

底稿：

0、假设：数据假设&模型假设

符号：另外处理

一、数据处理&模型简介&状态的描述

数据分为两类：一类描述具体比赛得分状态，一类描述本球技术类型

分类展示（）

571	1303	Frances Tiafoe	Grigor Dimitrov	24:58:22
588	1303	Frances Tiafoe	Grigor Dimitrov	25:12:47

模型解释：（假设里会有公式，待补充）

我们认为，选手的技术（含体力等因素）与心态会影响到对局得分

为了利于分析，我们在技术层面分析所有的原始技术变量，即心态层面只涉及到对于比分的影响

这两个因素都会随着时间而变化，我们假设其满足马尔可夫性假设。

而这两者是“得分”这一观察变量的隐变量

假设 2 观测独立性假设即任意时刻的观测状态只仅仅依赖于当前时刻的隐藏状态。

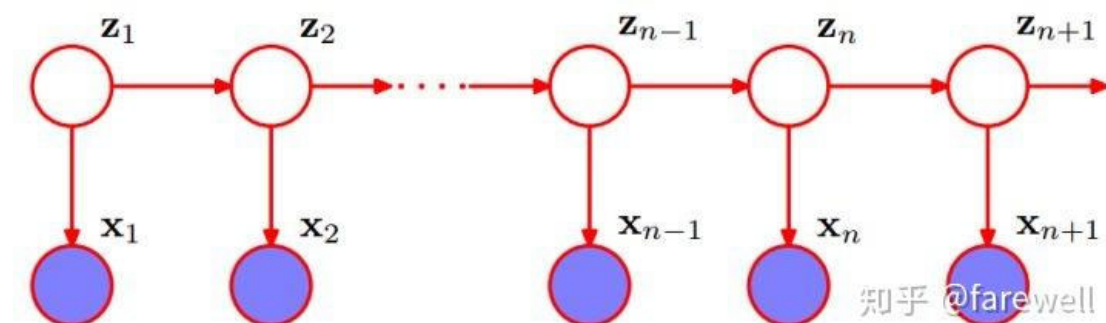
即得分仅依靠于此时的技术与心态

则将随机变量作为结点，以两个随机变量相关或者不独立连边，构成一个网络。我们认为这些边为有向边，即每一球的心态不受下一球的印象，且本球的得分不决定本球的技术与心态。

此网络为有向无环图（DAG），考虑其底图（Undirecter graph）的性质：

- 最大块为 2
- 块-割点图严格成链。
- 所有边都是割边

（网络示意图）



状态变量就是 z_n ，观测变量就是 x_n $n = 1, 2, \dots$

纵向箭头表示观测方程，横向箭头表示状态（转移）方程。

（有水印，4 重画）

对于顺序数据来说，之前的概率图描述了两个重要的模型。如果潜在变量是离散的，那么我们得到了隐马尔科夫模型。但此时潜在变量和观测变量都是离散变量，那么我们就得到了动态系统。特别的，如果认为潜在变量和观测变量满足线性高斯条件（结点的条件概率分布对于父结点的依赖是线性高斯的形式），则此时系统为线性动态系统 (linear dynamical system)。

动态系统是一种时间序列模型，我们可以在这个模型中对某个变量进行后验估计，这个变量叫状态变量。

但是，我们无法直接获取到状态变量的信息，只能间接地获取到与状态变量相关的隐变量的信息，并且这个信息是带有噪声的。叫观测变量/隐变量/先验变量。观测变量的分布由状态变量的取值确定，这个条件概率分布叫作**观测方程**。

另外新的观测发生的时刻，状态变量的分布与上一个观测时刻状态变量的取值相关，由**状态（转移）方程**决定。

其中，“分布”体现出转移过程是有噪声的。

我们做出如下假设：（对整体成立的）

预备知识一：卡尔曼滤波中的条件独立

1. 当给定某一时刻的状态变量 z_n 时，该时刻的观测变量 x_n 与之前的任意观测变量条件独立，即

$$p(x_n | z_n, x_{n-1}, \dots, x_1) = p(x_n | z_n) \quad (4)$$

这一点很好理解。根据卡尔曼滤波模型的假设，当我们知道了状态变量的取值时，那么对应的观测变量的分布就确定了，它与更早之前的观测无关（事实上与更早之前的任意状态变量也无关），状态变量的取值就是所需的全部信息。

2. 当给定某一时刻的状态变量 z_{n-1} 时，下一时刻的状态变量 z_n 与之前的任意观测变量条件独立，即

$$p(z_n | z_{n-1}, x_{n-1}, \dots, x_1) = p(z_n | z_{n-1}) \quad (5)$$

这一点也很好理解。根据卡尔曼滤波模型的假设，当我们知道了上一时刻状态变量的取值时，那么下一时刻状态变量的分布就确定了，它与更早之前的观测无关（事实上与更早之前的任意状态变量也无关），前一时刻状态变量的取值就是所需的全部信息。

（例子或许添加，便于解释）

为了说明线性动态系统的概念，让我们考虑下面这个简单的例子，它经常在实际问题中出现。假设我们希望使用一个有噪声的传感器测量一个未知量 z 的值，传感器返回一个观测值 x ，表示 z 的值加上一个零均值的高斯噪声。给定一个单次的测量，我们关于 z 的最好的猜测是假设 $z = x$ 。然而，我们可以通过取多次测量然后求平均的方法提高我们对 z 的估计效果，因为随机噪声项倾向于彼此抵消。现在，让我们将情况变得更复杂。假设我们希望测量一个随着时间变化的量 z 。我们可以对进行常规的测量 x ，从而我们得到了 x_1, \dots, x_N ，我们希望找到对应的 z_1, \dots, z_N 。如果我们简单地对测量求平均，那么由于随机噪声产生的误差会被消去，但是不幸的是我们会仅仅得到一个单一的平均估计，对 z 的变化进行了平均，从而引入了一种新的误差。

Data Cleaning and Visualization

考虑对所有比赛的技术数据做分析，但 2023-wimbledon-1310 除外（该场的 rally 和 speed_mph

全部是无效值），采用刚刚的第二种方法，即选择两位选手体力总/最近 20 球体力消耗的对比而非总里程

（在这里的数据处理包含了发球，特殊写入）

数据挖掘&特征工程：

记状态向量为 Z

（默认所有变量为关于 N 的函数，即代表第 N 球的状态，如 Z 应为 $Z(N)$ ，后全部省略 “(N)” ，但 N 的含义保留）

Z 的分量为：

① 近期得分情况导致的心态波动

② 技术性因素

分别记为 V ， Q ，即 $Z=[V, Q]$

其中 V 是取值为 $[0,1]$ 的离散变量，0 代表劣势，1 代表优势

， Q 是取值为 $[0,1]$ 的离散变量，0 代表优势，1 代表劣势

● V

$V=W_V-AVG(V_0, V_1, V_2)$ ——加权平均

V_0 是关于近 mp (memorized points) 球的变量，取加权平均 Q_{v0}

前 mp 球的每球表现 对应下一球的预测

V_1 是关于近 1 game 得分 的变量，设本 game 得分为 n ，本人得分 m

前 game 的每球表现 对应下一 game 的预测

V_2 是关于本 set 的拿下 game 数的变量，由马尔科夫链得到的获胜概率

前 set 的每 game 表现 对应本 set 的预测

其中 V_0 关注得分的位置/次序/时间， V_1 、 V_2 不关注，只关注最终比例

V_0 的确定采用**滑动窗口**的思路处理数据

$$mp = \begin{cases} 0, \text{match 前 2 球} \\ 3, \text{game 前 3 球 且 非 match 前 2 球} \\ x, \text{game 前 } x \text{ 球} \end{cases}$$

