Multiview deconvolution

1 Richardson–Lucy deconvolution

1.1 基于Possion分布的推导[1]

1.1.1 物理模型

成像物理模型为

$$\mathbf{g} = \text{Possion}\left(\mathbf{f} \otimes \mathbf{h}\right)$$

采集到的图像为$\mathbf{g} = [g_1, g_2, \ldots, g_N]^T$，原图像为$\mathbf{f} = [f_1, f_2, \ldots, f_N]^T$，$\mathbf{h}$为卷积核，其中，N为图像维度。

> 补充：泊松分布是为了考虑光子噪声。光子噪声是由于到达传感器的光子数目发生变化，导致实际情况与理论情况发生偏差而产生的噪声。能否接收到光子可以用二项分布来进行描述，该像素点对于整个平面来说，接收到的概率较小，但是整个平面接收到的光子数目较多（相当于进行很多次实验），因此近似为泊松分布，因此光子噪声又称为泊松噪声[2]。其数学表达式如下：
>
> 在给定的时间间隔T下，到达某一个像素的光子数量n是一个随机变量，其服从泊松分布，其概率密度可以表示为
>
> $$p(n) = \frac{\mathrm{e}^{-\lambda}\lambda^n}{n!}, \quad n = 0, 1, 2, \ldots$$
>
> 其中，λ正比于T，是这个像素上收集到光子数的期待值，理论值。

1.1.2 推导过程

令$\mathbf{p} = \mathbf{f} \otimes \mathbf{h}$，表示卷积后的图像向量。根据泊松分布的模型可以得到

$$P(\mathbf{g} \mid \mathbf{p}) = \prod_i^N \text{Poisson}\left(p_i\right) = \prod_i^N \frac{p_i^{g_i} e^{-p_i}}{g_i!}$$

为进行极大似然估计，我们对上述表达式取对数得到

$$\ln\left(P(\mathbf{g} \mid \mathbf{p})\right) = \sum_{i}^{N}\left[(g_i \ln p_i - p_i) - \ln\left(g_i!\right)\right]$$

$\ln\left(g_i!\right)$是常数，故可省去，令似然函数为：

$$L(\mathbf{g} \mid \mathbf{p}) = \sum_{i}^{N}\left(g_i \ln p_i - p_i\right)$$

设卷积矩阵H，满足

$$\mathbf{p} = \mathbf{f} \otimes \mathbf{h} = Hf$$

对上式展开后有

$$p_m = \sum_{j}^{N} H_{mn} f_j$$

为了使得$L(\mathbf{g} \mid \mathbf{p})$最大，可以$\mathbf{f}$按照梯度方向(梯度上升)进行迭代，迭代公式为

$$\hat{\mathbf{f}}_{new} = \hat{\mathbf{f}}_{old} + \lambda \frac{\partial L(\mathbf{g} \mid \mathbf{p}(\mathbf{f}))}{\partial \mathbf{f}}\bigg|_{\hat{\mathbf{f}}_{old}}$$

下面对上述导数进行求解

$$\frac{\partial L(\mathbf{g} \mid \mathbf{p}(\mathbf{f}))}{\partial f_j} = \sum_{i=1}^{N} \frac{\partial p_i}{\partial f_j}(\frac{g_i}{p_i} - 1)$$

显然，对于

$$\frac{\partial p_i}{\partial f_j} = H_{i,j}$$

于是

$$\frac{\partial L(\mathbf{g} \mid \mathbf{p}(\mathbf{f}))}{\partial f_j} = \sum_{i=1}^{N} H_{i,j}(\frac{g_i}{p_i} - 1)$$

> 这里补充思路：因为最终要求导$\frac{\partial L(\mathbf{g}|\mathbf{p}(\mathbf{f}))}{\partial \mathbf{f}}$，需要写成矩阵和向量的形式，这里是H的第j列乘后面那个向量的第j的元素，然后相加，不满足的矩阵相乘的规律，所以比较直观的想法就是把H写成转置的形式

写成转置形式后，有

$$\frac{\partial L(\mathbf{g} \mid \mathbf{p}(\mathbf{f}))}{\partial f_j} = \sum_{i=1}^{N} H_{j,i}^{T}(\frac{g_i}{p_i} - 1)$$

可以得到矩阵形式，这里的除法指的是点点相除(element wise)

$$\frac{\partial L(\mathbf{g} \mid \mathbf{p}(\mathbf{f}))}{\partial f_j} = H^T \left(\frac{\mathbf{g}}{\mathbf{p}} - \mathbf{1}\right)$$

