

PP²: Poisson Process Photon Prediction

窦威、刘学伟、张爱强、王宇逸、武益阳、续本达

2024 年 11 月 20 日

第一章 探测器响应函数

张爱强、续本达

现在我们有一个窦威的 PMT 对点源响应模型，这个模型可以给出点源在 PMT 上的电荷沉积。但是为了能够替代 Geant4 的光学和电子学模拟，电荷沉积是不够的，还需要在得到点源在 PMT 上的波形响应。在现有窦威模型的基础上，加上时间维度。

所研究的探测器为球形，介质均匀各向同性。对于某一点 \vec{r} 沉积能量为 E 的事例 $\delta(\vec{r}, E)$ ，将会在球面上的 PMT_j 产生 Poisson 过程均值函数 $R_j(t; \vec{r}, E)$ ，代表 t 对应的光电子期望数密度。

记球心为 O ，如果对于每一个位于 r_j 的 PMT_j ，探测器除去与 PMT 面积和 QE 等相关的其它性质关于 Or_j 轴对称。以 Or_j 为轴，定义球极坐标 (r, θ_j, ϕ_j) ，则不同位置 Poisson PE 数期望 $\lambda_j(\vec{r}, E) = \int_0^\infty R_j(t; \vec{r}, E) dt$ 与 ϕ_j 无关，只与 r, θ_j 相关。Poisson 过程均值函数 $R_j(t; \vec{r}, E)$ 也只与 r, θ_j 相关，写作 $R_j(t; r, \theta_j, E)$ 。

如果在采样时，位置按照概率密度 $\rho(r, \theta_j)$ 随机选取，拓展的 Poisson 过程均值函数为

$$R_j^A(t, r, \theta_j; E) := \rho(r, \theta_j) \times R_j(t; r, \theta_j, E). \quad (1.1)$$

一旦采样的概率密度是均匀的，即 $\rho(r, \theta_j)$ 为常数，那么 R_j^A 退化为 R_j 。

$R_j^A(t, r, \theta_j; E)$ 可看作一个标量场。其定义域

$$\begin{cases} t \in [0, +\infty) \\ r \in [0, R] \\ \theta_j \in [0, \pi] \end{cases} \quad (1.2)$$

对应于一个实心半圆柱的标量场，满足 $R_j^A \geq 0$ 。

第二章 指数函数基近似

窦威、张爱强、王宇逸、续本达

类似于 Spherical Harmonics 方法，少量的函数基可以刻画对应的场。在实心圆柱上的函数基可以拆分成圆面上和 t 轴（即圆柱 z 轴）两组无关的基。圆面上的基目前计划选用 Zernike 多项式 $Z_l(r, \theta_j)$ ， t 轴上的函数基要求叠加后为正，暂且使用 Legendre 多项式 $P_l(t)$ 。未来可能发现更好的选择。

依据上面的思想，标量场 R_j 展开为

$$\log R_j(t; r, \theta_j, E) = \sum_{m,n} a_j^{mn}(E) P_m(t) Z_n(r, \theta_j) \quad (2.1)$$

在球对称的线性系统中，除了 $a_j^{00}(E)$ 之外的 $a_j^{mn}(E)$ 都与 j 和 E 无关，记为 a^{mn} 。选取某个特殊的 $E = \tilde{E}$ ，使得 $\sum_j a_j^{00}(\tilde{E}) = 0$ ，将 $a_j^{00}(\tilde{E})$ 简记为 a_j^0 ，同时将 $R_j(t; r, \theta_j, \tilde{E})$ 简记为 $R_j(t, r, \theta_j)$ 。那么有，

$$\log R_j(t, r, \theta_j) = a_j^0 + \sum_{\substack{m,n \\ m+n \neq 0}} a^{mn} P_m(t) Z_n(r, \theta_j). \quad (2.2)$$

