基于游戏后台数据生成和数据清洗的实现——以《英雄联盟》游戏为例

李锐 王科
四川大学锦城学院计算机与软件学院 四川 成都 611731

【摘 要】当代互联网的蓬勃发展伴随着信息量的指数级增长，以游戏数据为例，每时每刻有多少玩家同时在线，每天又有多少玩家击败对方玩家。繁杂的信息中，我们总能提取到有价值的信息，对其加以分析，就足以让我们把这些信息利用起来。例如，掌握游戏中竞争对手的数据，分析对手的玩法，找到对方的弱点，对对手进行克制，进而打败对手。基于以上需求，本项目模拟了比赛数据的生成，与展示分析，对英雄联盟游戏数据进行可视化处理。本文仅讨论数据的生成与过滤部分。

【关键词】大数据；英雄联盟；数据生成；数据过滤

1 绪论

1.1 大数据背景

自 2012 年开始，人们所处的网络环境俨然从“信息时代”阔步迈入了“大数据时代”。人们掌握着各种各样的信息。随着信息量的与日俱增，获取信息的方式和冗余信息、垃圾信息等数据的过滤就显得尤为重要，这样才能获取到对自身具有价值的信息，进而在此基础上进行长远的、正确的决策分析。

1.2 电竞背景

2001 年，由 WCG 举办的《星际争霸》大赛中，中国人首次站在了电竞领奖台，2007 年，《DOTA》的竞赛再次让中国人站上了世界的领奖台。《英雄联盟》游戏的出现，更是让“电子竞技”成为了一项值得令人关注的赛事。不仅有中国赛区的《LPL 英雄联盟职业联赛》，更有精彩的《英雄联盟全球总决赛》。自 2012 年，英雄联盟 S2 赛季中，中国 TPA 队伍获得全球总决赛的冠军开始，几乎往后每一届英雄联盟全球总决赛颁奖台都能看到中国队伍的身影。S8 赛季中，IG 电子竞技俱乐部勇摘桂冠，S9 赛季中，FPX 电子竞技俱乐部代表中国再续辉煌。

1.3 大数据与电竞的关系

每一场比赛当中，每一位选手都会为团队的胜利做出自己的贡献，在这背后，全都是海量数据的支撑：比赛地图上的走位路线、对对手伤害量、承受对手伤害量、局内经济增长速度、击杀次数、死亡次数、助攻次数以及由此对本场游戏表现进行评价的 KDA 分数。

当对这背后的个人选手数据进行足够的分析，则可以推测出选手的个人游戏习惯。对战队数据进行足够的分析，或许可以推测出对手下一场比赛可能采取的策略，以便指定应对措施。

倘若对这些数据视而不见，每一场游戏都随机应变，可能就抓不住对方的弱点，进而失去胜利的机会。只有善用数据，分析数据，才能使得每一场比赛都有它的价值，而不是仅仅当时分个成王败寇。

2 理论基础

要做一个数据的生成，首先应该首先爬取这些数据，明白它有那些数据字段，以及这些数据的规则。其次由于实际电竞比赛场次有限，再在对基础数据的基础上，按照一定数据规则，生成波动幅度不大的数据。然后再通过数据过滤，对生成的数据做出一定的限制。

2.1 数据爬取

通过网页浏览发现，虽然英雄联盟官网有一些数据，但是不方便进行爬取。另外有一家名叫《玩加电竞》的网站，在这一专注电竞的网站上，有需要的数据资源，且分类明确，信息抓取方便。

对于信息的爬取，本项目采取 python 中较为强大的 scrapy 爬虫进行信息的爬取。爬取的主要内容为该网站中，英雄联盟数据库中，历届全球总决赛的选手的各项 KDA 信息。

