

城市供水基础设施问题的事件识别

一、设计目的及意义：

主要有两个目的：

1. 提取工单文本中的问题关键词，从时间、空间和文本 3 个维度，提出一个聚类算法，识别事件。

2. 设计一个空间可视化方案，分析事件的空间分布规律。

意义：

1. 将城市供水的问题在时空上展现出来，对于普通民众可以直观上看到城市供水问题上的时空分布规律。

2. 将城市供水潜在的一些问题凸显出来，让管理者从中获取一些信息，更直观的了解问题，发现新问题，做出更好地决策。

二、主要方法与技术：

1. 数据预处理，将数据按关键字分类

1) 使用 jieba 分词，并获取关键词，并与人工检索相结合得到最终的关键词。

2) 添加否定不敏感的前缀后缀判断。

3) 引入关键字等价集来进行数据的事件分类。

2. 数据可视化：

1) 将事件按照时间和空间关系投影到地图上。

2) 地图支持 zoom，平移和放缩，放缩是按照鼠标为中心进行放缩的

3) lazy load：只显示屏幕内的数据，当数据超过 1000 时自动聚类。

4) 使用饼状图聚类。

5) 数据分为三大类 20 小类。三角，正方形，圆形代表三大类（水，设备，范围、同和 other），20 种颜色代表不同小类。

6) 按时间筛选，按事件筛选数据。

7) 动态时间浏览，要求动画同时不能卡死，

8) 使用 toolBar 查看详情，聚类时查看统计结果，不聚类时显示数据详情。

3. 火狐与 Google 的兼容

三、设计步骤:

数据预处理（LDA 主题提取的失败）

1. 数据格式转换:

Oracle SQL 语句 → MySQL SQL 语句 → CSV 格式数据

提供的数据库是 Oracle 的 SQL 语句，有一个时间的函数在 MySQL 中没有对应，于是手写了一个 python 脚本将 Oracle 的 SQL 语句（water.sql）转成 MySQL 的 SQL 语句(water_cahnge.sql)。

```
#coding=utf-8

import re

filep = open("water.sql");

filew = open("water_change.sql",'w');

for words in filep.readlines():

    words = re.sub("to_date\(", " ", words);

    words = re.sub(", 'dd-mm-yyyy hh24:mi:ss'\)", " ", words)

newwords = "";

i=0;

while(i < len(words)):

    word = words[i:i+19];

    if re.search("(\d\d\d-\d\d\d-\d\d\d\d\d\d:\d\d:\d\d\d)", word) != None:

        newwords += word[6:10]+"-";

        newwords += word[3:5] + "-";

        newwords += word[0:2]

        newwords += word[10:19];

        i = i + 19;

        continue;

    newwords += words[i];

    i = i+1;
```

```
filew.write(newwords);

filep.close();

filew.close();
```

结果示例如下：

Oracle SQL

```
insert into mmworkformok (WORKFORMID, ACCEPTTIME,
EVENTCAUSE, STANDARDADDRESS, LNG, LAT)

values ('20100101000016', to_date('01-01-2010 02:03:17',
'dd-mm-yyyy hh24:mi:ss'), '中山路派出所来电：此处跑水严重，请核
实。)
```

MySQL SQL

```
insert into mmworkformok (WORKFORMID, ACCEPTTIME,
EVENTCAUSE, STANDARDADDRESS, LNG, LAT)

values ('20100101000016', '2010-01-01 02:03:17', '中山路派
出所来电：此处跑水严重，请核实。)
```