R_j 需要存储的部分为 $N - 1$ (N 为 PMT 的个数) 个 a_j^0 和与阶数相关的 a^{mn} 。

2.1 系数拟合

模拟一组点源 (r^k, θ_j^k) ，其产生的 PE 为 (t_i, r_i, θ_{j_i}) ，其中的 i, j, k 分别为 PE、PMT 和事例的编号。写出非齐次 Poisson 过程的似然函数，

$$\begin{aligned} & \log \mathcal{L}(\{a_j^0\} \cup \{a^{mn}\}) \\ &= \log \prod_i [R_{j_i}(t_i; r_i, \theta_{j_i})] - \sum_{k,j} \int R_j(t; r^k, \theta_j^k) dt \\ &= \sum_i \left[a_{j_i}^0 + \sum_{\substack{m,n \\ m+n \neq 0}} a^{mn} P_m(t_i) Z_n(r_i, \theta_{j_i}) \right] \\ &\quad - \sum_{k,j} e^{a_j^0} \int \exp \left[\sum_{\substack{m,n \\ m+n \neq 0}} a^{mn} P_m(t) Z_n(r^k, \theta_j^k) \right] dt \\ &= \sum_i a_{j_i}^0 + \sum_{\substack{m,n \\ m+n \neq 0}} a^{mn} \sum_i P_m(t_i) Z_n(r_i, \theta_{j_i}) \\ &\quad - \sum_{k,j} e^{a_j^0} \int \exp \left[\sum_{\substack{m,n \\ m+n \neq 0}} a^{mn} P_m(t) Z_n(r^k, \theta_j^k) \right] dt, \end{aligned} \quad (2.3)$$

对 (2.3) 式的概率求极大值, 得到系数,

$$\hat{a}_j^0, \hat{a}^{mn} = \arg \max_{\substack{a_j^0, a^{mn} \\ \sum_j a_j^0 = 0}} \log \mathcal{L}(\{a_j^0\} \cup \{a^{mn}\}), \quad (2.4)$$

2.1.1 极大似然估计

(2.3) 式的带有 \hat{a}^{mn} 的积分较为困难, 在实际中可以使用 `logsumexp` 函数来离散求和, 以尽可能保证其数值稳定性。

\hat{a}_j^0 可解析求解, 对 (2.3) 式求 a_j^0 的偏导,

$$\frac{\partial \log \mathcal{L}}{\partial a_j^0} = N_j - e^{a_j^0} \sum_k \int \exp \left[\sum_{\substack{m,n \\ m+n \neq 0}} a^{mn} P_m(t) Z_n(r^k, \theta_j^k) \right] dt \quad (2.5)$$

其中 N_j 为模拟数据中第 j 个 PMT 上的总 hit 数, 即 $N_j = \sum_{i|j_i=j} 1$ 。进而可得

$$\hat{a}_j^0 = \log N_j - \log \left\{ \sum_k \int \exp \left[\sum_{\substack{m,n \\ m+n \neq 0}} \hat{a}^{mn} P_m(t) Z_n(r^k, \theta_j^k) \right] dt \right\}. \quad (2.6)$$

可以把 (2.6) 代入 (2.3) 获得一个仅有 a^{mn} 的式子。但它较为繁琐, 不如直接求解 (2.4)。

2.1.2 Poisson 回归

Poisson 回归给出不同于 2.1 节的另一种估计参数的方法。取足够大的时间区间, 等距分解为 t_w , 那么在 t_w 为中心 Δt 窗口中的 hit 数 $N_j(t_w; r, \theta_j)$, 服从 Poisson 分布,

$$N_j(t_w; r, \theta_j) \sim \pi[R_j(t_w; r, \theta_j) \Delta t]. \quad (2.7)$$

Poisson 回归的连接函数写作,

$$\log E[N_j(t_w; r, \theta_j)] = a_j^0 + \sum_{\substack{m,n \\ m+n \neq 0}} a^{mn} P_m(t_w) Z_n(r, \theta_j) + \log \Delta t, \quad (2.8)$$

