信源扩展

谢飏

@算海无涯-X

November 16, 2016

1. 原始方案
2. 扩展技术
3. 无限序列
4. 有限序列
5. 编码性能
6. 前沿展望

处理DNA数据

- ASCII符号格式的DNA数据:

  ```
  AGGCATTCTTTGTTACAGGATGAGAGGAGG
  CTGGCACAAGTGCAGGTCACACAGACCTTG
  CTGATAAAAGGATGAGATATGCCAGGTGTG
  GTGGCTCACACCTGTAATCCCAGCACTTTG
  GGATGCCAAGGTGGATGGATCATGAGGTCA
  GGAGTTTGAGACCAGCCTGGCCAAAGAGAC
  CAGCATGGTGAAACCCCATCTCTACTAAAA
  ATACAAAAATTGGCCAGGCGTGTGGTGGGT
  GCCTGTAATCCCAGCTACTTGGGAGGCTGA
  ```

- 可对每个符号进行编码，也即“符号—编码—符号”方案，例如:

$$C(\mathtt{A}) = 00, C(\mathtt{C}) = 01, C(\mathtt{G}) = 10, C(\mathtt{T}) = 11.$$

- 270个符号的ASCII序列需要2160位，编码后需要540位，实现了较好的数据压缩.

处理$\{0,1\}$数据

- $0,1$序列:

$$\begin{array}{c}
101011001011110010000110101010\\
110101001010110100010110101010\\
001010111010101001010110101001\\
111101001010100001011110001101
\end{array}$$

又该如何呢?

处理$\{0,1\}$数据

- 0, 1序列:

$$\begin{array}{c}
101011001011110010000110101010\\
110101001010110100010110101010\\
001010111010101001010110101001\\
111101001010100001011110001101
\end{array}$$

又该如何呢?

- 120个符号的序列需要120位, 编码也至少需要120位.

处理$\{0,1\}$数据

- 0, 1序列:

$$
\begin{array}{c}
101011001011110010000110101010 \\
110101001010110100010110101010 \\
001010111010101001010110101001 \\
111101001010100001011110001101
\end{array}
$$

又该如何呢?

- 120个符号的序列需要120位, 编码也至少需要120位.
- 似乎无法压缩?

处理$\{0, 1\}$数据

- $0, 1$序列:

$$
\begin{array}{c}
101011001011110010000110101010 \\
110101001010110100010110101010 \\
001010111010101001010110101001 \\
111101001010100001011110001101
\end{array}
$$

又该如何呢?

- 120个符号的序列需要120位, 编码也至少需要120位.
- 似乎无法压缩?
- 取值空间$\mathcal{X} = \{0, 1\}$, 字母表$\mathcal{A} = \{0, 1\}$.
- 问题在于: $\mathcal{X} = \mathcal{A}$, 符号与编码只能一一对应!

省略号的作用

- 如何表述一个长为L的全1序列(记为1^L)?

省略号的作用

- 如何表述一个长为L的全1序列(记为1^L)?
- 使用“符号—符号”编码, 但最终还是1^L(或者0^L).
- 没有改进!

省略号的作用

- 如何表述一个长为L的全1序列(记为1^L)?
- 使用“符号—符号”编码, 但最终还是1^L(或者0^L).
- 没有改进!
- 人类会用$11\cdots1$表示, 其中**省略号**非常值得借鉴.

省略号的作用

- 如何表述一个长为L的全1序列(记为1^L)?
- 使用"符号—符号"编码，但最终还是1^L(或者0^L).
- 没有改进!
- 人类会用$11\cdots1$表示，其中**省略号**非常值得借鉴.
- 例如$L=15$，那么其中$\cdot$代表1111，不妨将$\cdot$编码为0，那么编码为110001. 于是实现了压缩.

省略号的作用

- 如何表述一个长为L的全1序列(记为1^L)?
- 使用“符号—符号”编码, 但最终还是1^L(或者0^L).
- 没有改进!
- 人类会用$11\cdots1$表示, 其中**省略号**非常值得借鉴.
- 例如$L=15$, 那么其中$\cdot$代表1111, 不妨将$\cdot$编码为0, 那么编码为110001. 于是实现了压缩.
- 不妨将1^L这个序列按n个为一组进行分割, 显然每组都只能是n个连续的1(也即1^n).
- 对1^L进行编码的结果是L/n个0, 显然起到了压缩的效果.