使用 MySQL Workbech 建表并插入数据：

建表语句如下：

```
create table test.mmworkformok(WORKFORMID CHAR(14)
primary key, ACCEPTTIME datetime, EVENTCAUSE varchar(450),
STANDARDADDRESS varchar(100), LNG CHAR(10), LAT char(10));

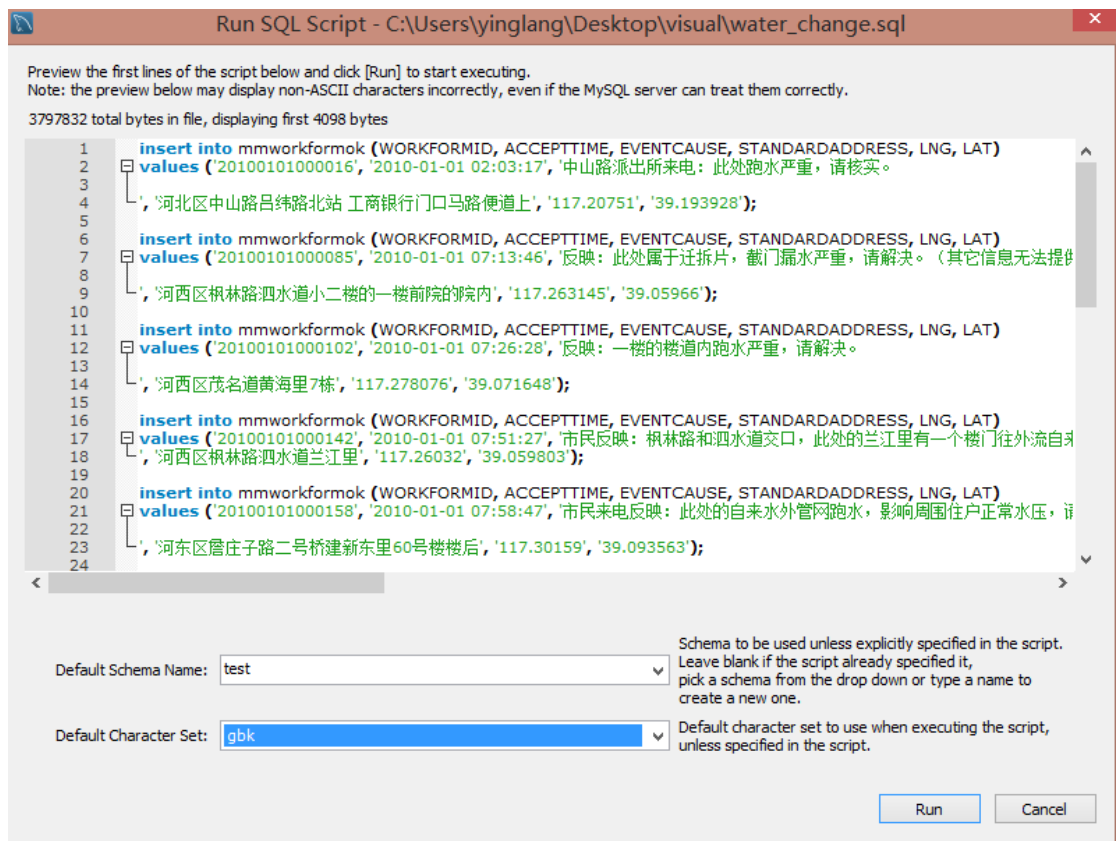
/*gbk 编码用以下*/

ALTER TABLE test.mmworkformok

CHARACTER SET = gbk ;
```

用 MySQL WorkBech 导入 SQL Script，注意要使用【File】→【Run SQL Script】，不要使用【file】→【OpenSQLScript】，后者使用点击闪电执行时会没执行一条停顿 1s。

选择合适的编码方式和数据库



从数据库导出 csv 格式的数据文件 water_data.csv （SQL 语句如下）

```

select * from vis.mmworkformok
into outfile '.\\water_data.csv'
fields terminated by ',' optionally enclosed by '"'
escaped by '\"'
lines terminated by '\r\n';

```

使用下面 SQL 语句找到导出的文件位置

```
show variables like '%datadir%';
```

最终得到 CSV 格式文件 water_data.csv 如下:

```

'20100101000016','2010-01-01 02:03:17','中山路派出所来电: 此处跑水严重, 请核实。
', '河北区中山路吕纬路北站 工商银行门口马路便道上', '117.20751', '39.193928'
'20100101000085','2010-01-01 07:13:46','反映: 此处属于迁拆片, 截门漏水严重, 请解决。 (其它信息
', '河西区枫林路泗水道小二楼的一楼前院的院内', '117.263145', '39.05966'
'20100101000102','2010-01-01 07:26:28','反映: 一楼的楼道内跑水严重, 请解决。
', '河西区茂名道黄海里7栋', '117.278076', '39.071648'
'20100101000142','2010-01-01 07:51:27', '市民反映: 枫林路和泗水道交口, 此处的兰江里有一个楼门往
', '河西区枫林路泗水道兰江里', '117.26032', '39.059803'
'20100101000158','2010-01-01 07:58:47', '市民来电反映: 此处的自来水外管网跑水, 影响周围住户正常

```

2. 提取 EVENTCAUSE 字段

从上面数据的结构可以看出，只有 EVENTCAUSE 字段是要经过一些处理才能使用的。

写一个 python 脚本截取 EVENTCAUSE 字段提取出来，这之间最大的问题在于 python 字符处理的编码问题。

Python 读写文件默认的编码方式是 ASCII，所以每次要使用下面代码改变其默认的编码

```
import sys
reload(sys)
sys.setdefaultencoding( "utf-8" ) # 不写这个，下面的 decode
和 write 函数会报错
```

如果不写上面的语句，很多时候，在后面使用 write 函数向文件里写入 utf-8 编码的数据会报编码错误。

将 utf-8 字符存入文件前最好先 encode 为 utf-8，读取 utf-8 文件并解析时，最好先 decode 为 utf-8。

注意的一点是 string 的 replace 函数在 decode 后会失效，所以一切 replace 操作最好在 decode 前。

```
s = '你是谁'.encode('utf-8') # 现在这句与上面的 s 声明等价
filep.write(s)
words=filep.read().replace('\n','').replace("'",'\\\'').de
code('utf-8') # 变成 unicode 后不能使用 replace,所以必须先行动
```

