Problem Chosen	2023	Team Control Number
C	MCM/ICM	2322645
	Summary Sheet	

如果世界词典周三的世界词典单词是“怪诞”，世界词典猜词者将猜出多少个世界词典单词?

摘要：随着世界越来越受欢迎，人们急切地在 Twitter 上报告他们每天数以万计的结果。关于这些数据，有三个非常自然的问题出现了:(1)我们可以使用这些数据来预测世界词典中给定目标单词的难度吗?(2)我们是否可以使用这些数据来预测未来的世界词典玩家报告趋势?(3)给定目标词的难度如何影响玩家报告和结果?在我们的论文中，我们开发了一个综合的贝叶斯模型，由三个子模型组成，这些子模型预测猜测次数的分布，Twitter 上报告结果的数量以及报告在困难模式下玩的玩家的数量。

最初，我们将单词分解为与相关难度特征相关的可量化特征。最值得注意的是，我们制定了一种新的世界特定熵度量，我们称之为子集熵，它有效地量化了典型玩家在初始猜测后透露的平均信息量。我们还开发了一种方法来表示玩家尝试的分布，从而表示观察到的单词难度，仅使用两个值α， β对应于 Beta 分布的累积质量函数。我们使用初步的拉索回归来分离最相关的单词难度预测因子，然后将其用于贝叶斯模型。

对于给定的日期和单词，我们的贝叶斯模型预测了一个单词的报告难度，报告的玩家数量，以及报告在困难模式下玩游戏的玩家数量。为了完成这三个任务，它由三个子模型组成，这些子模型在给定数据的情况下是条件独立的，这使得使用马尔可夫链蒙特卡罗(MCMC)从其后验中进行采样变得高效。

我们发现拥有更多独特字母的单词，在英语中的使用频率，在所有猜测中显示的黄色方块的平均数量以及子集熵都让玩家更容易猜测一个单词。我们还发现，较高的单词难度会减少玩家报告的次数。在《时代》随机选词的假设下，这可以解释为一种因果效应。

我们的模型能够对新数据进行结果预测，对旧数据进行回溯。我们的模型给出了 95%的预测区间，即 20238 到 27876 名玩家将在 2023 年 3 月 1 日报告“怪异”的结果，并且它将处于难度的第 50 百分位。最值得注意的是，我们的模型不仅提供了这样简单的点估计和预测区间，还提供了完整的后验分布。

关键词:熵，拉索回归，MCMC，贝叶斯方法，因果推理

目录

1 介绍

《世界大战》是一款基于语言的游戏，目前由《纽约时报》所有，在 2022 年初引起了轰动。这款游戏的目标很容易理解:在每天的开始有一个 5 个字母的目标单词，玩家必须猜测。玩家有六次尝试，并尝试在尽可能少的尝试中猜出这个单词，每次都要使用一个有效的英语单词。

如果取 5 个字母的所有可能序列，有 11,881,376 个可能的 5 个字母单词。即使将其限制在英语词典中的单词和今天常用的单词中，也只能分别减少到 12,000 和 4,000 个单词[2]。对于一个人来说，在六次尝试中随机猜出一个目标单词在统计上几乎是不可能的。然而，这就是贴图颜色反馈发挥作用的地方。对于给定的猜词，对于每个字母，如果对应的字母在目标单词中并且在正确的位置，则返回一个绿色的贴图，如果对应的字母在真实单词中但在错误的位置，则返回一个黄色的贴图，如果这两个都不为真，则返回一个灰色的贴图。有了这些信息，大多数玩家都能在六次尝试中从数千种可能性中猜出这个单词。

这款游戏吸引了数百万人的关注，人们在社交媒体上分享他们的猜词结果，并评论某些单词的难度。由于这种趋势，一个名为“@WordleStats”的 Twitter 账户应运而生，这个机器人可以统计所有发布的世界大战次数以及每天的挑战次数分布。通过这些数据，我们可以发现关于《世界大战》玩家行为的大量信息。特别是，在本文中，我们开发了一个模型，该模型利用 twitter 报告的趋势和由此产生的目标词的推断难度，从这些数据中可以预测未来的世界统计数据。

2 数据

2.1 数据清理

通过引用@WordleStats Twitter 机器人的 Twitter 帖子，我们修复了提供数据中出现的几个错误。为了完全透明，这些错误都记录在下面。

第 239 天:硬模式 3249→9249

第 314 天:tash→trash

第 500 天:硬模式 3667→2667

第 525 天:clean→clean

第 529 天:报告玩家 2569→25569

第 540 天:na¨ıve→naive(¨在世界中不是一个字母)

第 545 天:rprobe→probe

由于不同类别报告的百分比是四舍五入的，并不一定等于 100，所以我们将每一行的百分比除以它们的总和，得到比例。由于我们的贝叶斯模型预测了每个类别的球员数量，我们通过对每行应用以下方法将这些比例转换为计数:

1.将比例乘以当天的报告数量，得到带有十进制值的“计数”

2.四舍五入

3.将 1 加回计数，按照从舍入最多的计数到舍入最少的计数的顺序，直到总数再次与当天的报告数量相匹配。

此方法给出与给定百分比相对应的计数，是整数，其总和是当天的报告数。

2.2 Wordbank

为了对 5 个字母的单词及其属性建模，我们依赖于 Donald Knuth[6]创建的斯坦福 GraphBase (SGB) 5757 个 5 个字母的单词库，它提供了一个很好的近似，玩家可以猜测和期望作为目标的单词集。然后用几种不同的方式来使用这个词库。首先，它用于构建订单频率表和字母频率表。字母频率表告诉我们每个字母在 5 个字母的单词中出现的频率，并符合我们的预期。S 和 E 是最常见的字母，其次是 A 和 o。订单频率表然后显示给定某个字母在单词中，该字母在每个位置的时间比例(例如给定 A 在单词中，它在 24%的时间内是第四个字母)。

正如后面将详细介绍的那样，我们还在许多特定于单词的计算中使用这个表，即当 t 是实际的目标单词时，为给定单词 t 计算从 SGB 词库中随机均匀选择的任何猜测所返回的绿色、黄色和彩色瓷砖的平均值。此外，对于任何单词 g，当从 SGB 单词库中随机选择目标单词 t 并猜测 g 时，我们计算返回的平均颜色数，这提供了一个有点幼稚但合理的指标来评估玩家会使用的猜测单词。使用这个猜测词度量，我们编制了一个具有最高对应平均值的 30 个词的列表，我们将其作为一组常见的猜测词(见图 1)。这个列表将在后面的子集熵的计算中使用。

3 Word &难度

许多因素导致了与给定单词相关的困难。例如，“zingy”在直觉上似乎很难，原因有很多——它有不常见的字母(“z”和“y”)，只有一个“规范的”元音，而且在英语中通常是一个不常用的单词。另一方面，像“onion”这样的单词似乎也很难，尽管它的所有字母都相当常见，而且它在日常口语中的使用率要高得多。人们认为“洋葱”很难的原因是它会重复出现字母“o”和“n”，一旦人们已经发现了它的一个位置，他们就不太可能再猜一次。因此，给定一个单词，我们的第一个任务是列出并量化这些特征，以便在稍后评估单词难度时，我们可以简单地考虑与这些相关特征对应的值向量。