问题表述

- 设信源不断发出消息，以n个为一组，可得到分组

$$
\begin{array}{cccc}
X_1, & X_2, & \ldots, & X_n, \\
X_{n+1}, & X_{n+2}, & \ldots, & X_{2n}, \\
\cdots & & & \\
X_{(t-1)n+1}, & X_{(t-1)n+2}, & \ldots, & X_{tn}.
\end{array}
$$

问题表述

- 设信源不断发出消息，以n个为一组，可得到分组

$$
\begin{array}{cccc}
X_1, & X_2, & \ldots, & X_n, \\
X_{n+1}, & X_{n+2}, & \ldots, & X_{2n}, \\
\cdots & & & \\
X_{(t-1)n+1}, & X_{(t-1)n+2}, & \ldots, & X_{tn}.
\end{array}
$$

- 经过分组的新信源取值空间为$\mathcal{X}^n$，这种对随机变量进行扩展的技术对应着信源的n次**扩展**(extension).

问题表述

- 设信源不断发出消息，以n个为一组，可得到分组

$$\begin{array}{cccc} X_1, & X_2, & \ldots, & X_n, \\ X_{n+1}, & X_{n+2}, & \ldots, & X_{2n}, \\ \cdots & & & \\ X_{(t-1)n+1}, & X_{(t-1)n+2}, & \ldots, & X_{tn}. \end{array}$$

- 经过分组的新信源取值空间为$\mathcal{X}^n$，这种对随机变量进行扩展的技术对应着信源的n次**扩展**(extension).
- 相应的信源编码则是对$\mathcal{X}^n$中的所有元素进行编码，而且译码每次译出一组，自然恢复了原有序列.

无限序列编码实例

例. 设硬币Ⓤ具备一定控制朝向的魔力，它基本上可控制朝向为1，但会犯1次错误，并且Ⓤ不知道它在哪次投掷中犯错. 对不断投掷硬币Ⓤ的过程进行3次扩展并给出编码.

解: 硬币Ⓤ的原取值空间为$\mathcal{X} = \{0, 1\}$, 3次扩展信源的取值空间为

$$\mathcal{X}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}.$$

由于硬币Ⓤ只可能出现一次错误，于是扩展信源的取值空间只可能为

$$\mathcal{X}^* = \{011, 101, 110, 111\},$$

其中只有111出现的概率为1，其余均为0，根据Huffman编码可给出一种编码为

$$C(111) = 1, C(110) = 01, C(101) = 000, C(011) = 001.$$

截断问题

- 对于有限长度序列$X_1, X_2, ..., X_L$, 扩展方法类似于无限序列, 但是,

截断问题

- 对于有限长度序列$X_1, X_2, ..., X_L$, 扩展方法类似于无限序列, 但是,
- 扩展次数(分组长度)n有何要求?

截断问题

- 对于有限长度序列$X_1, X_2, ..., X_L$, 扩展方法类似于无限序列, 但是,
- 扩展次数(分组长度)n有何要求?
  - 如果L不是素数, 例如$L = 100$, 我们需要取n能够整除100, 例如4, 10等等.

截断问题

- 对于有限长度序列$X_1, X_2, ..., X_L$, 扩展方法类似于无限序列, 但是,
- 扩展次数(分组长度)n有何要求?
  - 如果L不是素数, 例如$L = 100$, 我们需要取n能够整除100, 例如$4, 10$等等.
  - 如果L是素数, 任意$n \neq L$都无法整除L. 例如0101000, 此时$L = 7$, 若取$n = 2$, 分组为$01, 01, 00$, 但会余下0, 这称为**截断问题**.

截断问题

- 对于有限长度序列$X_1, X_2, ..., X_L$, 扩展方法类似于无限序列, 但是,
- 扩展次数(分组长度)n有何要求?
  - 如果L不是素数, 例如$L = 100$, 我们需要取n能够整除100, 例如4, 10等等.
  - 如果L是素数, 任意$n \neq L$都无法整除L. 例如0101000, 此时$L = 7$, 若取$n = 2$, 分组为01, 01, 00, 但会余下0, 这称为**截断问题**.
- 一种简单的方案是将截断后的序列也纳入取值空间, 对于指定的L和n, 设L除以n的余数为r, 则我们需要考虑的取值空间变为$\mathcal{X}^n \cup \mathcal{X}^r$.

