Homework 1：From Entropy to KL Divergence with MATLAB Code and Data Example

KL Divergence中文叫做相对熵，也叫做Kullback-Leibler散度

1 What Is Entropy：熵的定义

首先我们先要知道什么是熵，熵是从物理学借鉴过来的定义，用来衡量某个随机变量的不确定性，直觉上，一个128面的骰子向上一面数字显然比投一枚硬币向上一面的图像更加难以确定，所以需要我们需要构建一个变量能够表征这种直观感受到的不确定性的差异。

2 不确定性与熵：从事件到随机变量

2.1 事件的不确定性

我们先考虑一个随机变量中的某个事件发生的不确定性，然后进一步延申到随机变量的不确定性。事件和随机变量的关系类似于，如果把骰子向上那一面的数字当做随机变量的一种表现形式，而“向上为1”则被当做从属于随机变量的一个事件。
现在假想一个6 面骰子投到“1”的这一事件，如果出现“1”的概率越高，那么反过来想，我们就更敢于确定，骰子向上的一面为“1”，换句话说，如果“骰子”出现“1”的概率越大，那么不确定性就越小，按照这样的直觉，我们定义关于事件P的函数f来表征不确定性：

$$f(P) = \mathsf{log}\frac{1}{p} = -\mathsf{log}p \tag{1}$$

这里的P指的是事件P的发生p表示事件P发生的概率，显然，当P越大f就越小。定义这个公式为log 形式的逻辑基础在于两个要求：

- f 与p成反比
- f的可加性，即$f(P_1, P_2) = f(P_1) + f(P_2)$,也就是说两个事件的不确定性之和，等于把两个事件打包以后当做一个事件的不确定性

进一步，我们从单一事件的不确定性出发定义随机变量的熵，简单的直觉就是将各种事件的熵取加权平均，所以我们有：

$$H(U) = E[-\mathsf{log} p_i] = -\sum_{i=1}^{n} p_i \mathsf{log} p_i \tag{2}$$

这里的$H(U)$就表示随机变量U的不确定性，U这个随机变量可能的取值集合是$P_1, P_2 ... P_n$, 需要注意的是，这是针对随机变量的取值为离散情况的公式，如果随机变量是连续的我们有：

$$H(U) = -\int p(x) \mathsf{log}(p(x)) dx \tag{3}$$

3 相对熵

3.1 相对熵的构建

相对熵直观上就是用来表示两个分布间到底距离有多远差别有多大，更具体一点来说，是按照固定权重，求两个随机变量各个事件的不确定性的加权平均数，“相对熵”的相对就在于比较两个熵之间的差别。因此首先我们先给出相对熵的公式如下：

$$KL(P||Q) = -\sum_{x \in X} p(x) \mathsf{log} \frac{1}{p(x)} + \sum_{x \in X} p(x) \mathsf{log} \frac{1}{p(x)} = \sum_{x \in X} p \mathsf{log} (\frac{p(x)}{q(x)}) \tag{4}$$

注意到$\sum_{x \in X} p(x) \mathsf{log} \frac{1}{p(x)}$与$\sum_{x \in X} p(x) \mathsf{log} \frac{1}{p(x)}$ 都是以$p(x)$当作“权重”，而非一个$p(x)$一个$q(x)$原因在于：人们为了估计两个随机变量的距离就需要在同一个标尺上比较。

3.2 相对熵的重要性质：不小于0

3.2.1 吉布斯不等式的含义

这个部分将给出该性质的证明，首先介绍吉布斯不等式：

若$\sum_{i=1}^{n} q_i = \sum_{i=1}^{n} p_i = 1$，且$p_i, q_i \in (0, 1]$,则有：

$$-\sum_{i=1}^{n} p_i \mathsf{log} p_i \leq -\sum_{i=1}^{n} p_i \mathsf{log} q_i \tag{5}$$

