔科夫链，即下一个词的出现仅依赖于它前面的一个词，基于此，建立 Bigram 模型，对中文熵进行估计：

1. 基于对语料库的统计计算转移概率 $P(c_i | c_{i-1})$ ，从而建立基于词的二元语言模型 M 。
2. 使用测试样本计算 $M(X_1 X_2 \dots X_n) = P(X_1)P(X_2 | X_1) \dots P(X_n | X_{n-1})$ ，代入式 (6) 得熵的估计。

3 代码

本文基于 Python 进行熵的估计工作，文件结构如下，完整代码见附件。

```

\DL-NLP COURSE\ENTROPY
|
| main.py
| merged.txt
| preprocess.py
| testtext.txt
|
|---text
|   |
|   | 多情剑客无情剑.txt
|   | 新华社六评俄乌冲突.txt
|   |
|   |---lm
|   |   |
|   |   | 三十三剑客图.txt
|   |   | 白马啸西风.txt
|   |   | 越女剑.txt
|   |   | 雪山飞狐.txt
|   |   | 鸳鸯刀.txt
|   |
|   |---test
|   |   |
|   |   | 书剑恩仇录.txt
|   |   | 侠客行.txt
|   |   | 倚天屠龙记.txt

```

```
天龙八部.txt
射雕英雄传.txt
碧血剑.txt
神雕侠侣.txt
笑傲江湖.txt
连城诀.txt
飞狐外传.txt
鹿鼎记.txt
```

• 3.1 文本预处理

对于目录下的 16 篇以 ANSI 编码的 txt 小说，进行文本的预处理：

1. 将文本分为两类：

- 构建语言模型所需的，置于 `text\lm` 目录下；
- 熵估计所需的，置于 `text\test` 目录下。

并使用 `preprocess.merge_text()` 分别对其进行合并处理。

2. 对两类文本进行分句后（得以各句为元素的列表），对其分词：

- `lm` 文本使用 `preprocess.word_seg()` 输出为以各句为元素的列表，句中各单词以空格分隔；
- `test` 文本使用 `preprocess.word_seg_tolist()` 输出为以各句为元素的嵌套列表，各句以为各个词为元素。

• 3.2 语言模型构建

取分词完毕的 `lm` 文本作为语言模型构建的输入。使用 `sklearn` 的 `CountVectorizer()` 对文本的一元和二元词频进行统计。注意 `CountVectorizer()` 默认的 `token_pattern` 为两个字母以上，在此修改此正则表达式，以捕捉中文中一字的词语。将词频统计结果返回为 `key` 为单词，`value` 为词频的字典，供下一步使用。

```
def bigram(seg_list):
    """使用sklearn的CountVectorizer对文本的一元和二元词频进行统计"""
    vec1 = CountVectorizer(token_pattern = r"(?u)\b\w+\b", ngram_range=(1,1), min_df
= 1)
    vec2 = CountVectorizer(token_pattern = r"(?u)\b\w+\b", ngram_range=(2,2), min_df
= 1)
    d1 = vec1.fit_transform(seg_list).toarray()
    d2 = vec2.fit_transform(seg_list).toarray()
    # 对各句的频率求和
    sum1 = np.zeros(np.size(d1,1))
    sum2 = np.zeros(np.size(d2,1))
    for line in d1:
        sum1 = sum1 + line
    for line in d2:
        sum2 = sum2 + line
    # 输出key为单词，value为词频的字典
    double_prob = dict(zip(vec2.get_feature_names(), sum2))
    single_prob = dict(zip(vec1.get_feature_names(), sum1))
    return single_prob, double_prob
```

• 3.3 熵估计

对 `test` 文本中每一句，计算 $M(X_1X_2 \dots X_n) = P(X_1)P(X_2|X_1) \dots P(X_n|X_{n-1})$ ，再由 $H(P, M) = \lim_{n \rightarrow \infty} -\frac{1}{n} \log M(X_1X_2 \dots X_n)$ 得交叉熵，再在整个文本中求交叉熵的平均值。

注意这里为了避免Python的 `math.log()` 中底数过小而无法计算，这里舍弃了一小部分概率过小的句子，这可能会导致熵略微偏低。

```
def entropy_estimation(test_case, single_prob, double_prob):
    """对test_case中每一句话进行交叉熵估计"""
    sum = 0
    num_sentence = len(test_case)
    # 计算每一句话出现概率
    for sentence in test_case:
        if len(sentence) != 0:
            r = 1
            for i in range(len(sentence)-1):
                item = sentence[i] + ' ' + sentence[i+1]
                if item in double_prob:
                    temp = double_prob[item]/single_prob[sentence[i]]
                elif sentence[i] in single_prob:
                    temp = 1/single_prob[sentence[i]]
                else:
                    temp = 1/len(single_prob)
                r *= temp
            # fail safe
            if r < 10e-100:
                num_sentence -= 1
                continue
            sum += -(math.log(r,2))/len(sentence)
    return sum/num_sentence
```

4 计算数据及总结

对 `test` 文本整体求平均熵，得结果为 6.413 bit/字。再对其中部分小说分别求其平均熵，得：

文本	射雕英雄传	神雕侠侣	倚天屠龙记	书剑恩仇录
熵 (比特/字)	6.586	6.522	6.3797	6.528

可以发现射雕三部曲的确是一部不如一部了。

考虑到语言模型和测试样本都是金庸的武侠作品，而模型包含测试样本的知识可能会使得交叉熵降低。尝试使用另一部武侠小说进行测试，发现古龙的《多情剑客无情剑》的熵也仅有 6.321 bit/字。继续用文风不同的《三体》，得 7.039 bit/字，可见样本知识导致交叉熵降低的现象确实存在。再取一段[新华社六评俄乌冲突](#)作为测试样本，此时熵达到了 8.468 bit/字，可得新华社果然是 high level。

参考文献

[1] C. E. Shannon, "A Mathematical Theory of Communication", Bell Sys. Tech. Journal, vol. 27, pp. 379-423, 1948.

- [2] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. C. Lai, and R. L. Mercer, "An Estimate of an Upper Bound for the Entropy of English," *Computational Linguistics*, vol. 18, no. 1, pp. 31–40, 1992.
- [3] P. H. Algoet and T. M. Cover, "A Sandwich Proof of the Shannon-McMillan-Breiman Theorem," *The Annals of Probability*, vol. 16, no. 2, pp. 899–909, 1988.
- [4] T. M. Cover and J. A. Thomas, "Entropy, relative entropy and mutual information," *Elements of information theory*, vol. 2, no. 1, pp. 12–13, 1991.
- [5] 吴军, 王作英, 《汉语信息熵和语言模型的复杂度》, *电子学报*, 期 10, 页 69-71+86, 1996.