有限序列编码实例(I)

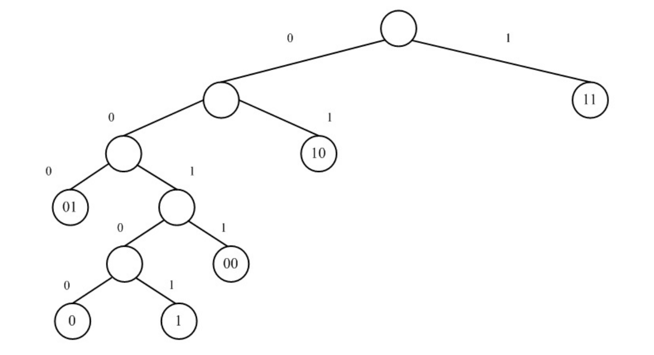
例. 设有权硬币w出现1的概率为0.9, 以2为扩展次数对投掷硬币w的过程进行数据压缩. 设投掷w的过程是i. i. d., 共投掷2000001次, 估算压缩前后描述该序列长度的比率(压缩比).
解: 硬币原取值空间为$\mathcal{X} = \{0, 1\}$, 包含截断的新信源取值空间为$\mathcal{X}^2 \cup \mathcal{X}$. 由于投掷次数相当大, 信源概率分布近似为:

$$p(11) = 0.81, p(10) = 0.09, p(01) = 0.09, p(00) = 0.01, \\ p(1) = 0, p(0) = 0.$$

采用Huffman编码, 可知编码C为:

$$C(11) = 1, C(10) = 01, C(01) = 000, C(00) = 0011, \\ C(1) = 00101, C(0) = 00100.$$

有限序列编码实例(II)



图：硬币ⓦ的Huffman编码

有限序列编码实例(III)

压缩比可估算如下:

- 前2000000次描述该序列的长度约等于
$$\frac{2000000}{2}\Big(p(11)|C(11)|+p(10)|C(10)|+p(01)|C(01)|+p(00)|C(00)|\Big).$$
根据前文可算出其值为1300000.
- 第2000001个符号只能为0或1, 其编码长度是$|C(0)|$或$|C(1)|$, 而这两个长度均为5.
- 于是压缩比为
$$1300005/200001 \approx 0.65.$$

有限序列编码实例(III)

压缩比可估算如下:

- 前2000000次描述该序列的长度约等于

$$\frac{2000000}{2}\Big(p(11)|C(11)|+p(10)|C(10)|+p(01)|C(01)|+p(00)|C(00)|\Big).$$

  根据前文可算出其值为1300000.

- 第2000001个符号只能为0或1, 其编码长度是$|C(0)|$或$|C(1)|$, 而这两个长度均为5.
- 于是压缩比为

$$1300005/200001 \approx 0.65.$$

从上述分析上看, 这种扩展确实减少了描述信息的长度, 获得了压缩效果.

压缩比

- 一般化的压缩比的定义是:
  - 压缩前信源使用字母表$\mathcal{A}$表示, 共需要L个符号描述信源发出的序列.
  - 压缩后信源使用字母表$\mathcal{A}'$编码, 且编码后共需要L'个符号描述该序列.

压缩比则为L'/L.

压缩比

- 一般化的压缩比的定义是:
  - 压缩前信源使用字母表$\mathcal{A}$表示, 共需要L个符号描述信源发出的序列.
  - 压缩后信源使用字母表$\mathcal{A}'$编码, 且编码后共需要L'个符号描述该序列.

  压缩比则为L'/L.
- 为公平起见一般取$|\mathcal{A}| = |\mathcal{A}'|$. 因为不同的字母表描述能力不一样.

压缩比

- 一般化的压缩比的定义是:
  - 压缩前信源使用字母表$\mathcal{A}$表示, 共需要L个符号描述信源发出的序列.
  - 压缩后信源使用字母表$\mathcal{A}'$编码, 且编码后共需要L'个符号描述该序列.

  压缩比则为L'/L.
- 为公平起见一般取$|\mathcal{A}| = |\mathcal{A}'|$. 因为不同的字母表描述能力不一样.
- 回想DNA序列问题.

压缩比

- 一般化的压缩比的定义是:
  - 压缩前信源使用字母表$\mathcal{A}$表示, 共需要L个符号描述信源发出的序列.
  - 压缩后信源使用字母表$\mathcal{A}'$编码, 且编码后共需要L'个符号描述该序列.