是关于 $\{a_j^0\} \cup \{a^{mn}\}$ 的线性组合。通过 Poisson 回归算法可求得 $\{\hat{a}_j^0\} \cup \{\hat{a}^{mn}\}$ 。

当 $R_j(t_w; r, \theta_j) \Delta t \ll 1$, Poisson 分布退化成 Bernoulli 分布, 可以使用 logistic 回归替代 Poisson 回归估计 $\{a_j^0\} \cup \{a^{mn}\}$ 。

2.2 Poisson 过程均值函数

得到 $\{\hat{a}_j^0\} \cup \{\hat{a}^{mn}\}$ 后, 即可写出标量场

$$R_j(t; r, \theta_j) = \exp \left(\hat{a}_j^0 + \sum_{\substack{m,n \\ m+n \neq 0}} \hat{a}^{mn} P_m(t) Z_n(r, \theta_j) \right). \quad (2.9)$$

在线性系统中, $R_j(t; r, \theta_j, E)$ 可分离变量,

$$R_j(t; r, \theta_j, E) = R_j(t; r, \theta_j) \times Y(E) \quad (2.10)$$

$Y(E)$ 即为探测器的能标。

平方 probe 同理。

第三章 平方函数基近似

武益阳、窦威、刘学伟、续本达

在实数域上分解 $\sqrt{R_j(t; r, \theta_j, E)}$ 也可以保证重建得到 Probe 恒正。平方表示可利用基底的正交归一性质，定义

$$\sqrt{R_j(t; r, \theta_j)} = \sqrt{E_j} \cdot \sum_{m,n} a^{mn} P_m(t) Z_n(r, \theta_j). \quad (3.1)$$

E_j 项与式 (2.2) 中的 a_j^0 类似，由粒子可见能量与 PMT 量子效率决定。

平方分解可由 Parseval 定理高效计算积分，依 Einstein 约定省略求和号，

$$\begin{aligned} & \int_{\underline{\tau}}^{\bar{\tau}} R_j(2\frac{\tau - \underline{\tau}}{\bar{\tau} - \underline{\tau}} - 1; r, \theta_j) d\tau \\ &= \frac{\bar{\tau} - \underline{\tau}}{2} \int_{-1}^1 (a^{mn} P_m(t) Z_n(r, \theta_j))^2 dt \\ &= \frac{\bar{\tau} - \underline{\tau}}{2} \int_{-1}^1 a^{mn} P_m(t) Z_n(r, \theta_j) a^{pq} P_p(t) Z_q(r, \theta_j) dt \\ &= \frac{\bar{\tau} - \underline{\tau}}{2} a^{mn} Z_n(r, \theta) \delta_{mp} a^{pq} Z_q(r, \theta) \\ &= \frac{\bar{\tau} - \underline{\tau}}{2} a_p^n a^{pq} Z_n(r, \theta) Z_q(r, \theta) \end{aligned} \quad (3.2)$$

其中实验测量时间 τ 在 $[\underline{\tau}, \bar{\tau}]$ 区间中， $t = 2\frac{\tau - \underline{\tau}}{\bar{\tau} - \underline{\tau}} - 1$ ， $dt = \frac{2}{\bar{\tau} - \underline{\tau}} d\tau$ 。在 (3.2) 中 R_j 的量纲为 T^{-1} ， a^{mn} 的量纲为 $T^{-\frac{1}{2}}$ 。

3.1 平方 probe likelihood 的凸性

三节中的平方 probe 对应的似然函数为

$$\log \mathcal{L} = \sum_{i=1}^{N_{PE}} \log [R_{j_i}(t_i; r_i, \theta_{j_i})] - \frac{\bar{\tau} - \underline{\tau}}{2} \sum_{k,j} \int R_j(t; r^k, \theta_j^k) dt \quad (3.3)$$