Word	Avg Colors
arose	2.12176481
raise	2.0656592
arise	2.0656592
aloes	2.0654855
stoae	2.06531179
laser	2.06461699
earls	2.06461699
reals	2.06461699
tears	2.06444329
rates	2.06444329
stare	2.06444329
aster	2.06444329
tares	2.06444329
snare	2.01233281
earns	2.01233281
nears	2.01233281
saner	2.01233281
nares	2.01233281
aisle	2.00937989
least	2.00816397
tales	2.00816397
steal	2.00816397
slate	2.00816397
stale	2.00816397
teals	2.00816397
tesla	2.00816397
taels	2.00816397
stela	2.00816397
reads	1.99426785
dares	1.99426785

	Frequency		Frequency
s	0.105367	y	0.030780
e	0.104534	m	0.029286
a	0.081570	h	0.028279
o	0.066528	b	0.024839
r	0.066354	g	0.023589
i	0.055307	k	0.020705
l	0.055098	f	0.019489
t	0.055063	w	0.017544
n	0.044641	v	0.011047
d	0.041028	x	0.004829
u	0.037832	z	0.004690
c	0.033490	j	0.003092
p	0.033177	q	0.001841

图 1:左:30 个基于重叠的最常见单词。右图:SGB 字库中字母的频率

3.1 元音

在所有基于猜字母的游戏中(游戏邦注:除了《世界语言》，比如《hangman》)，一个典型的策略是利用单词中元音字母出现的频率更高的字母。在世界语言中，特定元音的出现往往比非元音的出现更容易被发现，这导致了一个合理的假设，即元音较多的单词通常更容易被猜出。因此，计算和考虑每个单词的一个特征是它包含的元音数量(不包括“y”，因为这是一个传统上不常见的字母，因此与上面给出的为什么我们首先考虑元音的推理不一致)。

3.2 使用

人们对常用的单词更熟悉，因此更容易从给定的线索中猜测出来，这是有道理的。使用 python 中的 wordfreq 库[5]，可以很容易地为考虑的每个单词返回这个值。

3.3 Green & Yellow Tiles

目标单词的另一个理想特征是，在任何给定的猜测之后，世界语言都很有可能返回大量的绿色和黄色方块。一旦出现这种情况，玩家就会得到非常直接的提示，他们可以立即采取行动。因此，对于任何给定的单词，我们计算返回的绿色、黄色和彩色(即非灰色)瓷砖的平均数量，考虑到给定的单词作为目标，并假设猜测是从 SGB 单词库中均匀随机抽取的。

3.4 唯一字母

当《世界语言返回一个字母的颜色时，它并不会告诉玩家这个字母出现了多少次。因此，虽然最初可能更容易获得含有重复字母的单词的绿色贴图，但玩家在不同位置再次猜出相同字母的可能性也更小。因此，考虑一个单词中唯一字母的数量对猜测难度的影响似乎是合理的，因此这个值是为每个考虑的单词计算的。

图 2:如果第一次猜测是正确的，重复的字母不会出现，它没有表明有第二个 k。

3.5 熵

信息论的核心思想之一是熵，它是一种量化给定事件所传达的信息量的度量。在世界游戏中，我们可以将事件视为玩家移动的特定选择，即猜测单词以及世界游戏返回的相应位置颜色。很明显，这个事件给了我们一些关于目标单词的信息，一个合理的问题是有多少信息?一旦这样的值被正式量化，人们就可以认为“最佳”猜测词是那些平均产生更高信息量的词(在所有可能的 5 个字母的目标词的均匀分布上)。相反，目标词更难以猜测，通过一些有效的文本编码，哪些包含更多的信息，或者导致平均获得的信息量更少(在可能的猜测词的某些分布上)。这两个想法通过我们下面给出的两个熵公式形式化，第一个是标准香农熵函数的一个更典型的应用，第二个是我们为世界本身开发的一个更新颖的熵概念。

3.5.1 位置熵

熵的最初用例来自编码，更频繁的符号和常见的排列需要更少的比特来传输。我们的位置熵采用了这种方法，如果一个单词的字母和字母排列不太常见/不寻常，它就会包含更多的信息。使用来自 SGB 词库的字

母频率表和顺序频率表，对于给定的字母ϕ，我们计算 Pi(ϕ)作为单词在其 ilth ocation 中具有字母ϕ的概率。然后可以将这些概率代入标准的香农熵函数 H[4]，其中 X 表示一个 5 个字母的单词，X 对应 i 它的 1 个字母，如下 th 所示:

$$H(X) := \sum_{i=1}^{5} -P_i(X_i)\log_2(P_i(X_i))$$

请注意，在这个度量下，具有更多不寻常字母位置的单词，因此可以合理地认为更困难，将具有更高的值。

3.5.2 子集熵

我们的第二个熵公式是子集熵，这是我们开发的一种新颖的受 word 启发的度量，在给定的单词上，量化从某个分布中选择一个世界猜测后揭示的平均信息量。这个指标的动机是，在一个给定的目标词 t 上，一个选择的猜测词 g，信息是通过输出颜色和它们对应的位置获得的，这使得人们可以取消其他候选目标词 t′。例如，如果考虑的目标词是面包，而猜测的词是面包屑，那么世界的相应输出将允许我们排除吐司和螃蟹等词作为可能的目标词，但不允许我们排除 arbor。图 3 给出了一个说明这一想法和我们的符号的例子。

在我们详细介绍子集熵之前，首先我们将 ft(g)定义为在猜出单词 g 并且 t 是目标单词后候选答案池缩小的因子。更正式地说，考虑到图 3 中的表示法，我们有:

$$f_t(g) = \frac{n}{n_1}$$

其中 n 为原始候选答案池的大小，nis1 为结果候选答案池的大小。举个例子，当给定的目标单词为 t，导致可能的答案池从 4000 个单词缩减到 1000 个单词时，猜测 g 的值为 f(gt) = 4。请注意，如果 1 是一个比给定 2 目标词是 t 更好的猜测，我们有 f(g) > f(tg)1，而如果一 t 个目 2 标词比使用猜测 g 更容易猜 1 测，那么 2 f(g) > f(g)。t1 t2

在典型的熵风格中，我们对计算出来的这个因子取以 2 为底的对数，并定义:

$$I_t(g) = \log_2(f_t(g))$$

我们使用这个值来对应我们对目标 t 上猜测 g 事件所传达的信息的概念。直接选择这个值而不是 ft(g)的动机是，当单词 t 是目标时，子集熵的目标是量化第一次猜测之后的平均信息，因此需要对一些量化信息取平均值。由于有效 t 地测量了信息的乘法因子(而不是加法)，因此取几何平均值而不是算术平均值会更合适。然而，由于某些项的对数值的平均值正好对应于这些项的几何平均值的对数，我们可以有效地将 I(tg)在猜测 D 的某些分布上的期望作为这些因素的几何平均值的度量。有了这个理由，我们有以下子集熵的公式:

$$e(t) = \mathbb{E}_{g\sim\mathcal{D}}[I_t(g)]$$

其中 D 是可能猜测词的某个分布。

在我们模型的实际实现中，我们认为 D 是 30 个最常见的猜测词列表上的均匀分布(见图 1)。特别是，该模型假设玩家根据平均颜色度量猜测 30 个最佳猜测词中的一个。我们认为这是合理的，因为许多被猜出的常用词或与其相似的词，比如“slate”，似乎都在这个列表上。