有时 decode 后的 utf-8 文件第一个字符索引是从 1 开始，输出 0 索引位置总会报编码错误。

最终得到只有 EVENTCAUSE 字段的文件 new_water_data.csv

中山路派出所来电：此处跑水严重，请核实。

河北区中山路吕纬路北站工商银行门口马路便道上

反映：此处属于迁拆片，截门漏水严重，请解决。（其它信息无法提供）

河西区枫林路泗水道小二楼的一楼前院的院内

反映：一楼的楼道内跑水严重，请解决。

河西区茂名道黄海里栋

市民反映：枫林路和泗水道交口，此处的兰江里有一个楼门往外流自来水（具体楼层和门号市民不清楚），请

河西区枫林路泗水道兰江里

市民来电反映：此处的自来水外管网跑水，影响周围住户正常水压，请相关部门尽快维修。

3. python 分词组件——jieba

寻找并学习了 python 分词库 jieba，利用结巴进行了分词

jieba 是一个开源的轻量级中文分词组件，下载地址：

<http://www.oschina.net/p/jieba>

该组件使用的算法如下：

- 1) 基于 Trie 树结构实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图 (DAG)
 - 2) 采用了动态规划查找最大概率路径，找出基于词频的最大切分组合
 - 3) 对于未登录词，采用了基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法
- 它可以进行分词和提取关键字。

4. 进行词频统计（后面发现词频对于筛选关键字作用不大）

原本觉得关键字和词频之间会存在线性相关的联系，所以对 EVENTCAUSE 字段进行了词频统计，得到结果如下

```
rank,word,frequent
1,反映,10222
2,请,10099
3,市民,9762
4,的,7704
5,道,6128
6,号,5600
7,自来水,5085
8,此处,5000
9,水,4664
10,有,4642
11,家中,3703
12,楼门,3265
13,情况,3235
```

可以看到，在这里词频很高的词并不一定是很重要的词，既不能反映该 EVENTCAUSE 记录本身的特点。于是舍弃了这一做法。

5. 提取关键词

试图以 jieba 提取出的关键词为切入口，找到能反映 EVENTCAUSE 记录特点的关键词。

- 1) 利用 jieba 的关键字提取功能提取 500 个关键字

市民,反映,自来水,此处,到户,抄表,家中,维修,漏水,尽快,截门,井盖,水压,停水,核查,排查,核实,来电,情况,部门,自来水管,邻居,压低,圆形,相关,工单,严重,查看,无水,井内,邻居,联系,解决,位置,水表,消防栓,回复,丢失,清楚,调查,表井,及时,井井,电话,楼道,已抄表,设施,管道,交口,井跑,门跑,地面,用水,水流,正常,清水,持续,现场,小区,管网,一口,表井内,没有,工作人员,一个,单位,无法,不详,问题,左右,10,提供,施工,归档,时间,原因,一楼,00,公产,供水,私产,居民,附近,机动车道,产权,房龄,同层,告知,楼前,派出所,希望,失灵,道路,抢修,冒水,号楼,已经,与其,网跑,物业,集团,发现,相同,立管,再次,12,用户,导致,更换,关闭,一直,现在,上门,有人,楼门,铁管,往外,具体,居委会,开始,楼下,造成,塌陷,11,非常,要求,水质,认可,开启,报修,报警,热水器,突然,失效,地下,周围,入户,马路上,20,六楼,解释,地下管道,全楼,人员,户内,二次,住户,烦请,一样,井里,地点,姓名,表井里,使用,此楼,一段时间,有修,30,机械表,拆迁,信息,向上,督办,帮助,IC卡,换管,如此,水管,现象,非常低,马路,井圈,比较严重,底商,该处,已有,之前,整个,至今,由于,怀疑,安全隐患,无盖,不能,房管,情况严重,并向外,上述,结冰,不留,没水,一个月,今天,几乎,未留,二供,影响,往上,很小,污水,15,总表,查办,区内,不好,高峰,出现,路面,需要,两个,同工,旁边,关不上,17,21,卡号,无此,18,师傅,院内,经过,破损,是否,冻裂,表示,排水,此单,已下,地管,水井,时段,16,水费,名称,向外,现跑,可能,赛恩牌,未抄,此井,市政,接报,积水,开车,外网,半个,什么,方向,晚上,联系方式,主管,恢复,有水,有时,22,特别,经常,渗水,内部,早上,市政局,14,供热,门口,并且,表前,正在,厨房,无人,路上,最近,二楼,情况不明,服务,楼上,井室,同串,其它,锈蚀,称此,还是,处理,摔伤,破裂,一周,发黄,民警,卫生间,两天,损坏,消防,时有时无,井往,门牌号,三天,长时间,23,无二供,今晚报,内有,现已,河东区,19,下午,漏点,巡查,方形,井中,其他,内向,具体位置,不到,多次,公司,筷子,改造,目前,昨天,此事,井冒水,PVC,13,存在,费用,热线,刚刚,协商,12319,负责,女士,持续一年,施工单位,进行,七楼,修复,星期,大街,不符,远传表,整楼,掉入,材质,一下,塑料管,从此,及时处理,从井里,能够,地道,回访,非机动车,半年,几天,以上,五楼,28,外跑,楼门口,两口,反馈,小时,政民,生活,上午,路过,平房,工地,早晨,权属,警官,经常出现,危险,联系电话,号单,如找,老化,以前,零距离,同意,中间,比较,河西区,25,27,确定,询问,河北区,横管,新换,桥下,结果,燃气,之后,协调,坡道,黑牛,平台,楼栋,一天,记者,二级,十年,去往,问询,故障,属于,大直,不足,09,26,提前,公示,之间,字样,自行,家住,楼内,施工人员,可用,止水,平时,29,大队,但是,里面,搜打,自行车道,可以,消防栓,井有,表井中,24,50,打开,地址,只有,先生,友谊路,铸铁管,河东,红桥,南开区,很多,整楼,立交桥,不是,三个,富民路,周边,100,整串,楼层,马路边,交通,此片,着急,三楼,长方形,可见,外环线,花园,井上,对面,此门,一串,供水管,中环线,两年,车辆,具体地址,及井内,一圆井,水较,一年,近两天,看到,而且,公寓,中午,欠费,插卡,响声,靠近,楼到,外置,96860,称有,边道,通知,地方,河西,水龙头,卫国,出行,机械,马上,很大,四楼,强烈要求,装修,