为了方便推导，将 Legendre 与 Zernike 基底直积为一套基底，记做 $B_n(r, t, \theta_j)$ 。把 a^{mn} 记成一维的 a^n ，把 k, j 指标合并为 k ， θ_{j_i} 记为 θ_i ：

$$\log \mathcal{L} = \sum_{i=1}^{N_{PE}} \log [a^n B_n(t_i, r_i, \theta_i)]^2 - \frac{\bar{\tau} - \underline{\tau}}{2} \sum_{k,j} \int [a^n B_n(t, r^k, \theta_j^k)]^2 dt \quad (3.4)$$

其二阶导为

$$\begin{aligned}
\frac{\partial^2 \log \mathcal{L}}{\partial a_\mu \partial a_\nu} &= 2 \sum_i \frac{\partial^2}{\partial a_\mu \partial a_\nu} \log [a^n B_n(t_i, r_i, \theta_i)] - \frac{\bar{\tau} - \underline{\tau}}{2} \sum_{k,j} \int \frac{\partial^2}{\partial a_\mu \partial a_\nu} [a^n B_n(t, r^k, \theta^k)]^2 dt \\
&= 2 \sum_i \frac{a^n B_n(t_i, r_i, \theta_i) \frac{\partial^2}{\partial a_\mu \partial a_\nu} [a^n B_n(t_i, r_i, \theta_i)] - \frac{\partial}{\partial a_\mu} [a^n B_n(t_i, r_i, \theta_i)] \frac{\partial}{\partial a_\nu} [a^n B_n(t_i, r_i, \theta_i)]}{[a^n B_n(t_i, r_i, \theta_i)]^2} \\
&\quad - \frac{\bar{\tau} - \underline{\tau}}{2} \sum_k \int 2 \left\{ a^n B_n(t, r^k, \theta^k) \frac{\partial^2}{\partial a_\mu \partial a_\nu} [a^n B_n(t, r^k, \theta^k)] + \frac{\partial}{\partial a_\mu} [a^n B_n(t, r^k, \theta^k)] \frac{\partial}{\partial a_\nu} [a^n B_n(t, r^k, \theta^k)] \right\} dt \\
&= 2 \sum_i \frac{0 - B_\mu(t_i, r_i, \theta_i) B_\nu(t_i, r_i, \theta_i)}{R(t_i; r_i, \theta_i)} - (\bar{\tau} - \underline{\tau}) \sum_k \int [0 + B_\mu(t, r_k, \theta_k) B_\nu(t, r_k, \theta_k)] dt
\end{aligned} \tag{3.5}$$

将基底再拆解为 Legendre 与 Zernike，考虑 PMT 编号，并取负号：

$$\begin{aligned}
-\frac{\partial^2 \log \mathcal{L}}{\partial a_{mn} \partial a_{pq}} &= 2 \sum_{i=1}^{N_{\text{PE}}} \frac{P_m(t_i) Z_n(r_i, \theta_{j_i}) P_p(t_i) Z_q(r_i, \theta_{j_i})}{R(t_i; r_i, \theta_i)} + (\bar{\tau} - \underline{\tau}) \sum_{j=1}^{N_{\text{PMT}}} \sum_{k=1}^{N_{\text{ev}}} \int_{-1}^1 P_m(t) Z_n(r_i, \theta_{j_i}) P_p(t) Z_q(r_i, \theta_{j_i}) dt \\
&= 2 \sum_{i=1}^{N_{\text{PE}}} \frac{P_m(t_i) Z_n(r_i, \theta_{j_i}) P_p(t_i) Z_q(r_i, \theta_{j_i})}{R(t_i; r_i, \theta_i)} + (\bar{\tau} - \underline{\tau}) \sum_{j=1}^{N_{\text{PMT}}} \sum_{k=1}^{N_{\text{ev}}} \delta_{mp} Z_n(r^k, \theta_j^k) Z_q(r^k, \theta_j^k)
\end{aligned} \tag{3.6}$$