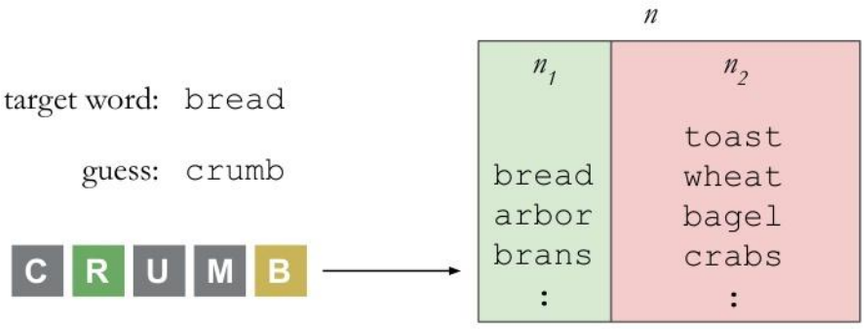

图 3:表示子集熵方法的图表

对于目标单词“bread”，系统会做出一个猜测，世界会告诉玩家 R 一定在第二个位置，而 B 一定在单词的某个位置，但不是最后一个位置。此外，《世界》告诉玩家 C、U 和 M 不在目标单词中。这个信息定义了一个与它兼容的 n 个单词的子集 1 。

3.6 单词难度的表示

为了帮助建模和解释，理想情况下，7 个向量数据(玩家在 1 次尝试，2 次尝试中获得单词)…，失败)，它有效地代表了观察到的一个单词的难度，应该用更少的数字封装。我们可以将这七个类别视为一个离散的空间，将观察到的比例视为它们之上的离散概率质量函数。通过假设 1 < 2 <…的排序。< X，然后我们取对应于概率质量函数的累积质量函数。通过映射(i 尝试)7→i/7 和 X 7→1 将域嵌入区间[0,1]后，结果发现，在[0,1]上的双参数连续分布 Beta 分布具有一个可以紧密拟合嵌入的累积质量函数的累积分布函数(见图 4)，然后我们用 Beta 分布的累积分布对读者尝试进行建模，其概率密度函数为:

$$f(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1},$$

其中，B(α， β)为 Beta 函数。为了找出最适合嵌入累积质量函数的α， β的确切值，我们使用 scipy[7]中实现的非线性最小二乘。通过观察上面的图以及从 Beta 分布的性质，我们可以总结出α和β的几个性质:

在其他条件相同的情况下，由于α↑，这个词被认为更困难。当其他条件相同时，当β↑条件相同时，该单词被认为更容易。α+β表示期望值周围分布的集中程度，值越高越集中。比值α+αβ 给出了 Beta 分布的期望值，这是一种表示一个单词的难度，数值越高对应难度越高。

由于这些属性，尤其是最后一个，α和β合在一起是一个单词难度的有效而简单的表示。图 5 显示了根据之前所有单词的观察报告分布计算出的α/(α + β)。

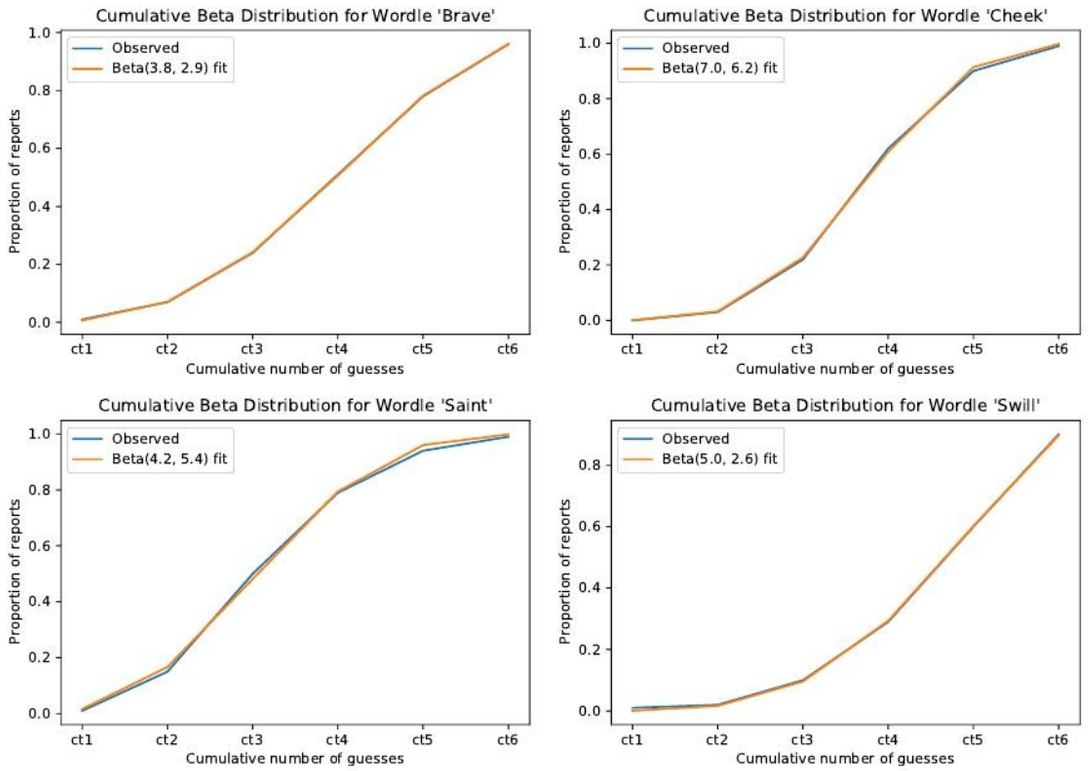

图 4:Beta 分布拟合各种难度的 Wordle 单词的累计猜测比例

4 建模方法

4.1 套索回归

一旦我们有了所有可以从单词本身中提取的变量，我们必须使用选择其中的一个子集来包含在贝叶斯模型中，因为 MCMC 所花费的时间随着模型中变量的数量而显着增加，我们必须保持运行时间合理。为了选择更重要的预测因子，我们使用套索回归。套索回归惩罚大系数，这反过来迫使不太重要的变量的系数为零。套索回归的成本函数如下:

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{P} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{P} |\beta_j|,$$

其中，xij is 为第 i 个数据点的 JTH 预测器的值，j β为 JTH 系数。λ是一个调优参数，实验后我们将其设置为 0.1 以获得所需的参数数(4 个)。从成本函数中可以看出，套索回归力求使真实值与预测值之间的误差最小，同时使所有系数的总和较小。

有了我们的参数，有七个独立的套索回归模型同时运行，一个是关于一次尝试成功的人的比例，一个是关于两次尝试成功的人的比例，以此类推，直到最后一个是关于那些失败的人的比例。出现了 4 个参数 re-