- 2) 从里面选取一些比较靠谱的关键字作为候选关键字，比如“漏水”，“停水”等。
- 3) 在 EVENTCAUSE 的文件 new_water_data.csv 中检索这些关键词，结合语境，看这些词能否代表该段话的语义。如果对于绝大多数 EVENTCAUSE，它可以代表语义，则把它作为一个事件，或者说作为一个关键词。

通过对所有数据的前中后段各约 300 条数据的分析，确认了一些关键字。

- 4) 通过人工的对数据分析，又得到一些关键字，最终将事件分成了三大类和 19 小类

跑水	漏水	积水	渗水	污水	水质	水压	停水	冰冻
消防	井盖	截门	水井	井	水管	水表		
附近	邻居	车道						

6. 否定分析

利用提取出的关键字将数据事件分类，分类效果出乎意外，80% 以上的数据得以分类，但查看归类结果后发现，分类存在很大问题。
比如：

西里号楼门后面
少水，但表盘无显示，反复试了几次都不行，无停水，赛恩牌IC卡卡号：，

这个记录明显不是停水事件，但由于关键字比对会被添加到停水事件中。

分析：主要是由于否定的存在，使得数据不准确。诸如“无停水”，“没有积水”，“是否漏水不清楚”这种语句的存在。

解决：

- 1) 确定一个否定词汇集合。

deny = u"没有,不,无,未,非"

- 2) 对匹配上关键字的句子(以,。!?等符号作为分割)中检查有没有否定前缀或否定后缀,因为数据中出现的多重否定基本没有,所以不做多重否定判断。如果有否定前缀或否定后缀,那么不将该记录归为该关键词对应的事件。
- 3) 如上面加上否定排除后又带来新的不准确地问题,比如“水质不好”,“没有井盖”等都是该关键字对应的事件,从而导致否定出错。

因此添加一个“否定不敏感”的判断。

否定不敏感: 如果在一个事件的描述前后加上否定词汇,这个描述还是反映这一事件的,那么称它为否定不敏感的,比如“水压”加上否定“水压不大”反映的还是水压问题;否则就是否定敏感的,比如“漏水”加上否定“没有漏水”。

结合数据本身描述语句的特点和人类语言的习惯定义这 19 类事件是否为否定不敏感(1 表示为否定不敏感,0 表示否定敏感)

跑水,0

漏水,0

积水,0

渗水,0

污水,0

水质,1

水压,1

停水,0

水表,0

冰,0

消防栓,0

井盖,1

截门,1

水井,0

井,0

水管,0

附近,0

邻居,0

机动车道,0

7 关键词等价集

加上了否定限制后，一半以上的数据没有匹配上任何关键字。

通过这些数据的分析，发现这些数据不能归类不是因为合适的类，而是同一事件表述形式太多了。

比如：

- 1 河东司法局马路对面的便道上，从一棵树下跑冒自来水。

2 市民反映：家中无水，同楼多户邻居均有此情况，不是抄表到户，故归档。

事件 1 就是跑水事件，事件 2 就是停水事件，只不过是不同形式的表达，因此一个最直接的解决方法就是建立一个关键词的等价词集。

具体建立方法如下：

- 1) 将 关键词等价集(一开始时只有相应的关键字本身) 和 它是否否定不敏感的 写到一个 keywords.txt 文件中。
- 2) 用程序读取 keyword.txt 文件，利用现有的关键词等价集进行事件匹配。如果未匹配的数据在 10% 以下，那么结束流程。
- 3) 分析没有匹配的前 50 条数据，在这些数据中找到能代表这些数据的关键词，将它加入到 keywords.txt 中语义最接近的关键词的等价集里，如果没有对应的关键词等价集，那么在 keywords.txt 中新添加一个关键词等价集，把该关键词加入其中。
- 4) 回到 2)