  压缩比则为L'/L.
- 为公平起见一般取$|\mathcal{A}| = |\mathcal{A}'|$. 因为不同的字母表描述能力不一样.
- 回想DNA序列问题.
- 显然压缩比越小, 数据压缩效率越高.

压缩比

- 一般化的压缩比的定义是:
  - 压缩前信源使用字母表$\mathcal{A}$表示, 共需要L个符号描述信源发出的序列.
  - 压缩后信源使用字母表$\mathcal{A}'$编码, 且编码后共需要L'个符号描述该序列.

  压缩比则为L'/L.
- 为公平起见一般取$|\mathcal{A}| = |\mathcal{A}'|$. 因为不同的字母表描述能力不一样.
- 回想DNA序列问题.
- 显然压缩比越小, 数据压缩效率越高.
- 但是, 扩展后的新信源编码较为复杂, 特别是截断问题让压缩比的理论分析更为困难.

压缩比

- 一般化的压缩比的定义是:
  - 压缩前信源使用字母表$\mathcal{A}$表示, 共需要L个符号描述信源发出的序列.
  - 压缩后信源使用字母表$\mathcal{A}'$编码, 且编码后共需要L'个符号描述该序列.

  压缩比则为L'/L.
- 为公平起见一般取$|\mathcal{A}| = |\mathcal{A}'|$. 因为不同的字母表描述能力不一样.
- 回想DNA序列问题.
- 显然压缩比越小, 数据压缩效率越高.
- 但是, 扩展后的新信源编码较为复杂, 特别是截断问题让压缩比的理论分析更为困难.
- 如何在理论分析中规避截断问题?

截断问题误差分析

定理. 以字母表$\{0,1\}$进行Huffman编码，在分组长度n很大时，不考虑截断问题所导致的码字期望长度至多有误差$|\mathcal{X}|^{-n}$.

证明: 非$\mathcal{X}^n$中元素概率均近似为0，Huffman编码的前面步骤中会处理这些元素，处理完后与$\mathcal{X}^n$中概率最小的元素s形成一棵树，该树的概率为$p(s)$，而这等价于仅对$\mathcal{X}^n$中元素进行编码，所不同的是s的码长会增加1，即误差为$p(s)$.

由于s概率最小，则

$$\sum_{x\in\mathcal{X}^n} p(s) \leq \sum_{x\in\mathcal{X}^n} p(x) = 1,$$

显然有

$$|\mathcal{X}^n|p(s) = |\mathcal{X}|^n p(s) \leq 1,$$

因此编码的期望长度误差$p(s)$上界可确定，即$|\mathcal{X}|^{-n}$.

截断问题误差分析

定理. 以字母表$\{0,1\}$进行Huffman编码，在分组长度n很大时，不考虑截断问题所导致的码字期望长度至多有误差$|\mathcal{X}|^{-n}$.

证明: 非$\mathcal{X}^n$中元素概率均近似为0，Huffman编码的前面步骤中会处理这些元素，处理完后与$\mathcal{X}^n$中概率最小的元素s形成一棵树，该树的概率为$p(s)$，而这等价于仅对$\mathcal{X}^n$中元素进行编码，所不同的是s的码长会增加1，即误差为$p(s)$.

由于s概率最小，则

$$\sum_{x\in\mathcal{X}^n} p(s) \le \sum_{x\in\mathcal{X}^n} p(x) = 1,$$

显然有

$$|\mathcal{X}^n|p(s) = |\mathcal{X}|^n p(s) \le 1,$$

因此编码的期望长度误差$p(s)$上界可确定，即$|\mathcal{X}|^{-n}$.

由于n趋近于无穷大时误差的极限为0，因此，我们可以忽略截断问题，即假定扩展后取值空间就是$\mathcal{X}^n$.

统计DNA序列n元组的频率代码实现(I)

可将n元组化成数字再用数组下标统计，但有可能空间消耗太大. 这里给出一种$O(L\log L)$时间的就地算法，其中L为序列长度并假设n整除L. 此外，文件中可能有一些空格等特殊符号需要摒除.