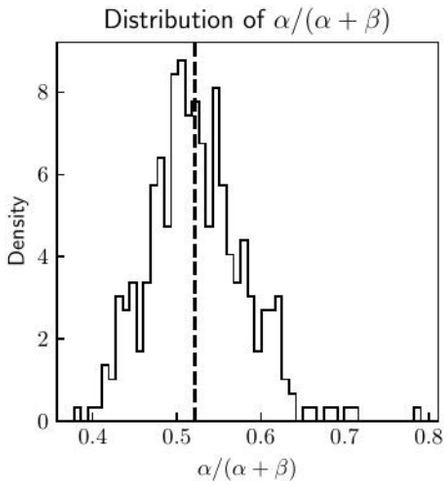

图 5:测量单词难度的α/(α+ β)的观察分布

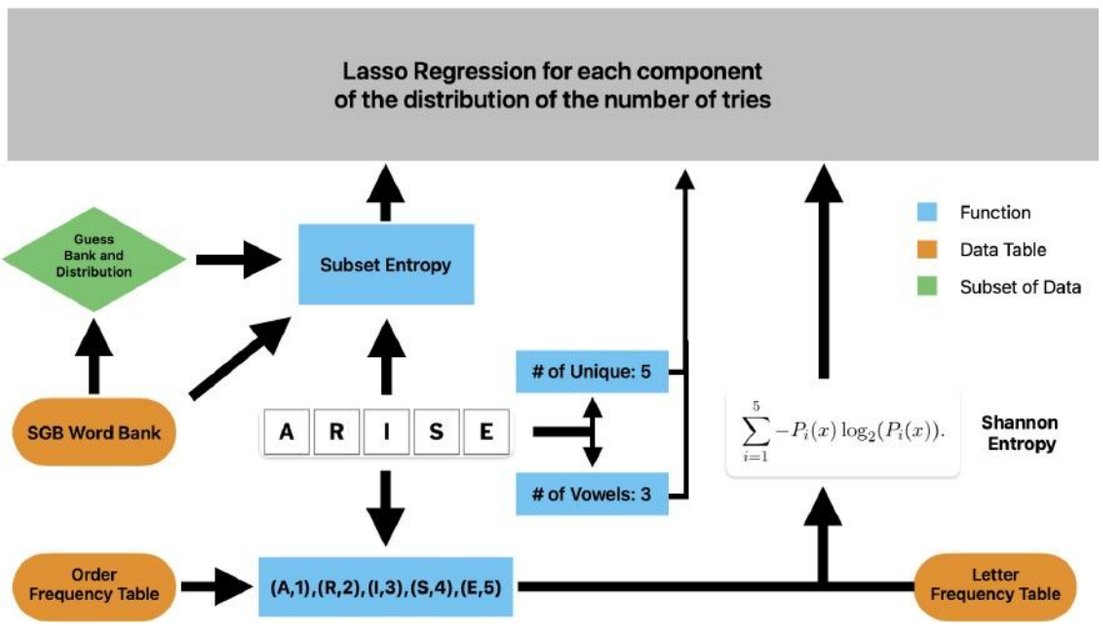

图 6:Lasso 回归的输入

在这些回归中反复使用非零系数:唯一字母的数量，单词的使用频率，显示的黄色方块的平均数量，以及子集熵。

4.2 贝叶斯预测模型

我们使用包含三个条件独立子模型的贝叶斯模型来建模尝试的分布(Try 模型)、报告的数量(reports 模型)和硬模式玩家的数量(或“hardmoders”模型中的“hardmoders”)。该模型如图 7 所示。

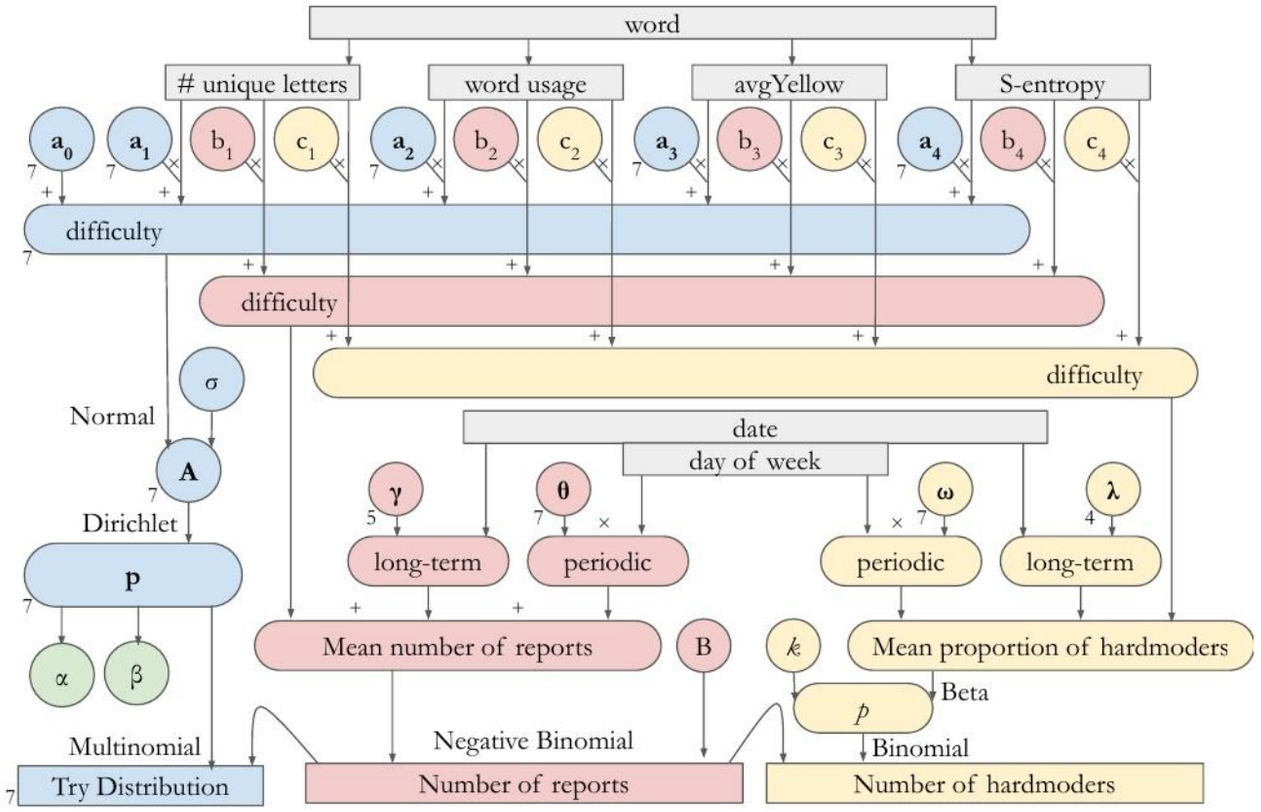

图 7:三种贝叶斯模型的组合图，圆角形状为变量，矩形为观测数据

数字表示向量的维数(其他所有变量和数据都是标量)。黄色为预测硬模比例的模型，红色为预测报告数的模型，蓝色为预测尝试数分布的模型。形状外的数字表示数据和变量在非标量时的维度。在观测数据的条件下，所有三个模型的随机变量都是独立的，这使得它们的后验可以在 MCMC 的三次独立运行中推断出来。注意，来自报告模型的报告数量被输入到 Try 模型和 Hardmoders 模型中。绿色部分是α和β值，它们是对应于一个单词的难度分布 p 的 Beta 分布的参数。

4.2.1 常用组件

三种贝叶斯模型中的每一种都有四个与难度相对应的预测因子。由于难度对尝试分布、报告数量和 hardmoder 数量的影响不一定相同，我们在这些模型中分别做了一个回归:

$$\mathbf{d}_{\text{try}} = \mathbf{a_0} + \mathbf{a_1}x_1 + \mathbf{a_2}x_2 + \mathbf{a_3}x_3 + \mathbf{a_4}x_4 \quad (1)$$

$$d_{\text{reports}} = b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 \quad (2)$$

$$d_{\text{hardmoders}} = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 \quad (3)$$

其中 a、b、c 是要估计的系数，xi 是预测因子。请注意，Try 模型将 dtry 和系数都 i 作为 7 个向量。还要注意的是，Try 模型是唯一一个在这个分量中有截距的模型，因为在其他模型中，截距是由其他项捕获的。每个系数都有一个 Normal(0,20)的先验，它足够宽，不能提供信息。

在报告的数量和硬模型的比例中，都存在周期性的周数效应。在报告和 Hardmoders 模型中，这些模型类似:

$$P_{\text{reports}} = \sum_{i=1}^{7} \theta_i \cdot \mathbf{1}(i\text{th day of week}) \tag{4}$$

$$P_{\text{hardmoders}} = \sum_{i=1}^{7} \omega_i \cdot \mathbf{1}(i\text{th day of week}) \tag{5}$$