(建立 keywords.txt 的目的是，当匹配程序写出后，要修改关键词等价集，不需修改程序，只需修改配置文件即可，更加方便，不易出错。)

通过上面的方法建立关键词等价集。

结果如下：

[跑水, 跑, 返, 冒, 流, 溢, 涌]; []; 0

[漏水, 漏]; []; 0

[积水]; []; 0

[渗水, 冒水]; []; 0

[污水]; []; 0

[水质]; [黄, 蓝, 白]; 1

[水压, 水流]; []; 1

[无水, 没有水, 停水]; []; 0

```
[水表];[冻];0
[冰, 冻];[];0

[消防栓];[损坏];0
[井盖, 井无盖];[掀开, 丢失, 坍塌];1
[截门];[失灵, 锈蚀];1
[水井];[塌陷];0
[井];[];0
[水管];[冻裂, 铁管];0

[附近];[施工, 装修];0
[邻居, 旁边];[];0
[机动车道];[];0
```

其中第二个中括号里的是对事件的一些具体描述的匹配，比如“水质不好，发现黄色浮着物”。这种描述中，“黄”就是一个描述词

最后分类挺好，有大概 90% 以上的数据被合理的分类（匹配上了关键词等价集）。

发现剩下的单子里有很多都是二次电话。

例如：

1 同 20090101000085 工单，故归档。

2 再次来电反映：可见 20100102000036 工单，已反馈自来水，让其耐心等待。故归档。

将他们（同，可见）再加入关键字，最后大概 99% 以上的数据被归类。

8. 生成 Json 数据文件

将分类好的数据加上数据本身有的时间和空间数据，存储到 json 文件里。（不知道有库，纯手写）

单条记录基本格式如下：

```
{"index": "0",
  "workFormId": "20100101000016",
  "LAT": "39.193928",
  "LNG": "117.20751",
```

```
"eventCause": "中山路派出所来电：此处跑水严重，请核实。",
"standardAddress": "河北区中山路吕纬路北站 工商银行门口马路便道上",
"acceptTime": "2010-01-01 02:03:17",
"eventSet": [
{"eventId": "0", "eventType": "跑水", "eventDescribe": []}]}
```

```
{"index": "18",
"workFormId": "20100101001986",
"LAT": "39.150673",
"LNG": "117.257545",
"eventCause": "反映：楼门前自来水表井截门失灵且漏水（铸铁），请尽快维修。",
"standardAddress": "河东区常州道爱营里8门306",
"acceptTime": "2010-01-01 15:13:30",
"eventSet": [
{"eventId": "1", "eventType": "漏水", "eventDescribe": []},
{"eventId": "8", "eventType": "水表", "eventDescribe": []},
{"eventId": "12", "eventType": "截门", "eventDescribe": ["失灵"]},
{"eventId": "14", "eventType": "井", "eventDescribe": []}]}
```

注：对于数据库中的一条数据记录可以分到很多个类所以分类是一个多分类，这也是由数据本身决定的。

整个流程的代码可参考附带的代码。

数据可视化

界面设计方案：

界面设计主要分为三个部分，

一：可视化显示主体部分

二：右侧工具栏部分

三：右下侧词频显示部分。

这个三个部分是由不同页面显示，然后由 frameset 整合至一个页面。

1: 可视化显示主体页面: 该页面是整个可视化视图的主体部分, 整个可视化方案是在天津地图上实现, 通过右侧工具栏可对其进行操作, 并在地图上显示相应的效果。在该部分页面以黑色为背景, 地图显示占据了整个页面。在未进行操作之前, 地图上不显示任何数据, 当鼠标悬停在地图上方时可以通过颜色变化来显示所在区域, 还可通过鼠标滚动来放大页面, 放大时是以鼠标所在位置为中心放大, 通过缩放方式更加有利于用户对数据的观察。根据用户在右侧工具栏中选择的交互方式地图上呈现相应的变化。当地图上呈现聚类数据时, 数据会以饼状图的形式在地图上分布。当用户将鼠标移动到某个饼状图的某一部分时, 会出现信息框显示相应的统计信息。当数据呈未聚类状态时, 鼠标移动至其上方时会显示整个工单信息。