核心代码

```
// 指定分组长度，本例中取12.
int n = 12;
// 基于映射的判定策略.
int value[256] = {-1};
value['A'] = 0;
value['C'] = 1;
value['G'] = 2;
value['T'] = 3;
// 读取文件.
fstream infile;
infile.open("DNA.txt", ios::in);
string s = string(n, ' ');  // 用于存储n元组.
```

统计DNA序列n元组的频率代码实现(II)


```
vector<string> V;
size_t k = 0;
while(!infile.eof())
{
    char ch;
    infile >> ch;
    if(value[ch] != -1)    // 判断合法与否.
    {
        s[k] = ch;
        k++;
        if(k == n)    // 读满n个DNA符号则放入V.
        {
            k = 0;
            V.push_back(s);
        }
    }
}
```

统计DNA序列n元组的频率代码实现(III)


```
// 对V进行排序.
sort(V.begin(), V.end());

// 线性扫描统计不同分组出现的频率.
size_t i = 0;
while (i < V.size())
{
    size_t j = i + 1;
    while (j < V.size())
        if (V[j] != V[i]) // 扫描到不等于V[i]的位置.
            break;
        else
            j++;
    // j - i就是V[i]的个数.
    cout << V[i] << " " << j - i << endl;
    i = j;
}
```

若干问题

- 信源扩展后能降低描述序列的长度，但究竟能降低到多少？或者说压缩比的极限是什么？

若干问题

- 信源扩展后能降低描述序列的长度，但究竟能降低到多少？或者说压缩比的极限是什么？
- Huffman编码虽然是最优码，但无法给出定量的理论分析，如何处理？

若干问题

- 信源扩展后能降低描述序列的长度，但究竟能降低到多少？或者说压缩比的极限是什么？
- Huffman编码虽然是最优码，但无法给出定量的理论分析，如何处理？
- 熵是否还能刻画编码性能的极限？

若干问题

- 信源扩展后能降低描述序列的长度, 但究竟能降低到多少? 或者说压缩比的极限是什么?
- Huffman编码虽然是最优码, 但无法给出定量的理论分析, 如何处理?
- 熵是否还能刻画编码性能的极限?
- 下面给出进一步分析.

i. i. d. 序列(1)

- 信源发出的消息组成的随机序列是i. i. d.，概率分布与X相同，可知扩展后的随机序列分布均相同，不妨记为X^n.

i. i. d. 序列(1)

- 信源发出的消息组成的随机序列是i. i. d., 概率分布与X相同, 可知扩展后的随机序列分布均相同, 不妨记为X^n.
- 进行n次扩展后, 字母表为$\mathcal{A}$情况下的对于X^n的的最佳码为C^n, 其期望长度为$l(C^n)$.

i. i. d. 序列(1)

- 信源发出的消息组成的随机序列是i. i. d.，概率分布与X相同，可知扩展后的随机序列分布均相同，不妨记为X^n.
- 进行n次扩展后，字母表为$\mathcal{A}$情况下的对于X^n的的最佳码为C^n，其期望长度为$l(C^n)$.
- 利用Shannon编码作为最佳码的上界估计，可知

$$\frac{H(X^n)}{\log|\mathcal{A}|} \le l(C^n) \le \frac{H(X^n)}{\log|\mathcal{A}|} + 1.$$

i. i. d. 序列(1)

- 信源发出的消息组成的随机序列是i. i. d., 概率分布与X相同, 可知扩展后的随机序列分布均相同, 不妨记为X^n.
- 进行n次扩展后, 字母表为$\mathcal{A}$情况下的对于X^n的的最佳码为C^n, 其期望长度为$l(C^n)$.
- 利用Shannon编码作为最佳码的上界估计, 可知

$$\frac{H(X^n)}{\log|\mathcal{A}|} \le l(C^n) \le \frac{H(X^n)}{\log|\mathcal{A}|} + 1.$$

- 从数据压缩的观点看, 要比较扩展后编码的性能, 必须比较每个随机变量平均所需要的期望描述长度, 于是可定义**每随机变量期望描述长度**(expected description length per random variable):

$$l(C^n/n) = \frac{l(C^n)}{n}.$$

i. i. d.序列(2)

- 由于$H(X^n) = nH(X)$, 因此

$$\frac{1}{n}\frac{nH(X)}{\log|\mathcal{A}|} \le \frac{l(C^n)}{n} < \frac{1}{n}\left(\frac{nH(X)}{\log|\mathcal{A}|} + 1\right),$$

$$\frac{H(X)}{\log|\mathcal{A}|} \le \frac{l(C^n)}{n} < \frac{H(X)}{\log|\mathcal{A}|} + \frac{1}{n}.$$