即：

整 match 的前 2 球，取 0

剩余所有时间，取【本 game 球数，3】的最小值，作为记忆 mp

$$V_0 = \begin{cases} 0.5, mp = 0 \\ \sum_{i=1}^{mp} W_{N+1-i} * Q_{v0}(mp, N+1-i), mp \neq 0 \end{cases}$$

其中 W_{N+1-i} 指向此盘前 $i-1$ 盘是否胜利，权重 Q_{v0} 遵循

$$Q_{v0}(mp, N+1-i) = \frac{mp+1-i}{mp(mp+1)/2}$$

也即：按 mp 球来算，第一球占比 mp ，第二球占比 $mp-1$ ，第三球占比 $mp-$

2，...，第 mp 球占比 1，后除以综合求出权重

$$V_1(n, m) = \frac{\sum_{i=0}^{m-1} C_{n-1}^i}{2^{n-1}}$$

需要满足

$$V_1(n, m) + V_1(n, n-m) = 1$$

即，若某 game 中，player1's point : player2's point = 6:4，

则在下一 game 中，对于 player1，n=10，m=6，

考虑杨辉三角第 9 层：

$$1 \ 9 \ 36 \ 84 \ 126 \ 126 \ 84 \ 36 \ 9 \ 1$$

player1 获得前 6 个数的权重，player2 获得前/后 4 个数，

即 player1 的 $V_1 = \frac{1+9+36+84+126+126}{1+9+36+84+126+126+84+36+9+1} = \frac{382}{512}$

$$V_2(n, m) = n, m \text{ 状态的胜率}$$

马尔可夫链模型（Markov Chain Model）：将比赛状态建模为马尔可夫链，推断获胜的概率作为 M 的值——

马尔可夫链是一组具有马尔可夫性质的离散随机变量的集合。具体地，对概率空间 $(\Omega, \mathcal{F}, \mathbb{P})$ 内以一维可数集为指标集 (index set) 的随机变量集合 $\mathbf{X} = \{X_n : n \geq 0\}$ ，若随机变量的取值都在可数集内： $X = s_i, s_i \in \mathcal{S}$ ，且随机变量的条件概率满足如下关系 [2]：

$$p(X_{t+1} | X_t, \dots, X_1) = p(X_{t+1} | X_t)$$

则 \mathbf{X} 被称为马尔可夫链，可数集 $\mathcal{S} \in \mathbb{Z}$ 被称为状态空间 (state space)，马尔可夫链在状态空间内的取值称为状态 [2]。这里

上式在定义马尔可夫链的同时定义了马尔可夫性质，该性质也被称为“无记忆性 (memorylessness)”，即 t+1 步的随机变量在给定第 t 步随机变量后与其余的随机变量条件独立 (conditionally independent)： $X_{t+1} \perp (X_{t-1}, X_0) | X_t$ [2]。在此基础

以本次比赛的数据分析每种状态的状态转移函数，做出了状态转移图。。。 (有实际关系的压缩的图)

此马尔科夫链是有限的，为一种吸收链。所以我们可以通过此模型，逆序递推出每种状态的最终胜率

递推公式： $P(statusA) = \sum_{status \text{ 可达}} P(status) * T(statusA \rightarrow status)$

结果 (表)。。。 (列表 0,1,2,3,4,5,6)

0	1	2	3	4	5	6
---	---	---	---	---	---	---

0	0.50	0.57	0.57	0.92	0.99	0.99	
1	0.43	0.50	0.66	0.79	0.99	0.99	
2	0.22	0.34	0.50	0.69	0.90	0.95	
3	0.08	0.21	0.31	0.50	0.72	0.98	
4	0.01	0.15	0.10	0.28	0.50	0.76	
5	0.01	0.01	0.05	0.02	0.24	0.50	0.65
6						0.35	0.50

综合：

以一定权重—— W_v 确定为 7:2:1 综合 V_0, V_1, V_2 , 得到 V 综合, 数据如下：

。。。 (折线可视化结果)

● Q:

主成分分析 PCA

简介：为了对“技术指标”降维，我们使用主成分分析，得到权重最重要的 10 个指标，对每个向量按照实际意义处理与解释，经过平滑化、标准化（正太化）、集中化的数据处理，得到每球的 Q 值

解释 PCA 模型：

共使用数据清洗后的 6966-314 个样本

29 个指标分别为 (改进列举)

who_to_serve consumption_of_strength(最近 20 球消耗体力比值)

consumption(总体力消耗比值)

p1_ace p2_ace p1_winner p2_winner p1_double_fault p2_double_fault

p1_unf_err p2_unf_err

p1_net_pt p2_net_pt p1_net_pt_won p2_net_pt_won p1_break_pt

p2_break_pt p1_break_pt_won

p2_break_pt_won p1_break_pt_missed p2_break_pt_missed

p1_serve_speed_ave

p2_serve_speed_ave width_1 width_2 depth_1 depth_2 depth_return_1

depth_return_2 的数据

假设有 n 个样本, p 个指标, 则可构成大小为 $n \times p$ 的样本矩阵 x :

$$x = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} = (x_1, x_2, \cdots, x_p)$$

则

假设我们想找到新的一组变量 $z_1, z_2, \dots, z_m (m \leq p)$, 且它们满足:

$$\begin{cases} z_1 = l_{11}x_1 + l_{12}x_2 + \dots + l_{1p}x_p \\ z_2 = l_{21}x_1 + l_{22}x_2 + \dots + l_{2p}x_p \\ \vdots \\ z_m = l_{m1}x_1 + l_{m2}x_2 + \dots + l_{mp}x_p \end{cases}$$

- (1) z_i 与 $z_j (i \neq j; i, j = 1, 2, \dots, m)$ 相互无关;
- (2) z_1 是 x_1, x_2, \dots, x_p 的一切线性组合中方差最大者;
- (2) z_2 是与 z_1 不相关的 x_1, x_2, \dots, x_p 的所有线性组合中方差最大者;
- (3) 依次类推, z_m 是与 z_1, z_2, \dots, z_{m-1} 不相关的 x_1, x_2, \dots, x_p 的所有线性组合中方差最大者。
- (4) 新变量指标 z_1, z_2, \dots, z_m 分别称为原变量指标 x_1, x_2, \dots, x_p 的第一, 第二, \dots , 第 m 主成分。

这组 z 能较好的反映 x 变量的特征, 在尽可能减少取消信息熵的前提下, 完成降维, 便于处理。

通过 SPSS 软件分析, 得到的主成分与贡献值累加和为 (可视化结果, 显示 10 项)

为了保证准确性, 拟打算取 10 项, 得到 80% 的贡献

对每个向量, 忽略一些影响较小的因素, 尽可能的每个变量做到线性相关, 如图, 并给出指标的理解

注意到可以正相关可以负相关, 但是在一个向量内必须同向

(可视化结果)