是型如 $\sum_i A_i A_i^\top$ 的 Hessian。其特征值非负，半正定。可以使用凸优化方法求解 $\log \mathcal{L}$ 的全局极值和 \hat{a}^{mn} 。

3.2 随机近似

将 $\{r_i, \theta_i\}$ 分为若干非交子集，记做 $\text{Bin}_{(r, \theta)}$ 。若每个子集测度近似相等且元素个数足够多，对应于高光产额下同一位置发射的光子足够多，则有，

$$\begin{aligned}
&\because P((t, t+dt]) = R(t; r, \theta) dt \\
&\therefore \sum_i \sum_{t_i \in \text{Bin}_{(r_i, \theta_i)}} \frac{P_m(t_i) Z_n(r_i, \theta_i) P_p(t_i) Z_q(r_i, \theta_i)}{R(t_i; r_i, \theta_i)} \\
&\approx \sum_i |\text{Bin}_{(r_i, \theta_i)}| E_{t_i} \frac{P_m(t_i) Z_n(r_i, \theta_i) P_p(t_i) Z_q(r_i, \theta_i)}{R(t_i; r_i, \theta_i)} \\
&= \sum_i \int_{-1}^1 \frac{P_m(t) Z_n(r_i, \theta_i) P_p(t) Z_q(r_i, \theta_i)}{R(t; r_i, \theta_i)} R(t; r_i, \theta_i) dt \\
&= \sum_i \delta_{mp} Z_n(r_i, \theta_i) Z_q(r_i, \theta_i)
\end{aligned} \tag{3.7}$$

因此总 Hessian 近似为：

$$-\frac{\partial^2 \log \mathcal{L}}{\partial a_{mn} \partial a_{pq}} \approx 2 \sum_{i=1}^{N_{\text{PEs}}} \delta_{mp} Z_n(r_i, \theta_i) Z_q(r_i, \theta_i) + (\bar{\tau} - \underline{\tau}) \sum_{j=1}^{N_{\text{PMT}}} \sum_{k=1}^{N_{\text{ev}}} \delta_{mp} Z_n(r^k, \theta_j^k) Z_q(r^k, \theta_j^k) \tag{3.8}$$

相比 (3.6) 可以减少计算量。

3.3 Probe 整合

对 N 个不同数据集使用相同阶数的多项式进行拟合，得到 N 个 Probe。每个 Probe 对应一组多项式拟合系数 \mathbf{a}_i ($i = 1, 2, \dots, N$)。对于每个 Probe，假设任意一组同等维数的系数 \mathbf{a} 服从均值为 \mathbf{a}_i ，协方差矩阵

为 Λ_i^{-1} 的多维高斯分布，可以得到其似然函数服从

$$\mathcal{L}_i(\mathbf{a}) \sim \exp \left[-\frac{(\mathbf{a} - \mathbf{a}_i)^T \Lambda_i (\mathbf{a} - \mathbf{a}_i)}{2} \right] \quad (3.9)$$

其中 $\Lambda_i = -\frac{\partial^2 \log \mathcal{L}_i}{\partial \mathbf{a} \partial \mathbf{a}^T}$ ，为在拟合过程中求得的 Hessian 矩阵。总似然函数应为

$$\mathcal{L}(\mathbf{a}) = \prod_{i=1}^N \mathcal{L}_i(\mathbf{a}) \sim \exp \left[-\sum_{i=1}^N \frac{(\mathbf{a} - \mathbf{a}_i)^T \Lambda_i (\mathbf{a} - \mathbf{a}_i)}{2} \right] \quad (3.10)$$