- 如果令n趋近于正无穷大, 则每随机变量期望描述长度$l(C^n/n)$存在极限, 即熵$H(X)/\log|\mathcal{A}|$.
- 进而可求出采用扩展技术与不采用扩展技术的压缩比界限为

$$\frac{H(X)}{H(X) + \log|\mathcal{A}|} < \lim_{n\to+\infty} \frac{l(C^n/n)}{l(C)} < 1.$$

i. i. d.序列(2)

- 由于$H(X^n) = nH(X)$，因此

$$\frac{1}{n}\frac{nH(X)}{\log|\mathcal{A}|} \leq \frac{l(C^n)}{n} < \frac{1}{n}\left(\frac{nH(X)}{\log|\mathcal{A}|} + 1\right),$$

$$\frac{H(X)}{\log|\mathcal{A}|} \leq \frac{l(C^n)}{n} < \frac{H(X)}{\log|\mathcal{A}|} + \frac{1}{n}.$$

- 如果令n趋近于正无穷大，则每随机变量期望描述长度$l(C^n/n)$存在极限，即熵$H(X)/\log|\mathcal{A}|$.
- 进而可求出采用扩展技术与不采用扩展技术的压缩比界限为

$$\frac{H(X)}{H(X) + \log|\mathcal{A}|} < \lim_{n\to+\infty}\frac{l(C^n/n)}{l(C)} < 1.$$

- 可以看出，$l(C^n/n)$是一个更清晰的指标，我们利用它接着讨论.

离散平稳信源

- 离散平稳分布中任意分组的联合概率分布完全相同，于是可用类似i. i. d.序列的方法进一步考虑离散平稳信源.
- 考虑$X_1, X_2, ..., X_n$，设其最佳码为C^n，其期望长度满足

$$\frac{H(X_1, X_2, ..., X_n)}{\log|\mathcal{A}|} \le l(C^n) < \frac{H(X_1, X_2, ..., X_n)}{\log|\mathcal{A}|} + 1,$$

仍有类似于i. i. d.随机序列的不等式

$$\frac{1}{n}\frac{H(X_1, X_2, ..., X_n)}{\log|\mathcal{A}|} \le \frac{l(C^n)}{n} < \frac{1}{n}\left(\frac{H(X_1, X_2, ..., X_n)}{\log|\mathcal{A}|}+1\right).$$

离散平稳信源

- 离散平稳分布中任意分组的联合概率分布完全相同，于是可用类似i. i. d.序列的方法进一步考虑离散平稳信源.
- 考虑$X_1, X_2, ..., X_n$，设其最佳码为C^n，其期望长度满足

$$\frac{H(X_1, X_2, ..., X_n)}{\log|\mathcal{A}|} \leq l(C^n) < \frac{H(X_1, X_2, ..., X_n)}{\log|\mathcal{A}|} + 1,$$

仍有类似于i. i. d.随机序列的不等式

$$\frac{1}{n}\frac{H(X_1, X_2, ..., X_n)}{\log|\mathcal{A}|} \leq \frac{l(C^n)}{n} < \frac{1}{n}\left(\frac{H(X_1, X_2, ..., X_n)}{\log|\mathcal{A}|}+1\right).$$

- 需要考察

$$\frac{1}{n}\frac{H(X_1, X_2, ..., X_n)}{\log|\mathcal{A}|}$$

是否存在极限?

熵率

- 不妨定义**熵率**(entropy rate):

$$H(\mathcal{X}) = \lim_{n \to +\infty} \frac{H(X_1, X_2, ..., X_n)}{n},$$

前提是该极限确实存在.

熵率

- 不妨定义**熵率**(entropy rate):

$$H(\mathcal{X}) = \lim_{n\to+\infty} \frac{H(X_1, X_2, ..., X_n)}{n},$$

前提是该极限确实存在.