当且仅当$p_i = q_i$时等号成立。

3.2.2 吉布斯不等式的证明

已知$\ln(x) \le x-1$，等号成立当且仅当$x=1$

$$\sum_{i=1}^{n} p_i \log(\frac{q_i}{p_i}) \le \sum_{i=1}^{n} p_i \frac{q_i}{p_i - 1} = \sum_{i=1}^{n} q_i - \sum_{i=1}^{n} p_i = 1 - 1 = 0 \tag{6}$$

综上，运用吉布斯不等式，我们有：

$$KL(P||Q) \ge 0 \tag{7}$$

4 相对熵的$MATLAB$计算及实践

这一部分我们将学着运用$MATLAB$来实践怎么算离散概率分布随机变量间的$KL-divergence$，我们参考到网上的一段代码，做了一定的修改,并附上了中文说明方便后来者迅速掌握.

%用来处理简单的$KL-divergence$

%这个函数的输入段需要输入4个变量，第一个是$varValue$也就是随机变量在不同情况下可能会出现的返回值，比如说[0 1 2 3 4]’,就指的是随机变量的观测值包括0到4，$pVect1$显示了X 向量中每个值出现的概率，比如[0.1 0.2 0.4 0.2 0.1]’,那么就说明出现数字2的概率，在$p1$的分布下是0.4，同理有p2的构建，既pVact2.

%第四个变量用来调节相对熵的性质，如果第四个变量不赋值，那么意味着计算出的为传统意义上的相对熵，但是存在另一个也很常用的相对熵的定义，也就是对称的相对熵，具体来说公式如下：

$$KL(P1(x), P2(x)) = sum[P1(x).log(P1(x)/P2(x))]$$

$[KL(P1,P2)+KL(P2,P1)]/2.$，也就是$P1$相对$P2$的相对熵加上$P2$相对$P1$的相对熵加和以后求均值，如果需要转换计算出这种相对熵，则需要将第四项赋值为”sym” 其他的一些细节，当X中出现重复数字时，我们将同个数字当作分别不同的数字来处理，代码将给出提示，同时对于$p1$，$p2$这两个向量，它们的元素之后需要为1，源代码也将给出这一检验过程

对于概率值为0的情况直接取log代码会报错所以我们加了一项eps,eps即$matlab$中的最小单位

```
function KL = KLDiv(varValue,pVect1,pVect2,varargin)
if ~isequal(unique(varValue),sort(varValue)),
    warning('KLDIV:duplicates','X contains duplicate values. Treated as distinct values.')
end
if ~isequal(size(varValue),size(pVect1)) || ~isequal(size(varValue),size(pVect2)),
    error('All inputs must have same dimension.')
end
% Check probabilities sum to 1:
if (abs(sum(pVect1) - 1) > .00001) || (abs(sum(pVect2) - 1) > .00001),
    error('Probablities don''t sum to 1.')
end
if ~isempty(varargin),
    switch varargin{1},
        case 'js',
            logQvect = log2((pVect2+pVect1)/2);
            KL = .5 * (sum(pVect1.*(log2(pVect1)-logQvect)) + ...
                sum(pVect2.*(log2(pVect2)-logQvect)));
        case 'sym',
            KL1 = sum(pVect1 .* (log2(pVect1)-log2(pVect2)));
            KL2 = sum(pVect2 .* (log2(pVect2)-log2(pVect1)));
            KL = (KL1+KL2)/2;

        otherwise
            error(['Last argument' ' "' varargin{1} '" ' 'not
```

```
                        recognized.'])
    end
else
    KL = sum(pVect1 .* (log2(pVect1)-log2(pVect2)));
end
```

主程序为：

```
X = [1 2 3 4 5]';
 P1 = [0.4 0.2 0.3 0.05 0.05]'+eps;
 P2 = [0 0 .5 .2 .3]' + eps;
KL=KLDiv(X,P1,P2)
```

实验结果为22.7565