2: 右侧工具栏: 该栏集合了本可视化设计方案中的大部分交互手段。主要由四个部分组成。第一个部分为聚类选择。因为整个可视化的数据量太大, 高达上万条, 如果没有相应的聚类, 数据在地图上显示将会重叠, 从而无法观察出数据中蕴含的信息。因此, 在该设计中为用户提供了聚类选择。在右侧工具栏中第二栏便提示用户是否选择聚类。用户可以点击右侧聚类选择按钮来进行交互。聚类选择按钮是由 css 所实现的一个选择按钮。该按钮由三部分组成, 一个整体框架, 一个划动条, 当用户点击按钮时划条将会移动, 在框架两侧还有状态显示字符, 随着用户的交互, 划条会移动, 状态字符也会改变。当屏幕上数据量过多时, 用户可以点击聚类按钮来实现聚类。但是当屏幕数据量大于 1000 时, 数据会自动聚类。

第二部分为关键字选择部分, 通过分析数据, 主要将所有事件分成 21 类。并用 21 个关键字表示。所以在该部分主要设计了 21 个按钮, 用户可以通过点击按钮来选择显示一类事件。可以同时选中多个事件。当事件未被选中时按钮呈浅色。当用户点击某个按钮时, 左侧地图上将会显示相应的事件。每一类事件均用不同的颜色表示。当在未聚类的状态下, 不同的事件还呈现不同形状。当某类事件被选中时, 按钮的颜色也会变化成相应的颜色。

第三部分为时间选择条。用户可以通过对时间选择条的操作来查看不同时段内的事件。当用户进行操作时需要选择一个起始时间和一个截至时间。此时地图上将会显示该时段内的事件。

第四部分为播放器。当点击播放器时, 事件会以时间条所选中的时间间隔为单位演变, 呈现动态效果。

以上四部分操作均可同时进行, 用户可以选择多项操作。

3: 右下图词频显示部分。

通过对数据分析，已将所有工单的关键词切出，并统计了相应的词频。该页面主要是用 **Bubble**（气泡图）来显示词频信息。每一个词用一个气泡表示。气泡大小与词频想对应。每一个气泡呈现不同颜色。每一个气泡中标识了词频信息（词，频率）。所有气泡以一个大气泡为中心。该中心气泡是某个气泡的放大版。因为每个气泡大小不一样。有的气泡信息可能不是很清楚。所以可以通过点击气泡使其交换至中间呈现放大状态。这样就有利于用户的观察。