θ和ω系数被建模为具有正态(0,1)先验分布。

4.2.2 Try 模型

尝试分布的模型(Try 模型)基本上是 Dirichlet-Multinomial 模型。对于给定日期的给定单词，我们对向量(t1, t2，…)建模。， t7)，其中 ti, i < 7 是在第 i 次尝试中获得该单词的报告数量，而 7 这是报告失败的数量，来自多项分布:

$$(t_1, t_2, t_3, t_4, t_5, t_6, t_7) \sim \text{Multinomial}\left(n, 7, (p_1, p_2, p_3, p_4, p_5, p_7)\right),$$

其中 nP 是当天报告的数量，p= (p, p, p1, p2, p3, p4, p57)是一个概率 7 向量(7i=1 p i = 1)，表示一个报告在第一次尝试、第二次尝试，以此类推时报告成功的概率。这个向量本身被建模为来自狄利克雷分布，这是一个离散概率分布的分布:

$$(p_1, p_2, p_3, p_4, p_5, p_7) \sim \text{Dirichlet}(7, (A_1, A_2, A_3, A_4, A_5, A_6, A_7)),$$

式中，A1= (A2, A3, A4, A5, A6, A, A, A7)为狄利克雷分布的浓度参数向量，Ai > 0。这些浓度参数的对数本身是从正态分布中得出的:

$$\ln(A_i) \sim \text{Normal}(d_{\text{try},i}, \sigma),$$

其中 distry,i 由公式 1 定义。σ是一个方差参数，其先验为指数(1)。

4.2.3 报告模型

报告数量的模型(报告模型)基本上是一个过度分散的泊松回归。某一天的报告数量可以清晰地建模为具有一定速率的泊松分布。然而，由于除了泊松误差之外，在观察到的报告数量中存在未被预测者考虑的额外变化，因此我们必须使用泊松分布的过分散版本(即行为类似泊松的分布，但有额外的误差)。当参数化正确时，负二项分布具有此属性。因此，我们将给定单词在给定日期的报告数量建模为:

$$n \sim \text{NegativeBinomial}\left(\exp(L_{\text{reports}} + P_{\text{reports}} + d_{\text{reports}}), B\right).$$

Lreports, Preports，和 reports d 分别对应于报告数量的长期趋势，周期性趋势和难度的影响。周期趋势的定义见式 4。难度的影响定义在式 2 中。B 是描述过分散的参数，并被建模为具有指数(0.01)先验分布，以便允许其较大。

由于长期趋势与 Gamma 分布形状的对应关系，我们将 Lasreports 参数化为一个函数，该函数可以回忆起 Gamma 分布的概率密度函数:

$$L_{\text{reports}} = \gamma_1((T - \gamma_5)^{\gamma_2}) \exp(-(T - \gamma_5)/\gamma_3) + \gamma_4,$$

其中 T 是单词的时间值。T 被缩放，使得数据中的第一个单词的 T = 0，最后一个单词的 T = 1。γ系数 i 根据其在方程中的作用，具有不同的先验分布。γ1、γ2、γ3 均具有指数(1)先验分布，4 γ具有指数(0.01)先验分布，5 γ具有正态(0,1)先验分布。当考虑到 T 的尺度时，这些先验的选择是没有信息的。

4.2.4 Hardmoders 模型

报告在困难模式下玩游戏的人数模型(Hardmoders 模型)基本上是一个β -二项回归。给定某一天的报告数量，硬模的数量用具有一定概率的二项分布来描述。然而，由于在观测到的硬模态数中除了二项误差之外还有额外的变化，而这些变化没有被预测者考虑在内，因此概率本身被建模为来自 Beta 分布。

因此，我们将给定单词在给定日期的硬模 h 数 n 建模为:

$$n_h \sim \text{Binomial}\,(n, p)$$

其中 n 是当天报告的数量，p 是有人报告玩硬模式的概率。P 本身被建模为:

$$p \sim \text{Beta}\,(\eta\kappa, (1-\eta)\kappa)\,,$$

地点:

$$\eta = \text{logit}^{-1}(L_{\text{hardmoders}} + P_{\text{hardmoders}} + d_{\text{hardmoders}})$$

注意，η是这个 Beta 分布的均值，为 E[p] = ηκ/(ηκ +(1 − η)κ) = η。相应地，κ控制着分布的扩散。Lhardmoders、Phardmoders、和 hardmoders 分别对应硬模数量的长期趋势、周期趋势和难度效应。周期趋势的定义见式 5。难度的影响定义在式 3 中。κ有效地控制了β -二项模型的过度分散，并被建模为具有半柯西(0.5)先验分布，该分布具有长尾并充当无信息先验。

然后，我们将 Lashardmoders 参数化为一个可以遵循观察到的形状的函数:

$$L_{\text{hardmoders}} = \lambda_1((T - \lambda_4)^{\lambda_2}) + \lambda_3,$$

其中 T 是单词的时间值。T 被缩放，使得数据中的第一个单词的 T = 0，最后一个单词的 T = 1。λ系数 i 根据其在方程中的作用，具有不同的先验分布。λ1、λ2 和λ3 都具有指数(1)先验分布，λ4 具有正态(0,1)先验分布。当考虑到 T 的尺度时，这些先验的选择是没有信息的。

4.2.5 获取后验

我们不是计算参数的简单点估计，而是在给定数据的情况下获得完整的后验分布。这些是通过使用马尔可夫链蒙特卡罗(MCMC)从变量的联合后验分布中迭代采样得到的。由于给定数据，三个子模型是条件独立的，因此我们分别在三个子模型上运行 MCMC。特别是，我们使用了 No U-Turn Sampler[1]，因为它是在 PyMC[3]中实现的。

5 模型结果

一旦获得三个子模型参数的后验分布，我们就可以从分布中收集信息，并通过模型输入单词和日期来进行预测和追溯，以获得后验预测。

5.1 参数后验的解释

模型参数的后验可以告诉我们一个词的哪些属性影响了一个词的难度，报告的分数的数量，或者硬模者的百分比。在 Try 模型中获得 i 的系数的后验都表明了对七个不同类别的结果分布的影响。较高的唯一字母数量、单词使用频率、显示的黄色方块的平均数量或子集熵都导致单词更容易猜测，这反映在第二次尝试猜出单词的人数的正 i 系数，以及第六次尝试猜出单词的人数的负 i 系数。cause 这个词在这里是有意使用的。系数表明的相关性可以通过简单的假设来解释为因果关系，即 Wordle 单词完全是随机选择的。这是因为，如果单词是随机选择的，任何相关性报告数量的回溯和过去单词和日期的 hardmoders。