整合 Probe 的系数 $\mathbf{a} = \hat{\mathbf{a}}$ 应使 (3.10) 最大，即使 (3.11) 最小

$$\begin{aligned} \mathbf{X}(\mathbf{a}) &= \sum_{i=1}^N (\mathbf{a} - \mathbf{a}_i)^T \Lambda_i (\mathbf{a} - \mathbf{a}_i) \\ &= \sum_{i=1}^N (\mathbf{a}^T \Lambda_i \mathbf{a} - \mathbf{a}_i^T \Lambda_i \mathbf{a} - \mathbf{a}^T \Lambda_i \mathbf{a}_i + \mathbf{a}_i^T \Lambda_i \mathbf{a}_i) \\ &= \mathbf{a}^T \left(\sum_{i=1}^N \Lambda_i \right) \mathbf{a} - \left(\sum_{i=1}^N \mathbf{a}_i^T \Lambda_i \right) \mathbf{a} - \mathbf{a}^T \left(\sum_{i=1}^N \Lambda_i \mathbf{a}_i \right) + \left(\sum_{i=1}^N \mathbf{a}_i^T \Lambda_i \mathbf{a}_i \right) \end{aligned} \quad (3.11)$$

令 $\mathbf{X}(\mathbf{a})$ 对 \mathbf{a} 求一阶偏导并令其为 0，可得

$$\frac{\partial \mathbf{X}(\hat{\mathbf{a}})}{\partial \mathbf{a}} = 2 \left[\left(\sum_{i=1}^N \Lambda_i \right) \hat{\mathbf{a}} - \sum_{i=1}^N \Lambda_i \mathbf{a}_i \right] = 0 \quad \Rightarrow \quad \hat{\mathbf{a}} = \left(\sum_{i=1}^N \Lambda_i \right)^{-1} \sum_{i=1}^N \Lambda_i \mathbf{a}_i \quad (3.12)$$

第四章 数据驱动的似然比估计

刘卓轩、温欣洋、褚衍博、郝传晖、续本达

4.1 amortized approximate likelihood ratio MCMC

该方法利用了似然比检验的原理。似然比检验 (likelihood ratio test) 是利用似然函数来检测某个假设 (或限制) 是否有效的一种检验。一般情况下, 要检测某个附加的参数限制是否是正确的, 可以将加入附加限制条件的较复杂模型的似然函数最大值与之前的较简单模型的似然函数最大值进行比较。如果参数限制是正确的, 那么加入这样一个参数应当不会造成似然函数最大值的大幅降低。因此, 我们倾向于接受让似然比最大的参数作为真实的参数。

对于一个真实的分类问题, 当 $X \sim P(X|\theta^*)$ 时 $\text{label}_y = 1$, 当 $X \sim P(X|\theta)$ 时 $\text{label}_y = 0$, 此时对于一个分类器有

$$d^*(x) = P(y=1|X) = \frac{P(X|\theta^*)}{P(X|\theta) + P(X|\theta^*)} = \frac{r(x|\theta^*, \theta)}{1 + r(x|\theta^*, \theta)} \quad (4.1)$$

可以从中计算出似然比为

$$r(x|\theta^*, \theta) = \frac{d^*(x)}{1 - d^*(x)} \quad (4.2)$$

从上述原理可以看出, 我们可以训练一个最优的分类器, 再通过该分类器计算得到似然函数比。以下简称该方法为 AALR。

4.2 利用 AALR 方法训练得到球对称模型下泊松过程到达率 λ

原理图如下图所示:

参见图 4.1, 我们利用 AALR 的方法, 通过训练一个分类器来得到泊松过程的到达率 λ 。我们给这个网络输入的参数有: 事件相对于 PMT 的位置参数 r 、 θ 以及到达 PMT 的时间序列 T_vector 或者到达 PMT 的光子个数 N 。在训练前, 我们会给出一个参考到达率 $\lambda_{ref}(r, \theta)$ 。神经网络首先会给出一个建议到达率 $\lambda_{neuro}(r, \theta)$, 结合我们的参考到达率 $\lambda_{ref}(r, \theta)$, 可以显式地写出其分类函数:

$$d^*(x) = \frac{f(X|\lambda_{neuro}(r, \theta))}{f(X|\lambda_{neuro}(r, \theta)) + f(X|\lambda_{ref}(r, \theta))} \quad (4.3)$$