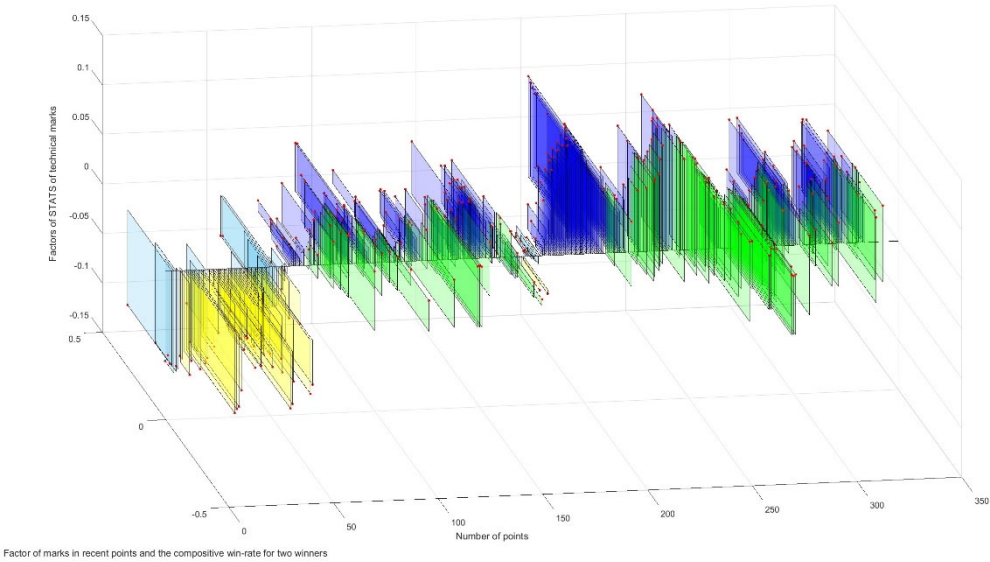
数据处理的方法：

经过平滑化、标准化（正太化）、集中化

(数据可视)

● 综合可视

最后可视化：



并依照图例分析经典对局：

轴解释

颜色解释

面积解释

第一局，黄块多，德约尽显优势

第二局，绿块多，且接近轴，说明双方难舍难分，但有些许蓝块，阿尔卡艰难赢下此局

第三局，蓝块明显，阿尔卡尽显优势

第四局，绿块多，偏向德约

第五局，最后还是蓝块多，阿尔卡拿下终局

二、V 数据处理的评估&教练的反驳

swings in play 分析

STEP 1 swings in play 的量化

swings in play 理解为优势转换，

定义为当点数差大于 5 时，局势判定为有利，则 swings in play 发生在局势转换的时候

STEP 2 蒙特卡洛模拟 200 次

考虑到构建教练的模型，涉及到对选手实力的评估，每球的随机因素，以及发球方
选手实力的评估，直接考虑一段够长的时间内选手得分的比率，我们在数据处理中直接用终
局得分的比例，最能接近选手的实力比

考虑到发球方，我们对是否自身发球，乘以“优势因素”（不给具体数值）

之后模拟 200 次

最后得到的分布为：（player1、player2 柱状可视化结果展示）

STEP 3 分析比较

对于 player1:

M 预测筛选后的均值 0.654，方差 0.02

R 预测筛选后的均值 0.568，方差 0.13

（散点可视化结果比较展示）

方差更小更稳定的预测，趋向也更符合（趋近 1）

按照逻辑斯蒂得分预估：

M 的得分为 7，R 的得分为 1.4

对于 player2:

M 预测筛选后的均值 0.317，方差 0.012

R 预测筛选后的均值 0.508，方差 0.100

（散点可视化结果比较展示）

方差更小更稳定的预测，趋向也更符合（趋近 0）

按照逻辑斯蒂得分预估：

M 的得分为 8.9，R 的得分为 0.6

考虑到希望对这两种预测模型的更综合的评分，我们构建了以逻辑斯蒂函数为基础的预测评分模型。

逻辑斯蒂函数的介绍与特性：

逻辑斯蒂函数是一种常见的 S 函数，连续函数、单调递增、当自变量为 0 时，因变量为 0.5，
当自变量接近正无穷，因变量接近 1，当自变量接近负无穷，因变量接近 0，

一个简单的Logistic函数可用下式表示：

$$P(t) = \frac{1}{1 + e^{-t}}.$$

评价体系：

我们考虑逻辑斯蒂的反函数，他有如下特性：

$$y = \ln x - \ln(1 - x)$$

预测为 0.5，值为 0，函数越接近线性

预测越接近 1，值越接近正无穷，增势平缓

预测越接近-1，值越接近负无穷，减势平缓

考虑到单次预测不能对整体的得分造成太多的影响，设置一个单次得分的限度，本模型设置为[-2,2]，当得分偏离限度时，取值超过限度时，控制为最接近得分的限度值。

逻辑斯蒂函数、 y' 函数与 y 关系、复合后函数，画图展示

对每个预测值评分后平均，得到对预测模型的评估

runs of success 分析

1 runs of success 的量化

runs of success 理解为连续得分，则

$$ROS_player1(N) = PTS_player1(N) \& PTS_player1(N-1) \& PTS_player1(N-2)$$

2 蒙特卡洛模拟 200 次 (player1、player2 柱状可视化结果展示)

3 分析比较

对于 player1:

M 预测筛选后的均值 0.624，方差 0.02

R 预测筛选后的均值 0.612，方差 0.13

(散点可视化结果比较展示)

方差更小更稳定的预测，趋向也更符合（趋近 1）

按照逻辑斯蒂得分预估：

M 的得分为 5.6，R 的得分为 3.1

对于 player2:

M 预测筛选后的均值 0.387，方差 0.02

R 预测筛选后的均值 0.468，方差 0.08

(散点可视化结果比较展示)

方差更小更稳定的预测，趋向也更符合（趋近 0）

按照逻辑斯蒂得分预估：

M 的得分为 5.7，R 的得分为 1.9

momentum 也不是随机的，而是与时间/赛程相关 (ARIMA)

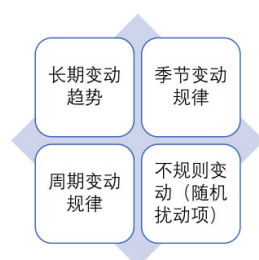
介绍

1 momentum 是时序的

时间序列分解

因为时间序列是某个指标数值长期变化的数值表现，所以时间序列数值变化背后必然蕴含着数值变换的规律性，这些规律性就是时间序列分析的切入点。

一般情况下，时间序列的数值变化规律有以下四种：



一个时间序列往往是以上四类变化形式的叠加。

2 ARIMA

ARIMA(p,d,q)模型

我们之前探讨的 $\{y_t\}$ 均是平稳的时间序列，然而时间序列可能是 d 阶的单位根过程。为此，我们需要先对数据进行差分处理，将其转换为平稳的时间序列后再进行建模。

$$\text{ARIMA}(p, d, q) \text{模型: } y'_t = \alpha_0 + \sum_{i=1}^p \alpha_i y'_{t-i} + \varepsilon_t + \sum_{i=1}^q \beta_i \varepsilon_{t-i}$$

$$\text{且 } y'_t = \Delta^d y_t = (1 - L)^d y_t$$

$$(1 - \sum_{i=1}^p \alpha_i L^i)(1 - L)^d y_t = \alpha_0 + (1 + \sum_{i=1}^q \beta_i L^i) \varepsilon_t$$