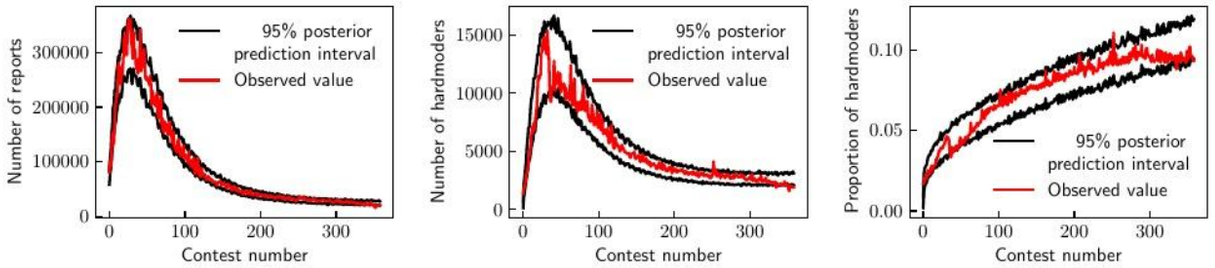

图 8:报告数量、hardmoders 数量和 hardmoders 占过去所有单词和日期的比例的后预测分布

这些模型似乎很好地捕捉了数据的长期趋势和变化。

单词属性和 Try 分布之间的差异一定是由于单词属性对 Try 分布的影响，而不是由于反向效应或混淆变量。

同样，bi 的后验分布基本上是正的。这表明，对于单词越简单，报告结果的人就越多。几乎所有 i 系数的后验分布都包含零，这表明，除了报告总数减少的影响外，对硬模者的数量没有额外的影响。相反，后验分布在很大 1 程度上是负的，这抵消了报告总数减少对 hardmoder 报告数量的影响。同样，从这些结果的相关性中可以读出因果关系，因为这是在假设一天的世界单词是随机选择的情况下解释变化的唯一可能方法。一种可能的因果解释是，在艰难模式下玩游戏的玩家更符合他们报告的结果。

令人惊讶的是，i θ和 i ω系数都以零为中心，表明没有证据表明存在周期性影响。这表明，结合单词难度和长期效应的影响，没有足够的数据来可靠地估计周期性效应。

5.2 Retrodiction

该模型可用于对过去的数据进行预测。图 8 显示了报告数量、hardmoders 数量、hardmoders 所占比例对给定日期和单词的后验预测，以及观测值。该模型似乎既捕捉到了数据的变化，也捕捉到了长期趋势。图 9 显示了真菌(过去的世界词)的尝试分布的后验预测，与事实非常吻合。

5.3 预测

该模型还可用于预测未来的未知数据。这既可以用于未来的日期和单词，比如 2023 年 3 月 1 日的怪异，也可以用于单独的日期(通过对该日期的后验预测，对所有观察到的世界单词进行平均)。图 10 显示了对 eerie 难度的预测。图 11 显示了对 3 月 1 日报告的分数数量、硬模型的数量和硬模型的比例的总体预测。表 1 给出了几种不同兴趣量的预测区间，如果 eerie 是 3 月 1 日的单词。

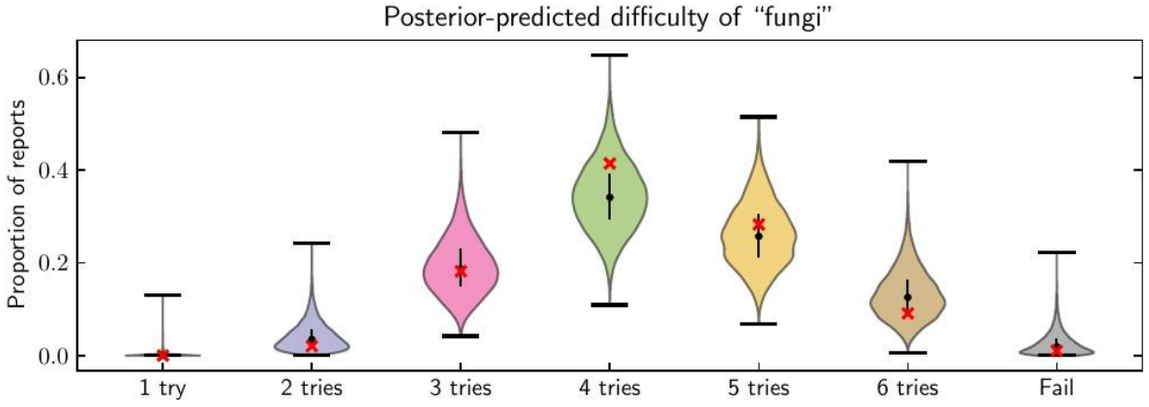

图 9:世界锦标赛选手报告的真菌分布的后验预测

表 1:2023 年 3 月 1 日出现的怪异的几个感兴趣量的预测区间

Variable	95%	80%	50%	Median
Number of reports	[20238, 27876]	[21479, 26365]	[22622, 25169]	23884
Number of hardmoders	[2194, 3239]	[2355, 3048]	[2509, 2870]	2683
Percentage of hardmoders	[9.97, 12.62]	[10.41, 12.15]	[10.79, 11.72]	11.25
Percentage in 1 guess	[0, 2.4]	[0, 0.82]	[0, 0.15]	0
Percentage in 2 guess	[1.09, 14.01]	[2.08, 10.45]	[3.39, 7.70]	5.23
Percentage in 3 guess	[12.16, 35,85]	[15.34, 31.03]	[18.49, 26.82]	22.46
Percentage in 4 guess	[21.29, 48.16]	[25.19, 43.2]	[29.2, 38.51]	33.79
Percentage in 5 guess	[12.48, 37.09]	[16.16, 32.19]	[19.4, 27.96]	23.54
Percentage in 6 guess	[3.73, 21.33]	[5.6, 16.99]	[7.6, 13.53]	10.37
Percentage failed	[0.06, 7.77]	[0.24, 4.91]	[0.66 3.09]	1.6

5.4 难度表示

给定一个单词的 p = (p1)后验分布的样本，…， p7) 7 个向量表示难度，我们可以使用上面描述的嵌入和优化方法来计算一个单词的α和β值的后验样本。图 10 中的 eerie 就是这样做的。值得注意的是，对于 eerie， α和β的后测样本位于α= β线上，这表明α/(α + β)≈0.52 的难度估计相当稳定，但对分布的确切形状的确定性较低(即大多数人是否会在三次或四次尝试中获得单词，或者是否会在两次，三次，四次和五次尝试之间更加均匀)。根据这种难度衡量标准，与之前的世界语言单词相比，这个单词大约位于单词的第 50 百分位数。