- 使用链式法则分拆可得

$$\frac{H(X_1, X_2, ..., X_n)}{n} = \frac{\sum_{i=1}^{n} H(X_i|X_{i-1}\cdots X_1)}{n},$$

不妨记$A_n = \sum_{i=1}^{n} H(X_i|X_{i-1}\cdots X_1)$, $B_n = n$, 它们满足Stolz定理的条件, 于是

$$\lim_{n\to+\infty} \frac{A_n}{B_n} = \lim_{n\to+\infty} \frac{A_n - A_{n-1}}{B_n - B_{n-1}},$$

于是熵率等价于

$$H(\mathcal{X}) = \lim_{n\to+\infty} H(X_n|X_{n-1}\cdots X_1),$$

前提仍然是该极限确实存在.

离散平稳信源的熵率

定理. 离散平稳信源必然存在熵率.

证明: 对于离散平稳信源, 其条件熵满足

$$H(X_n|X_{n-1}\cdots X_1) \leq H(X_n|X_{n-1}\cdots X_2),$$

由平稳性可知

$$H(X_n|X_{n-1}\cdots X_2) = H(X_{n-1}|X_{n-2}\cdots X_1),$$

则$H(X_n|X_{n-1}\cdots X_1)$单调递减, 且有下限

$$H(X_n|X_{n-1}\cdots X_1) \geq 0,$$

因此其极限存在, 也即熵率存在.

需要指出, 一般而言离散平稳信源的熵率比较难求得, 只有满足某些形式才能简单求解.

变长信源编码定理

定理. 离散平稳信源的每随机变量期望描述长度存在极限.

证明: 我们已经证明离散平稳信源满足:

$$\frac{1}{n}\frac{H(X_1,X_2,...,X_n)}{\log|\mathcal{A}|}\leq\frac{l(C^n)}{n}<\frac{1}{n}\left(\frac{H(X_1,X_2,...,X_n)}{\log|\mathcal{A}|}+1\right).$$

由于离散平稳信源必有熵率, 我们令n趋近于正无穷大, 于是每随机变量期望描述长度的极限满足

$$\lim_{n\to+\infty}l(C^n/n)=\frac{H(\mathcal{X})}{\log|\mathcal{A}|},$$

这称为**变长信源编码定理**(variable-length source coding theorem). 特别地, 信源为i. i. d.序列时, 熵率退化为熵.

变长信源编码定理

定理. 离散平稳信源的每随机变量期望描述长度存在极限.

证明: 我们已经证明离散平稳信源满足:

$$\frac{1}{n}\frac{H(X_1,X_2,...,X_n)}{\log|\mathcal{A}|} \leq \frac{l(C^n)}{n} < \frac{1}{n}\left(\frac{H(X_1,X_2,...,X_n)}{\log|\mathcal{A}|}+1\right).$$

由于离散平稳信源必有熵率, 我们令n趋近于正无穷大, 于是每随机变量期望描述长度的极限满足

$$\lim_{n\to+\infty} l(C^n/n) = \frac{H(\mathcal{X})}{\log|\mathcal{A}|},$$

这称为**变长信源编码定理**(variable-length source coding theorem). 特别地, 信源为i. i. d.序列时, 熵率退化为熵.

由此可见, 熵和熵率是描述信源信息量的定量指标.

计算每随机变量期望描述长度

例. 设某信源$X_1, X_2, ..., X_n, ...$这个随机序列是平稳Markov链，该随机序列的取值空间均为$\mathcal{X}$. 该平稳Markov链的单步概率转移矩阵为$\mathbf{P}$，平稳分布为$\boldsymbol{\pi}$. 分析该信源每随机变量期望描述长度的极限问题.

解：易知此信源是离散平稳信源，则熵率存在，且有

$$H(\mathcal{X}) = \lim_{n\to+\infty} H(X_n|X_{n-1}\cdots X_1),$$

由平稳Markov链可知

$$\begin{aligned} H(X_n|X_{n-1}\cdots X_1) &= H(X_n|X_{n-1}) \\ &= -\sum_{u\in\mathcal{X}}\sum_{v\in\mathcal{X}}\Big(\pi(u)\mathbf{P}_{uv}\log\mathbf{P}_{uv}\Big), \end{aligned}$$

于是每随机变量期望描述长度的极限为

$$l(C^n/n) = \frac{H(\mathcal{X})}{\log|\mathcal{A}|} = -\frac{1}{\log|\mathcal{A}|}\sum_{u\in\mathcal{X}}\sum_{v\in\mathcal{X}}\Big(\pi(u)\mathbf{P}_{uv}\log\mathbf{P}_{uv}\Big).$$