↑ ↑
AR(p) d阶差分

↑
MA(q)

差分自回归移动平均模型
(Autoregressive Integrated Moving Average Model)

AR(p)模型(auto regressive)

p 阶的自回归模型 (AR(p)模型) :

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots + \alpha_p y_{t-p} + \varepsilon_t$$

ε_t 是方差为 σ^2 的白噪声序列

(自回归: 将自己的1至 p 阶滞后项视为自变量来进行回归)

MA(q)模型(moving average)

q 阶移动平均过程 (MA(q)模型) :

$$y_t = \varepsilon_t + \beta_1 \varepsilon_{t-1} + \beta_2 \varepsilon_{t-2} + \cdots + \beta_q \varepsilon_{t-q}$$

ACF自相关系数

(1) 相关系数

X 和 Y 是两个随机变量, 则 $\rho = \frac{\text{cov}(X,Y)}{\sqrt{\text{Var}(X)}\sqrt{\text{Var}(Y)}}$ 称为 X 和 Y 的相关系数 (用来表示 X 和 Y 的线性关系)。

(2) 自相关系数(*autocorrelation*)

我们探讨的时间序列假定是平稳的: (1) $E(x_t) = E(x_{t-s}) = u$ (2) $\text{Var}(x_t) = \text{Var}(x_{t-s}) = \sigma^2$ (3) $\text{Cov}(x_t, x_{t-s}) = \gamma_s$

我们定义自相关系数: $\rho_s = \frac{\text{cov}(x_t, x_{t-s})}{\sqrt{\text{Var}(x_t)}\sqrt{\text{Var}(x_{t-s})}} = \frac{\gamma_s}{\gamma_0}$ 注意: $\gamma_0 = \text{Cov}(x_t, x_t) = \text{Var}(x_t)$

如果我们将 ρ_s 视为 s 的函数, 则我们称该函数为自相关函数(*autocorrelation function*), 其函数图像称为自相关图(*correlogram*).

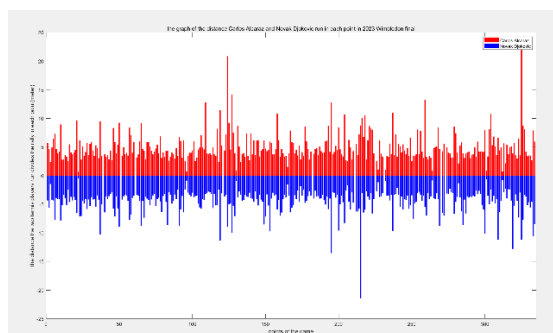
(自相关系数 ρ_s 表示了一个平稳序列 $\{x_t\}$ 中, 间隔 s 期的两个时间点之间的相关系数)

因为真实的自相关系数是未知的, 我们只能依赖样本数据进行估计:

$$\text{样本自相关系数: } r_s = \hat{\rho}_s = \frac{\sum_{t=s+1}^T (x_t - \bar{x})(x_{t-s} - \bar{x})}{\sum_{t=1}^T (x_t - \bar{x})^2}$$

注意: 如果 $\{\varepsilon_t\}$ 是白噪声序列, 那么 $\rho_s = \begin{cases} 1, & s=0 \\ 0, & s \neq 0 \end{cases}$

模型结果



三、模型结果&预测

为了建立一个可以预测比赛中这些波动的模型。

假设我们的测量方式正确，测量结果存在高斯噪声，即我们的测量结果可能会呈现出高斯分布的噪声。也即我们可以做出线性假设。

下面进入线性动态系统的分析

卡尔曼滤波分析

卡尔曼滤波是线性动态系统的子问题。

只需要搞明白两件事情：

- 1、先验+实验观测->后验，或者说：后验结果
- 2、1 中所述事实在一条[马尔可夫链](#)上反复发生。

问题就可以简单明了的解释明白。

其中我们要预测的结果就是后验，先验是技术因素 Q ，而实验观测即 M
先验就是之前的观测结果，即 M ，

它是一个在[隐马尔可夫链](#)上的[递归贝叶斯估计](#)，对于每一个时刻，上一时刻的状态转移作为本时刻的**先验**，建模为高斯分布，依据贝叶斯公式，通过本时刻建模为高斯分布的**观测**对这个先验进行修正，最后得到本时刻的**后验**，这个后验是前面两个高斯分布的乘积，依然是高斯分布，新的高斯分布的参数：均值和方差可以通过比较系数推导出来。这个后验将作为下一个时刻的先验，如此反复进行。

根据上面的描述，卡尔曼滤波的目标是在已知一系列观测的条件下，求状态变量的后验分布，记为 $\hat{\alpha}(\mathbf{z}_n)$ ，即：

$$\hat{\alpha}(\mathbf{z}_n) = p(\mathbf{z}_n | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \quad (3)$$

我们假设

此，发射概率模型为

$$p(\mathbf{x}_n | \mathbf{z}_n) = N(\mathbf{x}_n | C\mathbf{z}_n, \Sigma) \quad (1)$$

转移概率模型为

$$p(\mathbf{z}_n | \mathbf{z}_{n-1}) = N(\mathbf{z}_n | A\mathbf{z}_{n-1}, \Gamma) \quad (2)$$

其中， A 和 C 分别是线型变换系数矩阵， Γ 和 Σ 分别是高斯分布的方差。

在推导卡尔曼滤波公式的过程中，始终记住一点：我们所有已知的东西只有发射概率模型 $p(\mathbf{z}_n|\mathbf{z}_n)$ 和转移概率模型 $p(\mathbf{z}_n|\mathbf{z}_{n-1})$ ，所以所有推导的步骤都是为了“凑出”这两个条件概率表达式。而且，所有基于贝叶斯方法的推导过程，其实都是在反复利用贝叶斯定理做变换。基于以上两点，我们从目标表达式出发，应用贝叶斯定理：

$$\begin{aligned} p(\mathbf{z}_n|\mathbf{x}_n, \mathbf{x}_{n-1}, \dots, \mathbf{x}_1) &= \frac{p(\mathbf{x}_n|\mathbf{z}_n, \mathbf{x}_{n-1}, \dots, \mathbf{x}_1)p(\mathbf{z}_n|\mathbf{x}_{n-1}, \dots, \mathbf{x}_1)}{p(\mathbf{x}_n|\mathbf{x}_{n-1}, \dots, \mathbf{x}_1)} \\ &= \frac{p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n|\mathbf{x}_{n-1}, \dots, \mathbf{x}_1)}{p(\mathbf{x}_n|\mathbf{x}_{n-1}, \dots, \mathbf{x}_1)} \end{aligned} \quad (6)$$

其中，从第一个等号右边到第二个等号右边利用了（4）式的条件独立性。可以看到，此时已经凑出来了一个 $p(\mathbf{x}_n|\mathbf{z}_n)$ 。