2.2 数据分析

通过 python scrapy 爬虫爬取到的信息为整个赛季中，战队、游戏位置、KDA 等信息，对这些信息的数据类型、

数据特征进行分析，以便为数据生成步骤奠定理论基础。

2.3 数据生成

一般项目可以直接分析数据之后，优化数据爬虫，抓取更多信息。

由于本项目抓取数据量不够，故采用模拟数据生成。

通过对数据的分析，确定了需要的数据类型，引用 python 的 random 库与 faker 库对所需要的数据进行模拟生成。数据量根据前端参数确定。

2.4 数据清洗

在生成数据的时候，可能会有一些数据在生成时候无法控制，就需要数据大数据框架下的 flume 组件对生成的文件中的一条条数据进行过滤，剔除掉生成数据中，无法限制的那些非法数据，再进行数据库存储。

3 具体实现

3.1 数据爬取

新建 scrapy 爬虫项目，定义数据字典。通过 css 选择器以及 xpath 选择器，逐渐限定所需要的数据的范围，获取数据后添加到数据字典。在此基础上，进行基础数据清理工作，如获取到的数据包含了 html 标签等，进行内容的提取。

3.2 真实数据分析

爬取数据字段逐一进行数据分析：

选手：选手姓名，英文字符串。

战队：战队名称，大多 2 至 3 个字母，英文字符串。可使用枚举。

位置：“上单”、“打野”、“中单”、“辅助”、“ADC”可枚举字符串。

出场次数：整数。

KDA：Kill、Death、Assistant 的比值，计算公式为 (K+A)/D，保留一位小数。

参团率：团战参与次数与团战爆发次数的比值，百分数，保留一位小数。

场均击杀：击杀总数与场次的比值，保留一位小数。

单场最高击杀：单场最高击杀数，整数，一般不超过 30。

场均死亡：死亡总数与场次的比值，保留一位小数。

单场最高死亡：单场最高死亡数，整数，一般不超过 30。

场均助攻：助攻总数与场次的比值，保留一位小数。

分均经济：平均每分钟获得金币数量，300 至 500。

均分伤害：平均每分钟给出的伤害数。

3.3 模拟数据生成

在实际项目中，数据生成应该采用 scrapy 爬虫爬取有效历史比赛数据，以保证数据的真实性与可参考性。

本项目中，采用 python faker 库对随机数据进行生成。现对数据生成方法进行分析：

选手名字：faker 库 name 方法，取倒数第一个名字。

战队、位置：根据预定义的列表，通过 random.choice(list) 自动选择。

KDA、参团率、场均击杀、场均死亡、场均助攻：通过 faker.pyfloat() 进行生成，参数限定整数位数、小数位数和数据正负。

最高击杀、最高死亡、分均经济、分均伤害：通过 random.randint() 生成，参数限制上下限。

赛季、数据量：通过前端传入的参数进行限定。

3.4 模拟数据过滤

启动 flume 程序，监测生成文件的尾部是否有写入信息。如果有，则按照以下正则表达式规则进行数据过滤：

a. -\\d*\\.?\\d*

“-”指定数据中的负值，“\d*”指代 0 个或多个数字，“\\.”指代小数点，“?”指代 0 个或 1 个。该正则表达式主要功能是过滤负值，包括整数和小数：

b. 0\\.0\\d*

“0”指代数字 0，\\. 指代小数点，“\d*”指代 0 个或多个数字。该正则表达式的主要功能是：过滤数据当中过于小的异常数据。

c. (?<!\\.)[5-9][0-9][0-9]

“（）”限定一个原子组，“？ <！”是零宽负向后行断言，是正则表达式中的条件判断，判断为“前面不是什么”，“\\.”指代小数点，“【A-B】”指代从 A 到 B 的所有字符，连续的字符构成多位数。该正则表达式的主要功能是：过滤数据中的大于 500 的三位数。

如果每次写入的数据，有任意符合以上条件的数据流，则把数据过滤掉，不生成在文件里，实现数据的规范。

Flume 的 sink 组件以 hdfs 方式执行，设定存储的 hdfs 路径，使用本地时间戳每日轮询。把生成的模拟数据保存在 hdfs 内，再将数据导入到 hive（mysql 作为元数据库）。