Markov信源的熵率

定理. 设Markov信源发出消息为$W_1, W_2, \dots W_n, \dots$, 若它所蕴含的Markov链可达到平稳分布, 那么则该信源的熵率存在, 其值为

$$H(\mathcal{W}) = -\sum_{s\in\mathcal{S}}\sum_{w\in\mathcal{W}}\Big(\pi(s)p(w|s)\log p(w|s)\Big).$$

定理. 设C_s^n是针对Markov信源状态为s情况下分组长度为n的编码, 我们会针对不同的状态使用不同的编码, 而每随机变量期望描述长度的极限为

$$\lim_{n\to+\infty}\left(\sum_{s\in\mathcal{S}} p(s) l(C_s^n)\right) = \frac{H(\mathcal{W})}{\log|\mathcal{A}|}.$$

附注: 上述两个定理的证明篇幅较长, 可参见教材.

计算Markov信源熵率(I)

例. 设某Markov的消息集合为$\mathcal{W} = \{0, 1\}$, 它发出的消息仅依赖于前两个消息, 其转换情况如图所示, 分析该信源的熵率.

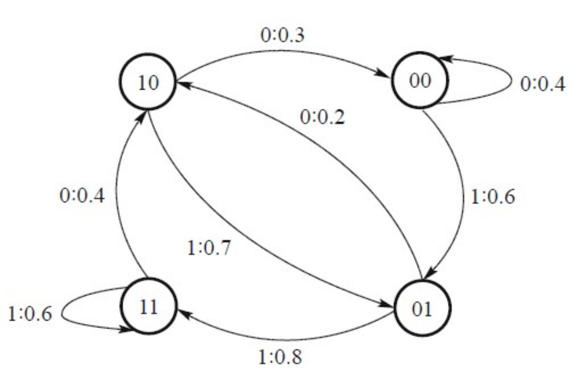


图: Markov信源示例

计算Markov信源熵率(II)

解：单步概率转移矩阵为

$$\mathbf{P}=\begin{bmatrix}0.4 & 0.6 & 0 & 0\\ 0 & 0 & 0.2 & 0.8\\ 0.3 & 0.7 & 0 & 0\\ 0 & 0 & 0.4 & 0.6\end{bmatrix}.$$

易知该Markov链是不可约的.
由于$\mathbf{P}_{00}^{(2)}>0$，可用归纳法证明$n>1$情况下$\mathbf{P}_{00}^{(n)}>0$，所以状态0是非周期性的，进而所有状态都是非周期性的.
可计算出平稳分布为

$$\boldsymbol{\pi}=\left(\frac{1}{9},\frac{2}{9},\frac{2}{9},\frac{4}{9}\right),$$

该信源熵率存在，其值为

$$H(\mathcal{W})=-\sum_{s\in\mathcal{S}}\sum_{w\in\mathcal{W}}\Big(\pi(s)p(w|s)\log p(w|s)\Big)=0.9\text{比特}.$$

进一步改进编码性能

- Huffman编码C_H^n的期望长度界限可表述为

$$l(C_H^n) = nH(X) + O(1)$$

进一步改进编码性能

- Huffman编码C_H^n的期望长度界限可表述为

$$l(C_H^n) = nH(X) + O(1)$$

- 2011年，Wojciech Szpankowski和Sergio Verdú两位学者在IEEE Transactions on Information Theory上发表了一篇论文：*Minimum Expected Length of Fixed-to-Variable Lossless Compression Without Prefix Constraints*. 他们取消了传统的前缀约束，以二元序列

$$\varnothing, 0, 1, 00, 01, 10, 11, \ldots$$

为编码C_{SV}^n，其期望长度改进为

$$l(C_{SV}^n) = nH(X) - \frac{1}{2}\log n + O(1).$$

进一步改进编码性能

- Huffman编码C_H^n的期望长度界限可表述为

$$l(C_H^n) = nH(X) + O(1)$$

- 2011年，Wojciech Szpankowski和Sergio Verdú两位学者在IEEE Transactions on Information Theory上发表了一篇论文：*Minimum Expected Length of Fixed-to-Variable Lossless Compression Without Prefix Constraints*. 他们取消了传统的前缀约束，以二元序列

$$\varnothing, 0, 1, 00, 01, 10, 11, \ldots$$

为编码C_{SV}^n，其期望长度改进为

$$l(C_{SV}^n) = nH(X) - \frac{1}{2}\log n + O(1).$$

The End