有人可能会说，上面的表达式中，|后面有那么一长串变量，我怎么记得住是怎么推导的？其实非常简单：既然它太长了我们就先把 $\mathbf{x}_{n-1}, \dots, \mathbf{x}_1$ 拿掉，这样就只剩下 $p(\mathbf{z}_n|\mathbf{x}_n)$ ，这时候再利用贝叶斯定理就很简单了吧：

$$p(\mathbf{z}_n|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n)}{p(\mathbf{x}_n)} \quad (7)$$

然后再把刚刚拿掉的 $\mathbf{x}_{n-1}, \dots, \mathbf{x}_1$ 放回到上式的每个概率表达式的|下面，就得到了（6）中第一个等式的变换。

为了推导出目标表达式的递推式，需要对（6）式右边继续变形，直到推出 $p(\mathbf{z}_{n-1}|\mathbf{x}_{n-1}, \dots, \mathbf{x}_1)$ 。观察（6）式，只能对分子的第二项下手了，但是该项没有 \mathbf{z}_{n-1} 怎么办？简单，那就引入 \mathbf{z}_{n-1} 然后再对其积分就可以了：

$$\begin{aligned} p(\mathbf{z}_n|\mathbf{x}_{n-1}, \dots, \mathbf{x}_1) &= \int_{\mathbf{z}_{n-1}} p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{x}_{n-1}, \dots, \mathbf{x}_1)p(\mathbf{z}_{n-1}|\mathbf{x}_{n-1}, \dots, \mathbf{x}_1)d\mathbf{z}_{n-1} \\ &= \int_{\mathbf{z}_{n-1}} p(\mathbf{z}_n|\mathbf{z}_{n-1})\hat{\alpha}(\mathbf{z}_{n-1})d\mathbf{z}_{n-1} \end{aligned} \quad (8)$$

其中，从第一个等号右边到第二个等号右边利用了（5）式的条件独立性。可以看到，此时又凑出来一个 $p(\mathbf{z}_n|\mathbf{z}_{n-1})$ ，同时递推式的前一项 $\hat{\alpha}(\mathbf{z}_{n-1})$ 也出来了，离我们的目标越来越近了。

同样，如果不看与变换无关的变量 $\mathbf{x}_{n-1}, \dots, \mathbf{x}_1$ ，（8）式看似复杂的变换过程其实就是我们熟知的全概率公式：

$$p(\mathbf{z}_n) = \int_{\mathbf{z}_{n-1}} p(\mathbf{z}_n|\mathbf{z}_{n-1})p(\mathbf{z}_{n-1})d\mathbf{z}_{n-1} \quad (9)$$

将（8）式带回（6）式就得到了最终的递推式：

$$\hat{\alpha}(\mathbf{z}_n) = \frac{p(\mathbf{x}_n|\mathbf{z}_n) \int_{\mathbf{z}_{n-1}} p(\mathbf{z}_n|\mathbf{z}_{n-1})\hat{\alpha}(\mathbf{z}_{n-1})d\mathbf{z}_{n-1}}{p(\mathbf{x}_n|\mathbf{x}_{n-1}, \dots, \mathbf{x}_1)} \quad (10)$$

因为在卡尔曼滤波模型中，发射概率模型和转移概率模型都是线型高斯分布，因此基于此推导出的所有概率分布也是高斯分布。高斯分布由均值向量和协方差矩阵表示

$$c_n N(\mathbf{z}_n | \boldsymbol{\mu}_n, \mathbf{V}_n) = N(\mathbf{x}_n | C\mathbf{z}_n, \Sigma) \int N(\mathbf{z}_{n-1} | \boldsymbol{\mu}_{n-1}, \mathbf{V}_{n-1}) N(\mathbf{z}_n | A\mathbf{z}_{n-1}, \mathbf{\Gamma}) d\mathbf{z}_{n-1} \quad (7)$$

利用《机器学习中的高斯分布》一文中关于边缘和条件高斯分布的结论，上式被积式是两个高斯分布的卷积，因此结果也是一个高斯分布，且卷积结果为

$$\int N(\mathbf{z}_{n-1} | \boldsymbol{\mu}_{n-1}, \mathbf{V}_{n-1}) N(\mathbf{z}_n | A\mathbf{z}_{n-1}, \mathbf{\Gamma}) d\mathbf{z}_{n-1} = N(\mathbf{z}_n | A\boldsymbol{\mu}_{n-1}, P_{n-1}) \quad (8)$$

其中，

$$P_{n-1} = AV_{n-1}A^T + \mathbf{\Gamma} \quad (9)$$

这样，（7）式的左右两边分别是两个高斯分布的乘积，形如 $p(x|y)p(y) = p(y|x)p(x)$ ，其中 $p(x)$ 、 $p(y)$ 、 $p(x|y)$ 、 $p(y|x)$ 都是高斯分布，且 $p(x)$ 、 $p(y|x)$ 已知，要求 $p(y)$ 、 $p(x|y)$ 。再次利用《机器学习中的高斯分布》一文中的结论可以得到

$$\boldsymbol{\mu}_n = A\boldsymbol{\mu}_{n-1} + K_n(\mathbf{x}_n - CA\boldsymbol{\mu}_{n-1}) \quad (10)$$

$$\mathbf{V}_n = (I - K_n C)P_{n-1} \quad (11)$$

$$c_n = N(\mathbf{x}_n | CA\boldsymbol{\mu}_{n-1}, CP_{n-1}C^T + \Sigma) \quad (12)$$

其中，

$$K_n = P_{n-1}C^T(CP_{n-1}C^T + \Sigma)^{-1} \quad (13)$$

其中，（10）～（13）式的递推式

（严谨用矩阵作的，下面是简单粗暴的解释）

就是它概率分布的均值。但是现在我们还是没有回答怎么根据粗略估计值的概率分布与雷达的测量值概率分布得到精确估计值的概率分布。其实这个也很简单。直接把这两个概率分布相乘即可。关于为何是直接相乘请参考这个回答《如何通俗并尽可能详细解释卡尔曼滤波？》。这是由贝叶斯滤波所推导得到的。所以我们得到当前时刻导弹位置的精确估计是

$x_t = x$ 在 t 时刻的粗略估计值 $\times z_t = N(6, 0.2^2 + 0.7^2) \times N(7, 0.1^2)$ 。上式乘法的结果是：

$$N(6, 0.2^2 + 0.7^2) \times N(7, 0.1^2) = N(6, 0.53^2) \times N(7, 0.1^2) \\ = N\left(\frac{6 \times 0.1^2 + 7 \times 0.53^2}{0.53^2 + 0.1^2}, \frac{0.1^2 \times 0.53^2}{0.53^2 + 0.1^2}\right)$$

。注意这里用到了高斯分布的乘法公式（这里我贴出来省的大家搜了）

$$N(\mu_1, \sigma_1^2) * N(\mu_2, \sigma_2^2) = N\left(\frac{\sigma_1^2 \mu_2 + \sigma_2^2 \mu_1}{\sigma_1^2 + \sigma_2^2}, \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)$$