4 实现效果

4.1 数据生成前端

设定基础的必填参数，通过 axios 请求把参数传到后端。

用户：文本框填写。

生成数据条数：文本框填写。

赛季选择：下拉列表选取。

确认提交按钮：提交参数，开始生成数据。

重置按钮：清空参数，重新选择。

图 1 数据生成前端

Fig.1 Data generation front page

4.2 数据生成后端

接收前端传的参数进行参数绑定，将当前前端数据参数存入数据库的“数据生成”表中。并调用系统命令执行数据生成脚本，产生模拟数据。该产生的数据文件被 flume 监听，以过滤生成的非法数据。

4.3 数据过滤后端

前端每次数据生成参数记录通过 jdbc 保存到数据库。

调用 Runtime 类运行数据生成脚本，参数为前端传入的赛季、数据量等数据。

根据预先设定的 flume 中的数据过滤正则表达式，对产生的数据文件进行逐行的监听和过滤。确保存入 hdfs 的数据不存在超出数据分析预期的情况，后续再调入数据库进行数据存储。保证后续数据调用的时候不会发生异常。

5 存在的缺陷

5.1 数据生成前端

数据生成页面仅两个输入框，一个下拉列表和“确定”、“重置”按钮。用户交互体验较差。后端数据库字段长度有限，不能任由用户输入过长的项目名称。数据生成需要一定的时间，过多的数据条数会导致用户等待时间较长。

5.2 数据处理后端

数据生成时，数据以逗号连接文本放置在文本文件中。在文本文件或 csv 文件中看来，用户能简单明了的直接对应哪个位置是哪种数据，但是在数据过滤后的数据流处理中，通过逗号分隔具体数据的时候，相对计算比较麻烦。需要从数据流中取值再存入数据库。

6 改进

6.1 数据生成前端

对前端页面进行如下改动：

a. 限制文本框最大输入长度为 20 字符。解决因用户输入过长而导致的信息存入失败的问题。

b. 添加文本字符统计。使得用户能够一目了然已输入长度和最大长度。

c. 添加一键清除属性。使得用户删除信息更加便捷。

d. 设置所有项目为必填项。以免造成数据库为空字符的异常状况。

e. 给文本框加上提示信息。

f. 提交后呈现加载状态。

图 2 改进后的前端

Fig.2 Improved data generation front page

6.2 数据处理后端

在后端对数据的处理上，由于原文本文件形式不易区分键值，也不便调用数据。故将数据生成步骤中，由生成字符串文本文件的方式改为生成 json 键值对的形式。在读取每一行数据的同时也能更方便调用数据进行操作。

7 总结

本文中，介绍了模拟英雄联盟游戏大数据的生成与过滤，它可以在后续的英雄联盟游戏数据可视化与战队数据、选手玩法等提供了基础的数据保证，方便我们从各个维度进行数据的分析，如：战队的战力雷达图分析、战队 KDA 分析、平均战队杀敌数量等。在真实上线项目中，我们的数据应当采用爬虫爬取的真实英雄联盟游戏历史数据，这样才能使得数据更加有分析的价值。

不仅局限于电竞，生活中仍有各种各样的数据存在，并且都在与日俱增，在这“大数据时代”日趋成熟的阶段，掌握好大数据技术，不仅是掌握当下，更是在掌握未来。

【参考文献】

[1] 方晶晶 , 刘佳峰 . 数据挖掘技术在企业服务中的应用与研究 [J]. 电脑编程技巧与维护 ,2020(06):105-107.

[2] 黄素萍 , 常加强 , 高妍 . 海量数据的分析研究 [J]. 科学技术创新 ,2020(15):60-61.

[3] 罗长银 , 陈学斌 , 宋尚文 , 刘洋 . 数据预处理技术在异构数据中的应用 [J]. 软件 ,2020,41(05):6-13.

[4] 王萍 . 领域大数据应用开发与运行平台技术研究 [J]. 信息与电脑 (理论版),2019(14):147-148.

[5] 李绍杨 . 领域大数据应用开发与运行平台技术研究 [J]. 信息记录材料 ,2018,19(01):46-47.