我们称其为参考分类器。利用与 AALR 类似的方法, 通过后向传播将该分类器的 BCE 传播回到生成建议到达率 $\lambda_{neuro}(r, \theta)$ 的网络中, 进而让其梯度下降得到更好的建议到达率 $\lambda_{neuro}(r, \theta)$ 。在不断训练的过程中, 参考分类器会逐渐趋近于真实分类器, 即

$$d^*(x) = \frac{f(X|\lambda_{neuro}(r, \theta))}{f(X|\lambda_{neuro}(r, \theta)) + f(X|\lambda_{ref}(r, \theta))} \rightarrow \frac{f(X|\lambda_{true}(r, \theta))}{f(X|\lambda_{true}(r, \theta)) + f(X|\lambda_{ref}(r, \theta))} \quad (4.4)$$

此时我们可以认为 $\lambda_{neuro}(r, \theta) \rightarrow \lambda_{true}(r, \theta)$, 即找到了泊松过程的参数 λ

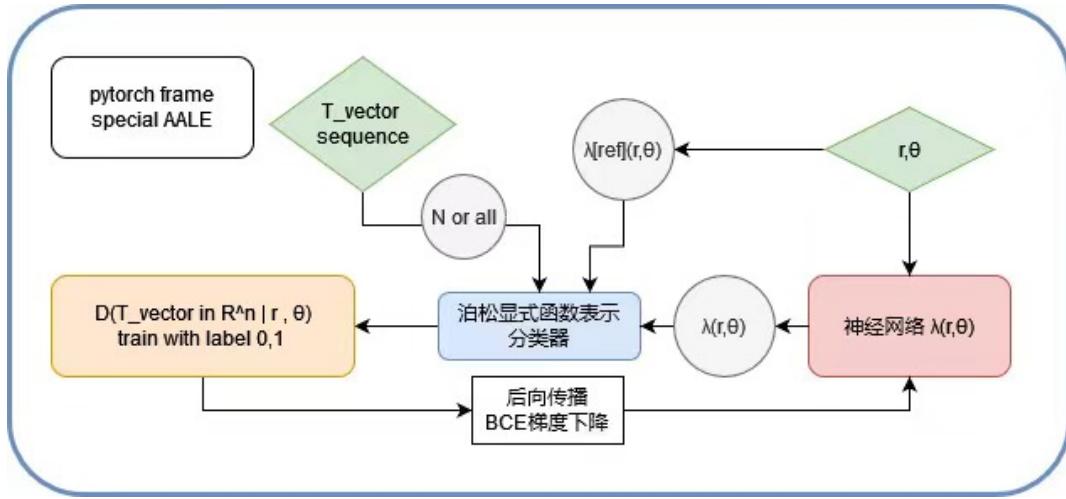


图 4.1: 利用 AALR 方法训练得到球对称模型下泊松过程到达率

4.3 含有时间的 AALR 解决方案

4.3.1 问题概述

对于非齐次泊松过程，单位时间的到达强度“到达率”定义为 $R(r, \theta, \tau) = \frac{d\lambda(r, \theta, \tau)}{d\tau}$ 。由于泊松过程的无记忆性，对任意时刻 T 的“到达强度”分布为： $\lambda(r, \theta, T) = \int_0^T R(r, \theta, \tau) d\tau$ 因此在时间 0-T 之间的到达数目满足非齐次泊松分布：

$$P(N(T)|\lambda(r, \theta, T)) = \frac{\lambda(T)^N(T)}{N(T)!} e^{-\lambda(T)} \quad (4.5)$$

4.3.2 方案 1：使用 t-vector 方法

t-vector 是同一 PMT 上的 PE 到达时间序列。我们可以直接使用 t-vector 和 $R(t)$ 计算分类器概率：

$$P(t_1, t_2, \dots, t_N|R(t)) = \frac{\lambda^N}{N!} e^{-\int R(\tau) d\tau} \prod_{i=1}^N \frac{R(t_i)}{\lambda} = \frac{\prod_{i=1}^N R(t_i)}{N!} e^{-\int R(\tau) d\tau} \quad (4.6)$$