。大家可以对比着直观理解那看看有什么相似的地方。事实上可以看到我们就是把方差作为可信度，方差越大越不可信。然后也就是说用方差做权重进行加权和得到均值。

$$x_t = N\left(\frac{6 \times 0.1^2 + 7 \times 0.53^2}{0.53^2 + 0.1^2}, \frac{0.1^2 \times 0.53^2}{(0.53^2 + 0.1^2)^2}\right) = N\left(\left(1 - \frac{0.53^2}{0.53^2 + 0.1^2}\right) * 6 + \frac{0.53^2}{0.53^2 + 0.1^2} * 7, \frac{0.53^2 + 0.1^2}{(0.53^2 + 0.1^2)^2}\right)$$

。而正态分布是在均值那个地方的概率最大。所以当前时刻导弹位置的最优估计就是上面这个概率分布的均值。而 $\frac{0.53^2}{0.53^2 + 0.1^2}$ 这个东西就叫做卡尔曼增益！

得到的波动结果：（待可视化）

[\[易懂\]如何理解那个把嫦娥送上天的卡尔曼滤波算法 Kalman filter? - 知乎 \(zhihu.com\)](#)
以及等等的原理

四、处理的评估&教练的期望

简介：为了分析 the flow of play 转变的指标，我们使用 BOCD: Bayesian Online Changepoint Detection 处理 30 场比赛（除去 1310）的 N，得到所有 N 的显著断点具体信息，并对这些断点的产生情况进行分析，最后分析出 5 项最容易产生 the flow of play 转变的指标。

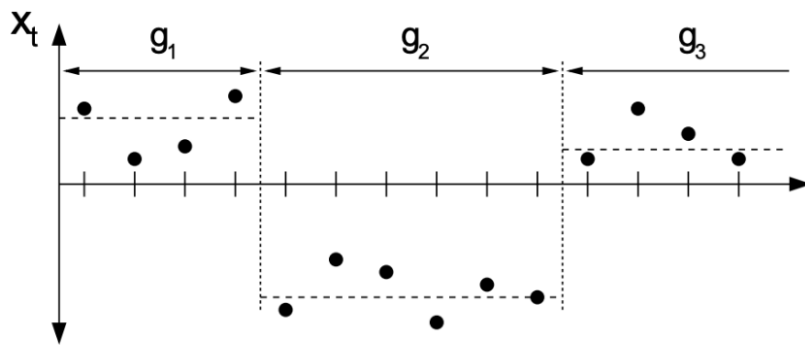
注意到这里选择 N，是因为为了取消具体得分失分对选手的影响。选手得分失分会在一定程度上一般化的影响选手，而单纯的单次得分失分，在高等级的比赛中，不会对选手的心理造成显著影响。如何得分失分才是我们所关注的。(N 为上预测高斯分布的均值)

● 贝叶斯变点检测

解释：<https://zxh.io/posts/zh/2020-08-22-bocd/>

给定一个数据序列，在某个时间点，数据的某个（或某些）参数可能由于系统性因素（而非偶然性因素）而突然发生变化，那么这个时间点被称为**变点** (hmfs1jutrsy)。 **变点检测** (hmfs1jutrsyijyhyts) 就是要估计出变点的位置。

假设有一个观测序列 $x_1, x_2, \dots, x_T \in \mathbb{R}^d$ ，我们用 $x_{a:b}$ 来表示 x_a, x_{a+1}, \dots, x_b 。 $x_{1:T}$ 可以被划分成一些不重叠的 **product partitions**，即每个 partition 中的数据点都是 i.i.d（独立同分布）的，都服从参数为 θ_p 的分布 $p(x_t | \theta_p)$ ，每个 partition 的参数 $\theta_p \sim p(\phi)$ 也是 i.i.d 的：



而 BOCD 会估计当前时刻的 run length（距离上一个变点已经过了多少时刻）的后验分布 $p(r_t | x_{1:t})$ ， t 时刻的 run length 为 r_t 。显然 r_t 只可能有两种情况：

$$x = \begin{cases} 0 & t \text{ 时刻是变点} \\ r_{t-1} + 1 & t \text{ 时刻不是变点} \end{cases}$$

最后一步能成立依赖于以下两个假设：

1. 前面提到的假设 1，即只用当前 partition 的数据来预测 x_t 的分布：

$$p(x_t | r_t, r_{t-1}, x_{1:t-1}) = p(x_t | r_{t-1}, x_{1:t-1}^{(r)}) \quad (\text{假设 1})$$

因为 r_t 只依赖于 r_{t-1} （假设 2），所以这里 r_t 也可以省掉。

2. r_t 只依赖于 r_{t-1} 。即给定 r_{t-1} ， r_t 对其他所有变量都条件独立：

$$p(r_t | r_{t-1}, x_{1:t-1}) = p(r_t | r_{t-1}) \quad (\text{假设 2})$$

（下图第一行是贝叶斯公式

那么我们就需要计算 run length 的分布 $p(r_t | x_{1:t})$ ：

$$p(r_t | x_{1:t}) = \frac{p(r_t, x_{1:t})}{p(x_{1:t})}$$

那么我们需要算联合概率 $p(r_t, x_{1:t})$ ：

$$\begin{aligned} p(r_t, x_{1:t}) &= \sum_{r_{t-1}} p(r_t, r_{t-1}, x_{1:t}) && (\text{边缘概率公式}) \\ &= \sum_{r_{t-1}} p(r_t, r_{t-1}, x_{1:t-1}, x_t) \\ &= \sum_{r_{t-1}} p(r_t, x_t | r_{t-1}, x_{1:t-1}) p(r_{t-1}, x_{1:t-1}) && (\text{链式法则}) \\ &= \sum_{r_{t-1}} p(x_t | r_t, r_{t-1}, x_{1:t-1}) p(r_t | r_{t-1}, x_{1:t-1}) p(r_{t-1}, x_{1:t-1}) && (\text{链式法则}) \\ &= \sum_{r_{t-1}} p(x_t | r_{t-1}, x_{1:t-1}^{(r)}) p(r_t | r_{t-1}) p(r_{t-1}, x_{1:t-1}) && (\text{假设 1, 2}) \end{aligned} \quad (\text{放头尾即可})$$

由此方程得知应使用递推求解，在这里我们对**比赛数据建模**，游戏开始

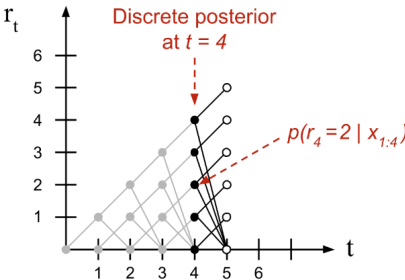
的时刻一定是个变点。那么显然有递推初始值 $p(r_0 = 0) = 1$

其中：

$$p(r_t, x_{1:t}) = \sum_{r_{t-1}} \underbrace{p(x_t | r_{t-1}, x_{1:t-1}^{(r)})}_{\text{UPM}} \underbrace{p(r_t | r_{t-1})}_{\text{变点先验}} \underbrace{p(r_{t-1}, x_{1:t-1})}_{\text{已求得的信息}}$$

Zsijwrs1 Uwtgfgmxyzh R tijq (ZUR)