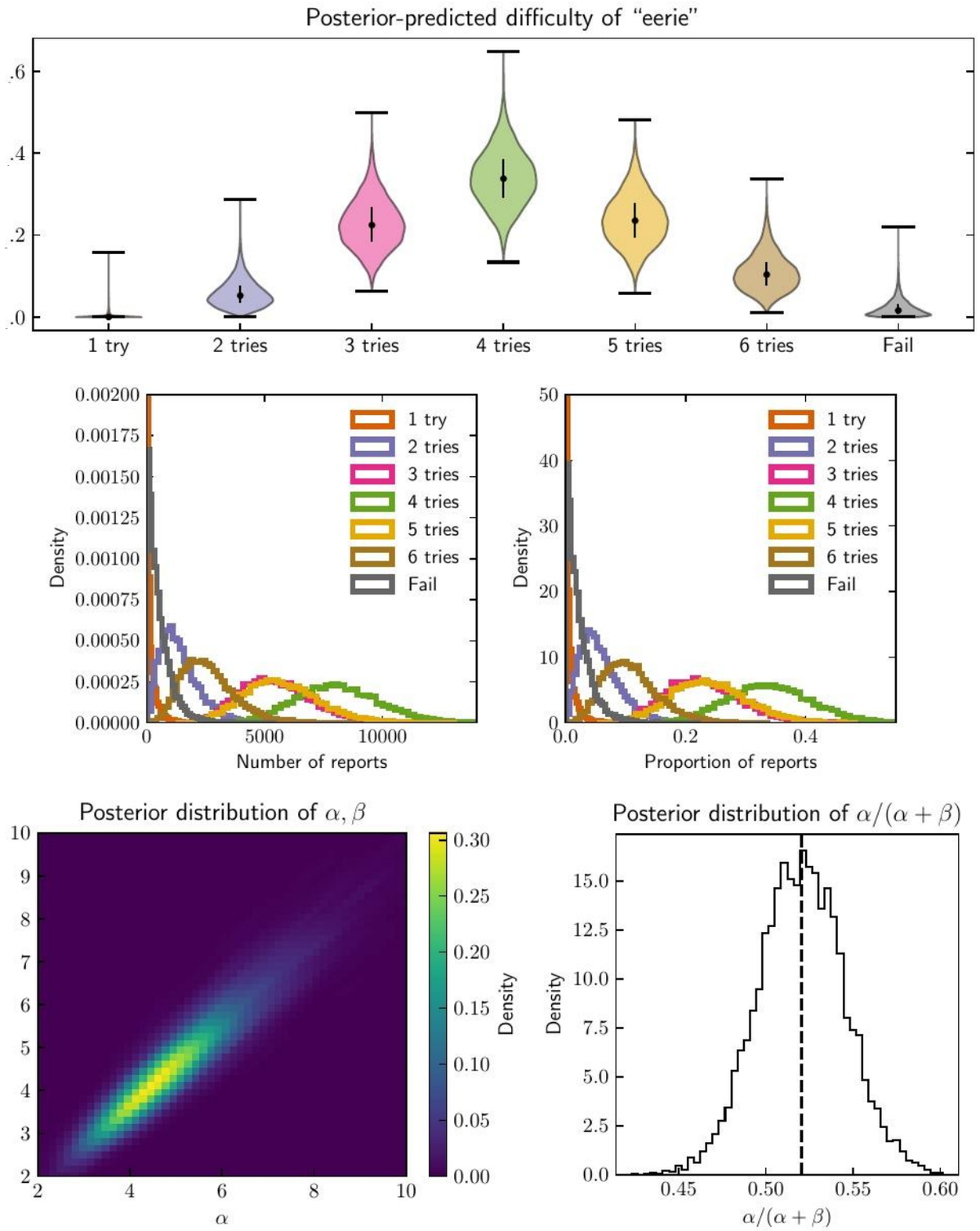

图 10:《世界争霸》玩家报告的怪异难度的后验预测

上图:七个类别中每个类别报告比例的后置小提琴图，其中点和黑线表示中位数和 50%的间隔。中间:7 个类别中每个类别的报告所占比例的直方图。下图:拟合后的α、β和相应 Beta 分布的期望值的后验分布，虚线表示中位数。我们的模型预测，eerie 是一个中等难度的单词，大多数玩家在第四次尝试或之前就猜到了。

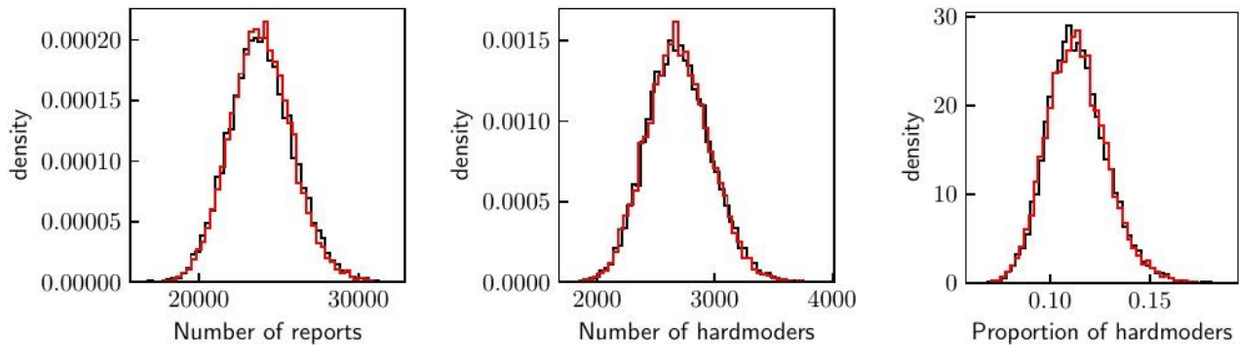

图 11:2023 年 3 月 1 日的报告数量、硬模数量和硬模比例的后测分布(黑色)

如果 eerie 是 2023 年 3 月 1 日的世界单词(红色)。对于这三个变量中的每一个，当对所有之前的单词进行平均时，以及当特别针对 eerie 时，分布之间几乎没有任何差异。

6 模型评估

6.1 限制

子集熵预测器并没有捕捉到一个单词在第一次猜测之后的猜测难度。例如，如果一个玩家，在第一次猜测之后，已经透露了一个单词以 atch 结尾，这个单词可能是 watch, catch, latch，等等。子集熵并不能反映这种难度来源。

子集熵要求我们对玩家首次猜测的分布做出假设。

与传统的统计方法相比，使用 MCMC 采样的贝叶斯模型非常慢。使用 MCMC 从后验分布中抽样的效率远不如使用梯度下降法或解析解来找到使成本函数最小化的参数的最优值

该模型没有考虑到不同日子里玩家平均技能的变化，尽管在单词难度分布中，时间的相关性不如单词本身的方面。

Reports 和 Hardmoders 模型没有考虑数据的自相关性，而是假设了一个稳定的长期趋势，这可能低估了长期预测中固有的不确定性。

6.2 优势

使用贝叶斯框架可以获得任何预测的完整分布，而不是简单的最佳拟合和预测区间，需要近似值和强分布假设。例如，其他模型可能无法捕捉到后验预测分布中的不对称性，即 1 次尝试中猜测怪异的人数和失败的人数。

贝叶斯预测中的不确定性不仅包括数据中真实噪声的不确定性，还包括模型本身的系数和参数估计的不确定性。

我们的模型允许单词的参数对 Try 分布、报告数量和 hardmoders 数量有不同的影响，这是合适的，因为单词难度的不同方面可能会影响这些结果。

我们的模型仅用两个参数(α和β)表示单词的难度，这两个参数可以用不确定性进行预测。当一个世界语言的“难度”是由具有不同背景、策略和投入程度的真人来玩时，是无法精确衡量的。

7 结论

我们用几种方法模拟了世界语言中单词的猜测难度。我们没有一个单一的、规定性的度量标准，而是提出了几个以不同方式对应单词难度的度量标准。然后，我们使用 Lasso 回归选择这些预测因子的一个子集。

通过使用 Beta 分布作为猜测数分布的表示，我们能够将一个单词的难度信息浓缩成两个数字，并从那里浓缩成一个数字，这使我们能够直接将单词相互比较。

使用创新的贝叶斯模型，我们能够预测未来玩家将报告对未来单词进行多少猜测，有多少玩家将在 Twitter 上报告他们的结果，以及有多少玩家将报告在困难模式下玩游戏。该模型的创新之处在于，它由三个子模型组成，当这些子模型以数据为条件时，它们是独立的。因此，整体模型可以相当大，同时仍然足够高效地执行马尔可夫链蒙特卡罗(因为 MCMC 可以在每个子模型上单独运行)，并从后验分布中获取样本。考虑到模型的贝叶斯性质，它还能够以概率分布的形式而不是简单的预测间隔的形式提供任何预测和反演的不确定性。我们提供了几个预测，假设 eerie 是 2023 年 3 月 1 日的世界词，并提供了几个回溯来确认我们的模型与过去的数据一致。

References

[1] Matthew D Hoffman, Andrew Gelman, et al. “The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.” In: J. Mach. Learn. Res. 15.1 (2014), pp. 1593 – 1623.

