

可视化工作——第一部分

1.任务描述

1.1 数据简介

大连海事大学中国欧盟可用性研究中心(Sino European Usability Center)制作的可以安装在移动设备上的用户体验数据采集工具 CAUX(content-aware user experience research), 这个工具采集移动设备的运行状况(手机的电量、亮度、加速度、耳机接口、网络状况等), app 使用情况(打开, 退出, 退回后台的时间等), 通过日志法记录采集到的数据并录入数据库。

这部分工作目标是基于 app 各种状态的时间点对用户的 app 使用情况进行可视化。结合本部分工作的分析目标, 提取了“ 2018-03-23” 和“ 2018-03-24” 这两天的状态时间, 状态信息, 应用名称, 由于只是对于可视化技术的尝试性使用, 只涉及到了这两天的 137 条数据。

1.2 可视化任务

单个用户的 app 使用行为分析:

- (1)用户在某个周期内(周/月/年)的对于单个 app 的使用频率/时间的可视化
- (2)用户在一天内对于单个 app 使用情况的可视化
- (3)以上两者可以进行联系, 在观察员观察对周期内某一天的数据感兴趣时可以查看当天详情

2.工作概览

这部分工作可以分为数据存储模块、数据传输模块、数据处理和可视化模块。

数据存储模块采用了 mysql 数据库进行存储;

数据传输模块即在 php 文件内连接数据库，接收前台 html 页面传入的关键字生成 SQL 语句在数据库里进行检索，并通过 json 文件将检索结果发送到 html 页面，其中的 JavaScript 代码段使用 Ajax 接受 json 文件中的数据；

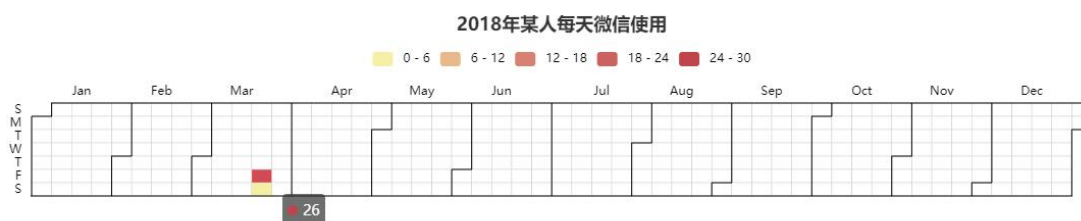
数据处理模块即 JavaScript 中对于 app 状态时间数据格式的转换、app 状态持续时间统计、app 使用频率/总时间计算的代码段。可视化模块即 JavaScript 中使用 echarts.js 的代码段，这部分工作参考调用了 echarts.js 官方示例中的热力图和自定义图的代码。

3.可视化方法和具体实现

对于任务 1，使用热力日历图进行实现；对于任务 2，采用了甘特图进行实现；对于任务 3，实现方法是：在对热力日历图中的某一天数据点击时，将其日期值传到网页 url 并进行刷新，刷新后网页读取到网页 url 栏日期值，显示相应日期的甘特图。

3.1 热力日历图

热力图主要通过颜色去表现数字的大小，并通过图例来帮助研究人员理解和筛选所需要显示的数据。而热力日历图这是将热力图与日历相结合，进行一段时间内时序数据的可视化，通过对一整年数据的同时展示，希望研究员能发现其中周期性的规律。在这里我们采用了颜色来表现用户在当天对于“微信”app 使用的频率，使用频率越高，颜色越接近深红色，频率越低，颜色越接近浅黄色。由于我们只抽取了两天的数据进行小规模实现，图中只有 2018-03-23 和 2018-02-24 这两天有显示，其中 23 号的频次是 26，24 号的频次是 1。当光标移动到相应的日期时，会出现提示框显示相应的频次



3.2 甘特图

甘特图(gantt charts)又称为横道图，条状图，通过横着条状来显示和时间相关的项目随着时间变化的情况。在这个工作中我们使用纵轴显示 app 的名称，横轴显示时间的甘特图来对任务 2 进行时序数据可视化。如图，为某一用户在 2018-03-23 当天的“微信” app 使用情况。由于只是对单一 app 进行可视化，故纵轴只有“微信”一项，横轴为时间，其区间为当天的 0:00 到当天的 24:00(第二天 0:00)。考虑到凌晨睡觉时间段 app 次数使用极少，以及其他时间段单次 app 使用持续时间较短，为了防止显示大面积空白和因持续时间过短导致条状过细使得研究人员错漏数据的情况发生，将初始化显示的时间段设置为 7:12-12:00(30%-50%)，并在下方设置数据区域缩放，研究人员可以通过鼠标滚轮缩放，通过拖动缩放条调整查看的时间区间。当光标移动到相应的条状图形是，会出现提示框显示 app 名称和本次使用持续时间。



4.工作遇到的困难

4.1 对于可视化工具的不熟悉, 之前采用的是 python 里的 pyecharts 包(提供了 echarts.js 的 python 接口)进行数据可视化并发挥 python 对于数据处理的优势, 但由于对于 pythonWeb 框架的不熟悉导致进度受阻, 后续采用了 php 进行 web 后端建设, 并将数据处理模块的工作放到 JavaScript 中进行, 一方面需要处理的数据不多, 另一方面可以直接调用 echarts.js 进行数据可视化, 只需要关注于 JavaScript 代码的编写即可实现可视化。

4.2 对于 mysql, php, ajax, jquery.js, echarts.js 的学习和运用, 由于甘特图是使用自定义实例代码修改而成的, 学习资料较少, 需要仔细阅读官方文档。

4.3 数据格式处理

一开始出现数据无法可视化成功的情况, 后续通过对每个字符串都进行输出发现, 由于数据库的时间类型(datetime)在 JSON 文件中会被转换为字符串, 且与 echarts 中所要求的时间字符串格式不一致。再调用 echarts.js 进行可视化之前需要人为地转换为时间戳格式再计算转换为符合格式要求的时间字符串格式。

对于持续时间的计算, 需要将 JSON 中的字符串格式转换为时间戳格式, 相减之后再换算显示成“XX 小时 XX 分钟 XX 秒”。