Let's now propose the following arbitrary and key step, （这里为什么这么假设，可能是EM算法推导出来的步长，这里需要再考虑,也有可能是为了消掉那一项），这里的除法指的是点点相除(element wise)

$$\lambda = \frac{\hat{\mathbf{f}}_{old}}{H^T \mathbf{1}}$$

于是可以推得

$$\begin{aligned} \hat{\mathbf{f}_{new}} &= \hat{\mathbf{f}}_{old} + \lambda \frac{\partial L(\mathbf{g} \mid \mathbf{p}(\mathbf{f}))}{\partial \mathbf{f}} \bigg|_{\hat{\mathbf{f}}_{old}} \\ &= \hat{\mathbf{f}}_{old} + \frac{\hat{\mathbf{f}}_{old}}{H^T \mathbf{1}} \odot H^T \left(\frac{\mathbf{g}}{\mathbf{p}} - \mathbf{1}\right) \\ &= \frac{\hat{\mathbf{f}}_{old}}{H^T \mathbf{1}} \odot H^T \frac{\mathbf{g}}{\mathbf{p}} \end{aligned}$$

将向量$\mathbf{p}$代入可以得到

$$\hat{\mathbf{f}_{new}} = \frac{\hat{\mathbf{f}}_{old}}{H^T \mathbf{1}} \odot H^T \frac{\mathbf{g}}{H \hat{\mathbf{f}}_{old}}$$

1.1.3　结果分析与整理

式中H为卷积矩阵，其具备循环矩阵的特性[3]，有

$$H^T\mathbf{1}=\mathbf{c}$$

上式的$\mathbf{c}$为常数向量，$c_i=C=\sum h(n)$，即为卷积核的能量之和。

整理后，换成卷积形式，可以得到

$$\mathbf{f}_{new}=\frac{\mathbf{f}_{old}}{C}\left(\mathbf{h}\star\frac{\mathbf{g}}{\mathbf{h}\otimes\mathbf{f}}\right)$$

此外，使用RL会得到一个较为稀疏的解，其原因是，对于似然函数公式中的某一项

$$\sum_{i}^{N}p_i=<\vec{1},\mathbf{p}>$$

并且考虑到在迭代过程中$\mathbf{g}$是非负的，所以相当于是一个$\|p\|_1$。

1.2　代码实现过程

1.2.1　伪代码

step1: A1 = ifft(fft2(h).*fft2(f))

step2: A2 = g.*A1

step3: A3 = conj(fft2(f)).*A3

step4: A4 = f./C.*A3

1.2.2　fft做相关补充证明

$$
\begin{aligned}
F[f(-t)] &= \int_{-\infty}^{+\infty} f(-t)e^{-jwt}dt \quad (\text{令 } u=-t) \\
&= \int_{+\infty}^{-\infty} f(u)e^{-jw(-u)}d(-u) \\
&= \int_{-\infty}^{+\infty} f(u)e^{-j(-w)u}du \\
&= F(-\omega)
\end{aligned}
$$

当$f(t)$为实数时，$F(-w)=F^*(w)$

1.3 Ref

[1] Richardson–Lucy deconvolution, Wikipedia

[2] https://blog.csdn.net/qq_39332723/article/details/107918367

[3] 卷积的循环矩阵求解方法*xiamentingtao的博客-CSDN博客循环卷积矩阵*

2 Multiview Richardon Lucy Deconvolution

对于多视角解卷积，**当我们假设不同视角拍摄到的图像是独立的**，对于我们通过物分布$\mathbf{f}$获得图像分布$\mathbf{g}=[\mathbf{g_1^T},\mathbf{g_2^T},\ldots,\mathbf{g_M^T}]$(其中M为视角个数)的概率为

$$
P(\mathbf{g}|\mathbf{f})=\prod_{k=1}^{M}P(\mathbf{g_k}|\mathbf{f})=\prod_{k}^{M}\prod_{i}^{N}\text{Poisson}\left(p_i^{(k)}\right)=\prod_{k}^{M}\prod_{i}^{N}\frac{p_i^{(k)g_i^{(k)}}e^{-p_i^{(k)}}}{g_i^{(k)}!}
$$

w其推导过程几乎与单视角Richardon Lucy解卷积相同，只是在外部嵌套了叠加项，此处不再赘述，最终推导出的迭代公式为

$$
\mathbf{f}_{new}=\sum_{k}\mathbf{f}_{old}\frac{1}{C_k}\left(\mathbf{h}\star\frac{\mathbf{g}}{\mathbf{h}\otimes\mathbf{f}}\right)
$$