这张图说明了这个递推流程：



变点先验

贝叶斯方法的特点就是能利用先验信息，所以我们可以把我们对变点的先验估计通过 hazard function 塞进模型里。

我们假设：

$$p(r_t | r_{t-1}) = \begin{cases} H(r_{t-1} + 1) & \text{if } r_t = 0 \\ 1 - H(r_{t-1} + 1) & \text{if } r_t = r_{t-1} + 1 \\ 0 & \text{otherwise} \end{cases}$$

$H(\tau)$ 是 hazard function，描述的是个体在 τ 时刻死亡的概率（...）。当然‘死亡’这个词可以换成别的什么特殊事件，比如在这里 $H(\tau)$ 描述的就是在 run length = τ 时（在这之前都没出现变点）出现变点的概率。

Survival Function

设 $T \geq 0$ 是一个表示当前 run length 长度的随机变量。 $S(\tau)$ 是 survival function，表示 run length 超过 τ 的概率：

$$S(\tau) = P(T \geq \tau) = 1 - F(\tau) = \sum_{\tau'=\tau}^{\infty} f(\tau')$$

其中 $F(\tau) = P(T < \tau)$ 。 $f(\tau)$ 表示当前 run length = τ 的概率，是一个概率密度函数。可以看到有：

$$f(\tau) = F'(\tau) = -\frac{dS(\tau)}{d\tau}$$

Hazard Function

那么由 hazard function 的定义：

$$\begin{aligned} H(\tau) &= \lim_{\Delta\tau \rightarrow 0} \frac{p(\tau \leq T < \tau + \Delta\tau | T \geq \tau)}{\Delta\tau} \\ &= \lim_{\Delta\tau \rightarrow 0} \frac{p(\tau \leq T < \tau + \Delta\tau)}{p(T \geq \tau)\Delta\tau} \\ &= \frac{f(\tau)}{S(\tau)} \end{aligned}$$

这里 $f(\tau)$ 是我们自己定的一个先验。

当 $f(\tau)$ 正好被定为一个指数分布（或几何分布（离散））时：

$$f(\tau) = \lambda e^{-\lambda\tau}$$

$$F(\tau) = 1 - e^{-\lambda\tau}$$

$$S(\tau) = 1 - F(\tau) = e^{-\lambda\tau}$$

此时 $H(\tau) = \frac{f(\tau)}{S(\tau)} = \lambda$ 是一个常数（但论文里面写的是 $\frac{1}{\lambda}$ ，我也不知道我哪里推错了...）。

UPM

好的那么现在 $p(x_{t+1} | r_t, x_t^{(r)})$ 就是最后一个要算的东西了，我们把它写成边缘分布：

$$p(x_{t+1} | r_t, x_t^{(r)}) = \int \underbrace{p(x_{t+1} | \eta)}_{\text{EF model}} \underbrace{p(\eta_t^{(r)} = \eta | r_t, x_t^{(r)})}_{\text{EF posterior}} d\eta$$

$\eta_t^{(r)}$ 表示在 t 时刻 run length = r 时的超参数，第一项的意义是 x 服从参数为 η 的**指数族分布 (Exponential Family)**，然后对这个指数族分布建模。第二项是 η 的后验。

η 的后验并不好计算，而且这里还要对每个 η 求积分，也不好算。所以我们往往希望 η 的先验跟它的似然**共轭 (conjugate)**，使得先验与后验的形式相同，这样就能简化计算。

指数族分布可以写为统一的形式：

$$\begin{aligned} p(x | \eta) &= h(x) \exp(\eta^\top U(x) - A(\eta)) \\ &= \frac{1}{\exp(A(\eta))} h(x) \exp(\eta^\top U(x)) \\ &= g(\eta) h(x) \exp(\eta^\top U(x)) \end{aligned}$$

其中， $h(x)$ 是 underlying measure (没找到啥合适的翻译)； $U(x)$ 是充分统计量 (sufficient statistic)，包含样本集合的所有信息，如高斯分布中的均值 μ 和方差 σ ； $g(\eta) = \frac{1}{\exp(A(\eta))}$ 是正则函数 (normalizer)； $A(\eta)$ 是对数正则函数 (log normalizer)，用于保证：

$$\frac{1}{\exp(A(\eta))} \int h(x) \exp(\eta^\top U(x)) dx = 1$$

$A(\eta)$ 叫 log normalizer 这个名字是因为由上式可以推出：

$$A(\eta) = \log \int h(x) \exp(\eta^\top U(x)) dx$$

跟似然 $p(x | \eta)$ 共轭的 η 的先验为：

$$p(\eta | \chi, \nu) = f(\chi, \nu) g(\eta)^\nu \exp(\eta^\top \chi)$$

其中 χ, ν 是超参数, $f(\chi, \nu)$ 取决于指数族分布的具体形式。

由贝叶斯公式, 后验正比于似然 \times 先验, 所以:

$$\begin{aligned}
 p(\eta | X, \chi, \nu) &\propto p(X | \eta) p(\eta | \chi, \nu) \\
 &\propto \underbrace{\left(\left(\prod_{i=1}^N h(x_n) \right) g(\eta)^N \exp(\eta^\top \sum_{n=1}^N u(x_n)) \right)}_{\text{似然}} \underbrace{\left(f(\chi, \nu) g(\eta)^\nu \exp(\eta^\top \chi) \right)}_{\text{先验}} \\
 &\propto \left(\prod_{i=1}^N h(x_n) \right) f(\chi, \nu) g(\eta)^{N+\nu} \exp \left(\eta^\top \sum_{n=1}^N u(x_n) + \eta^\top \chi \right) \\
 &\propto g(\eta)^{N+\nu} \exp \left(\eta^\top \sum_{n=1}^N u(x_n) + \eta^\top \chi \right)
 \end{aligned}$$

最后一步是因为 $\left(\prod_{i=1}^N h(x_n) \right) f(\chi, \nu)$ 是跟 η 无关的, 所以可以去掉。

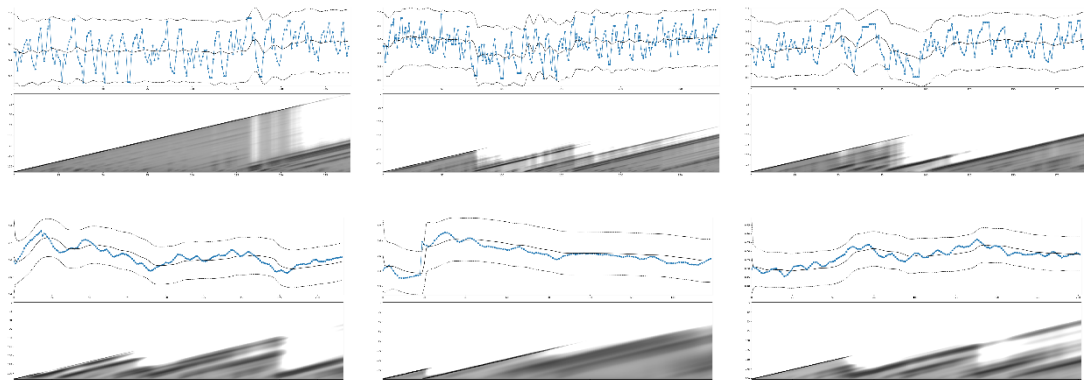
可以看到后验 $p(\eta | X, \chi, \nu)$ 跟先验 $p(\eta | \chi, \nu)$ 的形式是一样的, 只是超参数上有区别:

$$\begin{aligned}
 \nu' &= \nu_{\text{prior}} + N \\
 \chi' &= \chi_{\text{prior}} + \sum_{n=1}^N u(x_n)
 \end{aligned}$$