二：页面显示效果

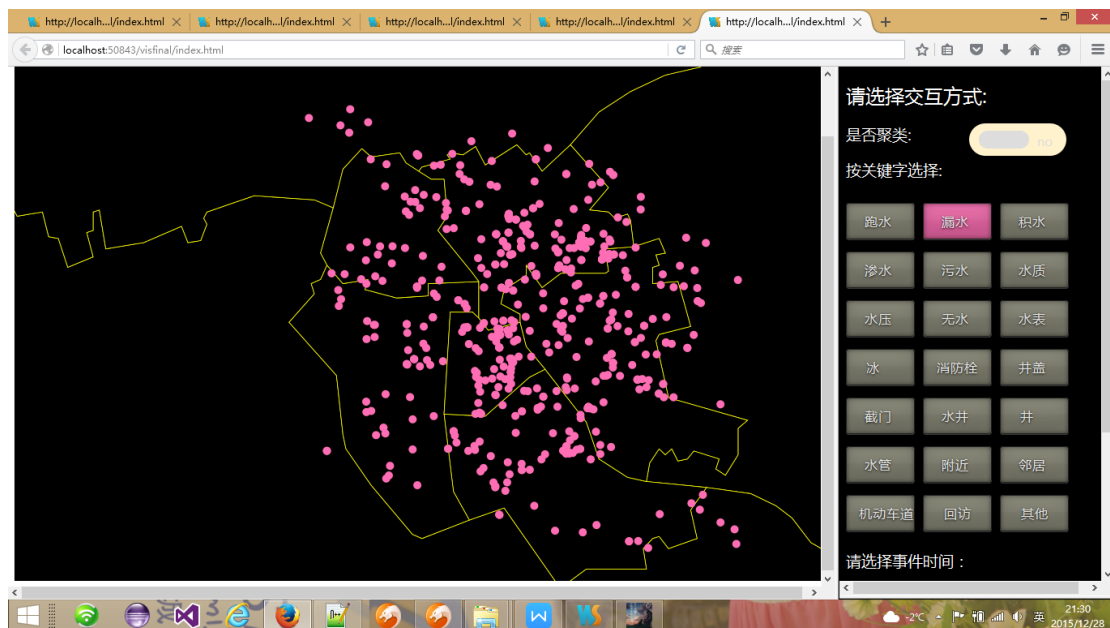
（1）初始页面





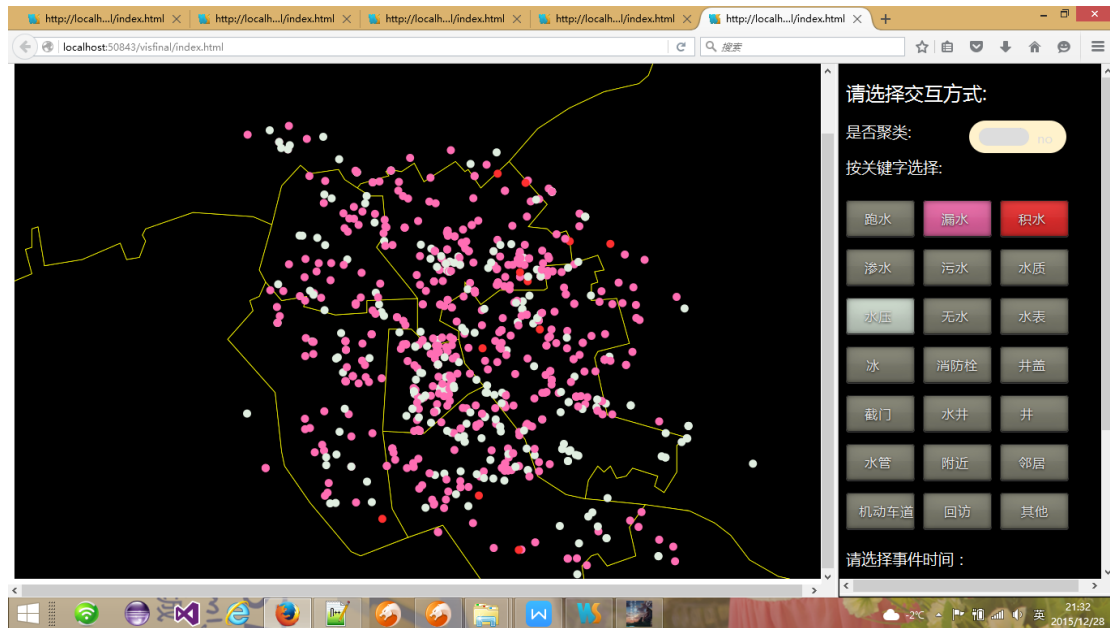
(2) 单事件未聚类状态

解决加载大量数据卡死的问题，lazy load，只显示屏幕内的数据。



(3) 多事件未聚类状态

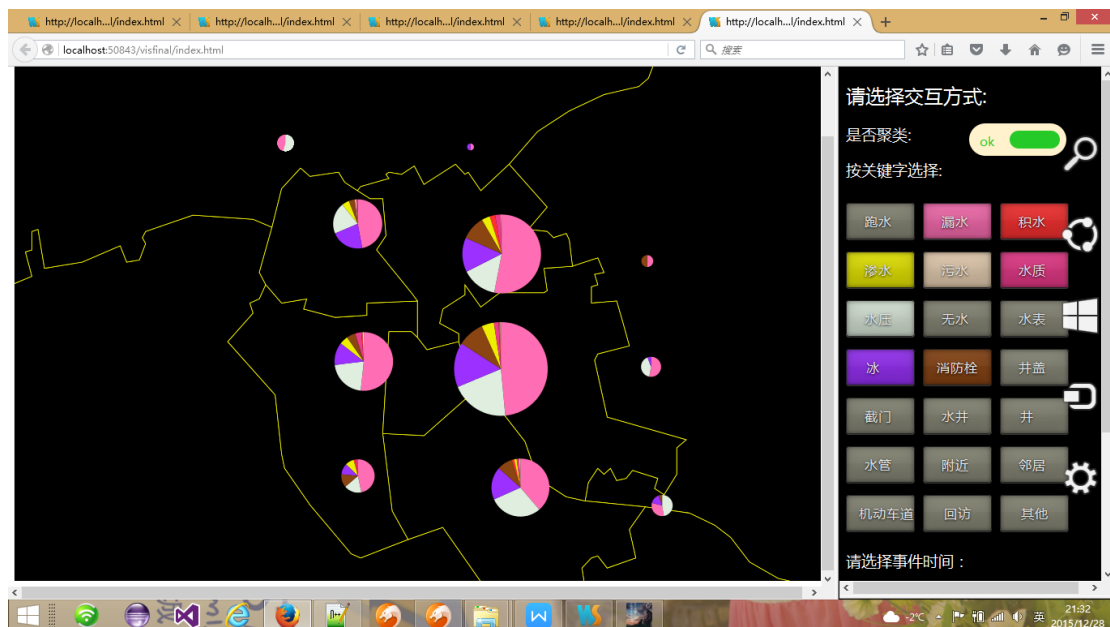
三大类 20 小类。三角，正方形，圆形代表三大类（水，设备，范围、同和 other），20 种颜色代表不同小类。