[2] Oxford University Press. Home : Oxford English Dictionary. url: https://www.oed.com/ browsedictionary.

[3] John Salvatier, Thomas V. Wiecki, and Christopher Fonnesbeck. “Probabilistic programming in Python using PyMC3 ” . In: PeerJ Computer Science 2 (Apr. 2016), e55. doi: 10.7717/peerj-cs.55. url: https://doi.org/10.7717/peerj-cs.55.

[4] Claude Shannon. “A Mathematical Theory of Communication”. In: Bell System Technical Journal 27.4 (Oct. 1948), pp. 623 – 656. doi: https : / / doi . org / 10 . 1002 / j . 1538 - 7305.1948.tb00917.x.

[5] Robyn Speer. rspeer/wordfreq: v3.0. Version v3.0.2. Sept. 2022. doi: 10 . 5281 / zenodo . 7199437. url: https://doi.org/10.5281/zenodo.7199437.

[6] Stanford University. Knuth: The Stanford GraphBase. url: https://www-cs-faculty. stanford.edu/~knuth/sgb.html.

[7] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: Nature Methods 17 (2020), pp. 261 – 272. doi: 10.1038/s41592-019-0686-2.

致纽约时报谜题编辑的信

亲爱的纽约时报谜题编辑器，

每天玩世界大战已经成为世界各地许多人的消遣。《纽约时报》已经建立了一个忠实的粉丝群，游戏的易用性也允许任何人偶尔接受每日挑战并将结果发布到社交媒体上。我们的部分目标是发现导致每日报告出现这种变化的原因，以及你作为谜题编辑器选择的单词难度如何影响玩家的结果。我们开发的贝叶斯模型可以提供未来玩家数量的分布，以及基于任何选择的单词的结果分布。

我们发现，玩家报告中最大的变化是由于单词的难度。当单词比较简单时，报告结果的人数就会更多。玩家数量整体呈下降趋势，这可能是由于《世界大战》作为一款游戏的老化。另一方面，文字难度也没有对报告玩高难度模式的玩家数量产生实际影响。我们还看到，随着时间的推移，难度模式玩家的比例也在增加，这表明更多的休闲玩家离开了游戏，而那些更专注的玩家，也就是那些更有可能玩难度模式的玩家，占据了更大的玩家基数。

对过去文字和日期的报告数量和硬模玩家的回溯。

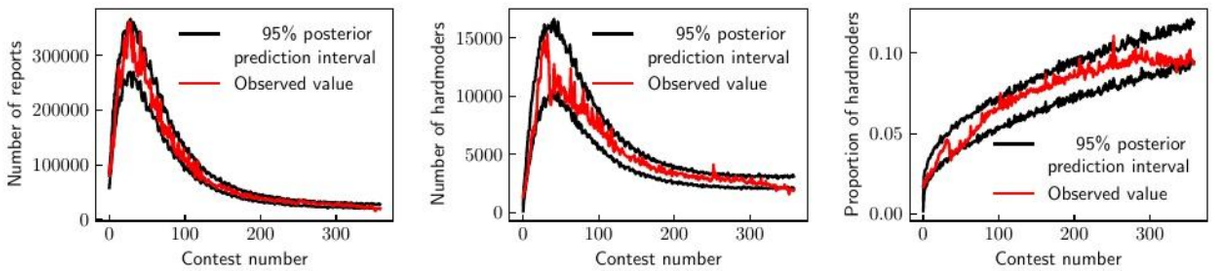

图 12:报告数量、hardmoders 数量和 hardmoders 占过去所有单词和日期的比例的后验预测分布。

至于预测单词的难度，我们发现我们可以用 Beta 分布准确地模拟玩家尝试的分布。这样的分布被参数化为(α， β)，我们发现相对于β更高的α表示这个单词更难猜。这些差异可以从“Swill”这个词的不同α和β值中看出，我们相信你会同意，这个词比“Saint”这样的词更难猜。然后，我们使用各种指标，包括唯一字母的数量，英语中的单词用法，以及与常见的开始猜测的字母差异来预测这些参数。

根据我们对单词难度评分的方法，“Eerie”的难度为 0.52，它将落在单词的第 50 百分位左右，与过去使用的“Rhyme”、“Equal”和“Quirk”等单词的难度相似。这是有直观意义的，因为虽然它有很多重复的字母，而且是一个边缘词，但“e”、“r”和“i”的共性使得很多最常见的开始猜测都会揭示这个词的信息。

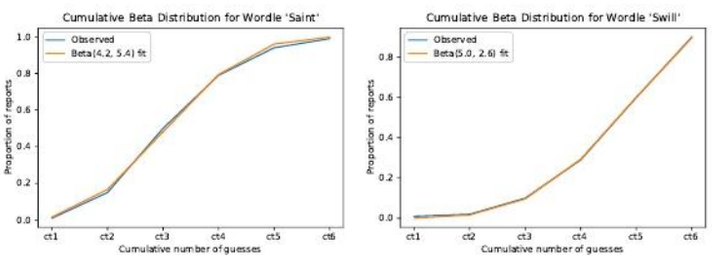

图 13:Beta 分布与不同难度世界大战的累计猜测比例相拟合。

如果 2023 年 3 月 1 日的世界恰好是“怪异的”，那么我们有 95%的信心在 Twitter 上报告的玩家范围将在[20238,27876]之间，其中大约 11.25%是困难模式玩家。我们预测大约 34%的玩家会在第四次尝试时报告猜

对了，我们预计大约 1.6%的玩家会猜错这个词。这些确切的结果也总结在下面的表格和图表中。我们希望这种分析可以帮助更好地为未来的世界选择提供信息，我们期待着即将到来的谜题。

表 2:2023 年 3 月 1 日出现的怪异的几个兴趣量的预测区间。

Variable	95%	80%	Median
Number of reports	[20238, 27876]	[21479, 26365]	23884
Number of hardmoders	[2194, 3239]	[2355, 3048]	2683
Percentage of hardmoders	[9.97, 12.62]	[10.41, 12.15]	11.25
Percentage in 1 guess	[0, 2.4]	[0, 0.82]	0
Percentage in 2 guess	[1.09, 14.01]	[2.08, 10.45]	5.23
Percentage in 3 guess	[12.16, 35,85]	[15.34, 31.03]	22.46
Percentage in 4 guess	[21.29, 48.16]	[25.19, 43.2]	33.79
Percentage in 5 guess	[12.48, 37.09]	[16.16, 32.19]	23.54
Percentage in 6 guess	[3.73, 21.33]	[5.6, 16.99]	10.37
Percentage failed	[0.06, 7.77]	[0.24, 4.91]	1.6

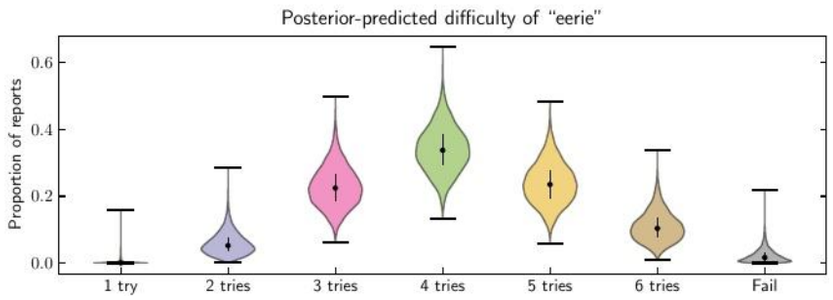

图 14:七个类别中每个类别的报告比例的后置图小提琴图，其中点和黑线表示中位数和 50%的间隔。