可以看到 (4.6) 内部有积分和微分，第一种思路是计算 $R(t)$ 的积分，第二种思路是计算 λ 的微分，计算积分需要我们手写 torch 模块，或者先用神经网络训练 λ ，再用另一个神经网络训练 $R(t)$ 的分布。

4.3.3 方案 2：对时域进行采样

若利用非齐次泊松过程的 t-vector 来描述到达事件数 $N(t)$ ，则其在时间域上应该是一系列阶跃函数的和： $N(t) = \sum_{i=1}^N H(t - t_i)$ 其中 $H(t)$ 为阶跃函数。故可以利用直方图在时间域上采样作为含时数据 $N(r, \theta, T)$ ，之后可以利用非齐次泊松分布的性质得到：

$$P(N(r, \theta, T)|\lambda(r, \theta, T)) = \frac{\lambda(r, \theta, T)^N(r, \theta, T)}{N(r, \theta, T)!} e^{-\lambda(r, \theta, T)} \quad (4.7)$$

即将时间作为一个新的维度变量，从而得到函数 $\lambda(r, \theta, t)$ ，再对其微分得到 $R(r, \theta, t)$ 。此方法不需要引入新的积分微分计算，是目前最有希望的方法之一。

4.3.4 应用 1: JUNO

类似于 TAO 数据集，若考虑时间，则输入参数变为 (r, θ, t) ，并将 $T - vecor$ 转化为时间域上的函数 $N(t)$ ，得到相应数据集和后验概率，即可使用 AALR 算法。注意：1. 为了保证神经网络输出稳定，需要设置积分的时间区间。2. 为了获得多种特征需要快速计算特殊多项式的方法，例如递推计算法。

4.3.5 应用 2: Osiris

由于对称性减少，需要额外考虑 z 轴维度，输入参数为 (r, z, θ, t) ，其余无太大变化，只不过参数量需求变大，训练时间翻倍。

4.3.6 目前的安排

- 获取并使用 JUNO 的含时数据集
- 获取并使用 Osiris 不含时的数据集
- 获取并使用 Osiris 含时的数据集

第五章 其它

窦威、武益阳、刘峻江、续本达

我们对 probe 的探索是漫长的，有不少没有完全实现的思想值得记录。

5.1 直方图近似

若直接取数据 $\{(t_i, r_i, \theta_{j_i})\}$ 的直方图或者核密度估计，其近似的分布是 $R_j(t; r, \theta_j)\rho(r, \theta_j)$ 。若求 $R_j(t; r, \theta_j)$ 需要扣除 $\rho(r, \theta_j)$ 的影响。当数据在在三维球中均匀生成时， $\rho(r, \theta_j) = r^2 \sin \theta_j$ 。

5.2 变分推断

与函数基分解和直方图一样，变分推断提供了一般性的近似 $R_j(t; r, \theta_j)$ 的方法。从 R 中提出归一化因子 λ 写作 $\lambda_j(r, \theta_j)R_*(t; r, \theta_j)$ ，仿照 (2.3)，

$$\begin{aligned} & \log \mathcal{L}[R_j(t; r, \theta_j)] \\ &= \log \prod_i [R_{j_i}(t_i; r_i, \theta_{j_i})] - \sum_{k,j} \int R_j(t; r^k, \theta_j^k) dt \\ &= \sum_i [\log \lambda_j(r_i, \theta_{j_i}) + \log R_*(t_i; r_i, \theta_{j_i})] - \sum_{k,j} \lambda_j(r^k, \theta_j^k) \end{aligned} \tag{5.1}$$

使用变分推断的近似框架例如 normalized flow 逼近 $R_*(t; r, \theta_j)$ ，可能得到比多项式近似数值上更稳定的结果。