所以如果似然是指数族分布, 那么在先验的超参数上加一个项就可以得到后验的超参数, 是种很简便的方法。

(整理压缩。。。)

执行程序进行 GTHI, 所得示意图如下



节选半决赛&四分之一决赛的, 顺序为从左上到右下 1501 1502 1503 1504 1601 1602

对所有 Q 的显著断点具体信息结果进行处理, 共得到 129 个断点。统计每个断点的主要因素, 去除发球方因素, 其中选择 5 项频数最高的, 也即最容易产生 the flow of play 转变的指标, 并进行解释, 如下

列表:

	频率	解释
对方 breakpoint/发球局胜负	31/129	
接发球	27/129	
失误	23/129	
体力比值	19/129	
发球直接得分	18/129	

序列 a 预测序列 b

系统定义为场上局势
初始阶段系统为空闲状态，等待接收序列 a 的事件。如果序列 b 发生事件则该事件判断为 miss。
当系统接收到序列 a 的事件后，立刻开始接收序列 b 的事件。持续时间为一段接受窗口期 T_{accept} 。
如果系统在接受窗口期内，没有接收到序列 b 的事件。那么序列 a 的事件判断为 miss。系统返回初始/空闲状态。
如果系统在接受窗口期内接受到序列 a 的事件，则该事件判断为 interrupting，接受窗口期不变。
如果系统在接受窗口期内接受到 b 的事件则该对事件判断为 success，系统进入一段真空期。
真空期的时长不是固定值，它随接受窗口期用时的增加而减小，但最小值大于接收窗口期
在真空期内接收到任何 ab 事件，都将该事件判断为 interrupting。
真空期结束后，系统返回空闲状态

按时间顺序处理。待所有事件接受完成为止。即可得到每件事件的预测情况进而分析出预测模型的准确性。

处理得到结果，：

总事件数	Miss	Interrupt	Success
204	33	23	148
267	46	73	148

其中 interrupt 为数据集固有误差，应排除于样本外

总体概率：可视化

将预测效果借用混淆矩阵的概念进行量化分析：

精确率： $148/(148+33)=81.77\%$

召回率： $148/(148+46)=76.29\%$

F1 分数：0.78935

精确率 (Precision)：模型预测为正类别的样本中有多少是真正的正类别。

召回率 (Recall)：实际为正类别的样本中，有多少被模型正确预测为正类别。

F1 分数：精确率和召回率的调和平均数，综合考虑了查准率和查全率。

2. 精确率 (Precision)

精确率表示模型预测为正类别的样本中有多少是真正的正类别，计算方式为：

$$Precision = \frac{TP}{TP+FP}$$

3. True Positive Rate (TPR) - 召回率/灵敏度

召回率，也称为 True Positive Rate (TPR) 或灵敏度，是指在所有实际为正类别的样本中，模型能够正确预测为正类别的比例。其计算方式为：

$$Recall(TPR) = \frac{TP}{TP+FN}$$

高召回率意味着模型能够有效地捕捉到实际为正类别的样本，是在医疗诊断等领域非常重要的指标。

7. F1 分数(F1-score)

F1 分数是精确率和召回率的调和平均数，它综合了两者的性能，计算方式为：

$$F1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall}$$

F1 分数的取值范围是 [0, 1]，越接近 1 表示模型的性能越好，同时考虑到了模型在查准率和查全率之间的平衡。F1 分数对于二元分类问题非常有用，特别是当我们希望在精确率和召回率之间取得平衡时。高 F1 分数意味着模型在查准率和查全率之间取得了良好的平衡。

这些指标提供了多方面的信息，帮助我们全面了解模型的性能。准确率衡量了模型在所有样本上的整体表现，精确率关注模型在预测为正类别的样本上的准确性，召回率关注模型在实际为正类别的样本上的覆盖程度，而 F1 分数平衡了精确率和召回率。

混淆矩阵和这些性能指标共同提供了对分类模型性能全面的理解，并帮助我们评估模型的优缺点，进而进行必要的改进。

通用性分析

- 女子:q（局数无影响）
- 球场表面:q 变化
- 乒乓球:更强得分潮

以下部分语文建模

给出比赛建议

战术建议涉及到控制 V（面对失分的积极心态）、Q（技术提高）、以及战术针对

修正建议

建议增加维度：天气、风速，等等

六、备忘录

报告结构建议

- 引言：**简要介绍研究的背景、目的和重要性。
- 理论基础：**讨论势头在体育比赛中的角色，以及先前研究的相关理论。
- 方法论：**
 - 数据收集：**描述使用的数据集及其来源。
 - 特征工程：**解释选择的特征及其对预测势头转换的重要性。
 - 模型选择与开发：**概述所选模型的理由，包括模型的类型、训练过程和参数调整。
- 结果分析：**
 - 势头分析：**展示模型如何捕捉比赛中的势头变化。
 - 势头的影响：**讨论势头变化对比赛结果的潜在影响。
 - 模型泛化能力：**报告模型在不同比赛、运动和条件下的测试结果。
- 讨论：**分析模型的局限性，提出未来研究方向。
- 结论：**总结研究发现，并强调其对教练和选手准备的实际意义。
- 附录：**包括数据集描述、代码实现和额外的图表或表格。

模型总结与教练建议备忘录

这一部分提供给教练的建议，基于上述分析的结果。

建议摘要

1. **势头的角色**：我们的分析显示，势头确实在比赛中扮演了一个重要角色。

势头的增加与选手的表现提升相关联，而势头的减少可能预示着比赛流向的转变。

2. **势头变化的指标**：某些关键指标，如连胜点、重要得分时刻的表现，可以作为势头变化的前兆。

3. **准备选手**：

4. 加强心理训练，特别是在面对势头不利时，帮助选手保持冷静和专注。

5. 分析对手的比赛数据，识别可能引发势头转换的模式或弱点。

6. 在训练中模拟势头转换的情况，提高选手的适应能力。

基于模型结果，我们可以向教练提供以下建议：

- 关注特定的比赛转折点指标，如得分差、发球权利用等，这些因素可能预示着比赛流向的即将变化。
- 训练选手识别并利用比赛中的转折点，比如在连续得分或面临重要得分时的心理准备和战术调整。
- 分析不同对手的比赛数据，以定制化的策略应对不同的比赛情况。

Conclusion

References（待整理）

[1]: Wikipedia contributors. "Wordle." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 22 Jan. 2024. Web. 23 Jan. 2024.

[2]: Bonthron, Michael. "Rank One Approximation as a Strategy for Wordle." *arXiv preprint arXiv:2204.06324* (2022). Print.

Gf~jxfs Tsqsj Hmfsljutr~~sy~~ Ijyjhyts3W~fs Uwjsxhtyy Fifr x fsi

If{ni OH3R fhPf~}fw} n{ 755<3