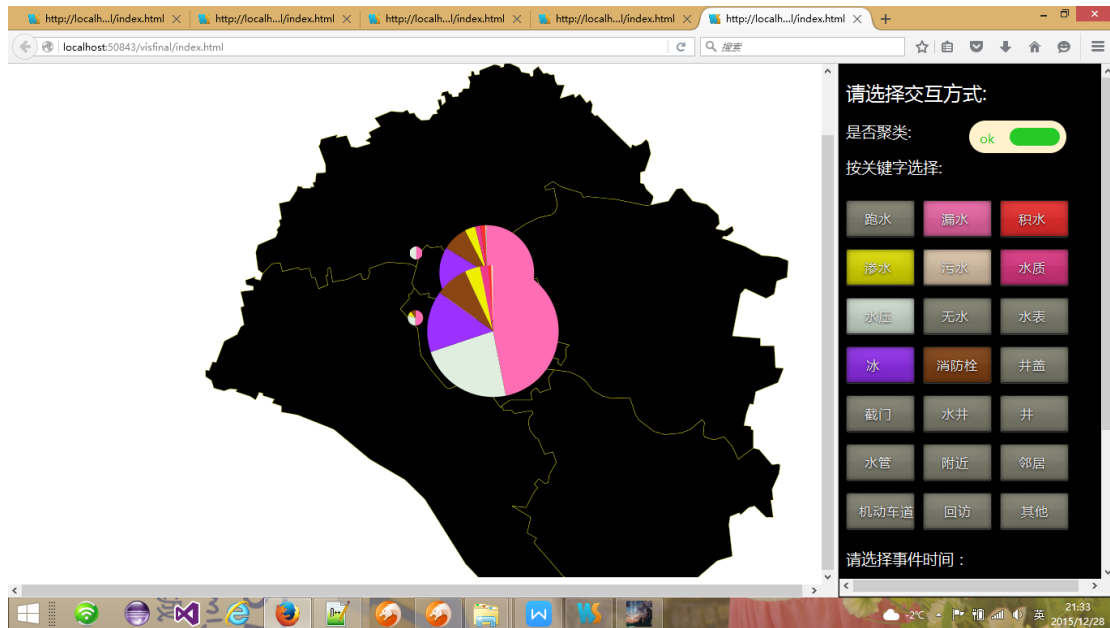
(4) 多事件聚类状态

饼状图聚类。（相对 kmean-s 聚类的优势，速度快，而且 keans 聚类含义不明确。相同种类事件聚类，一个是聚类怎样按区域划分不好确定，一个是同一个区域内有 20 多个事件会显得杂乱无章，按区域聚类。所以饼状图聚类有优势）

当数据超过 1000 时自动聚类。



(5) 缩小状态



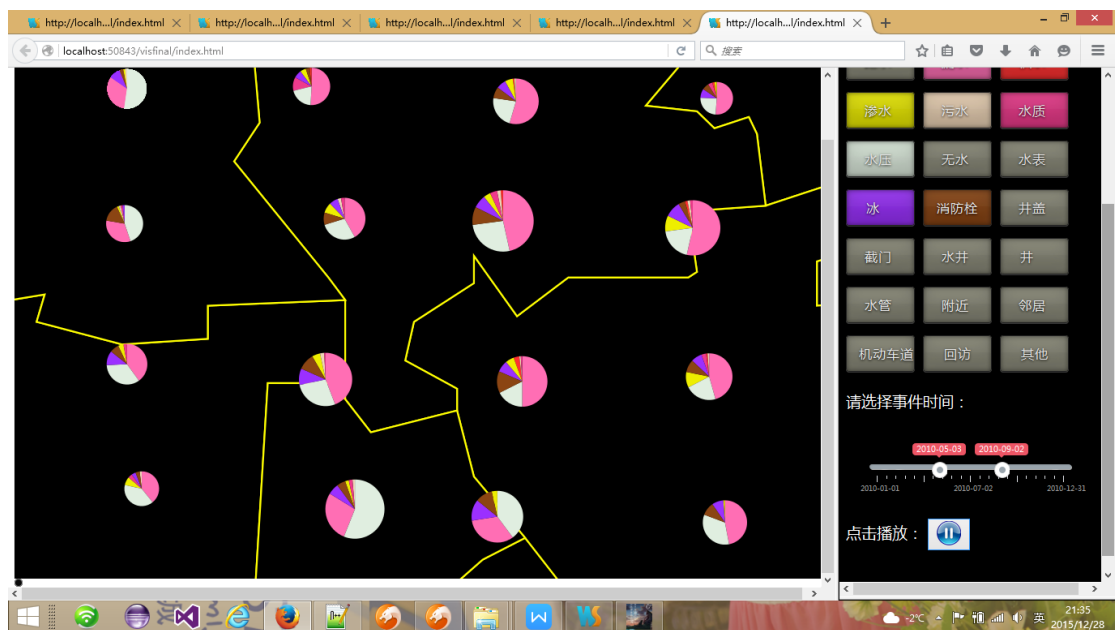
(6) 放大

Zoom 放缩按鼠标为中心放缩，而不是一直以屏幕中心放缩。



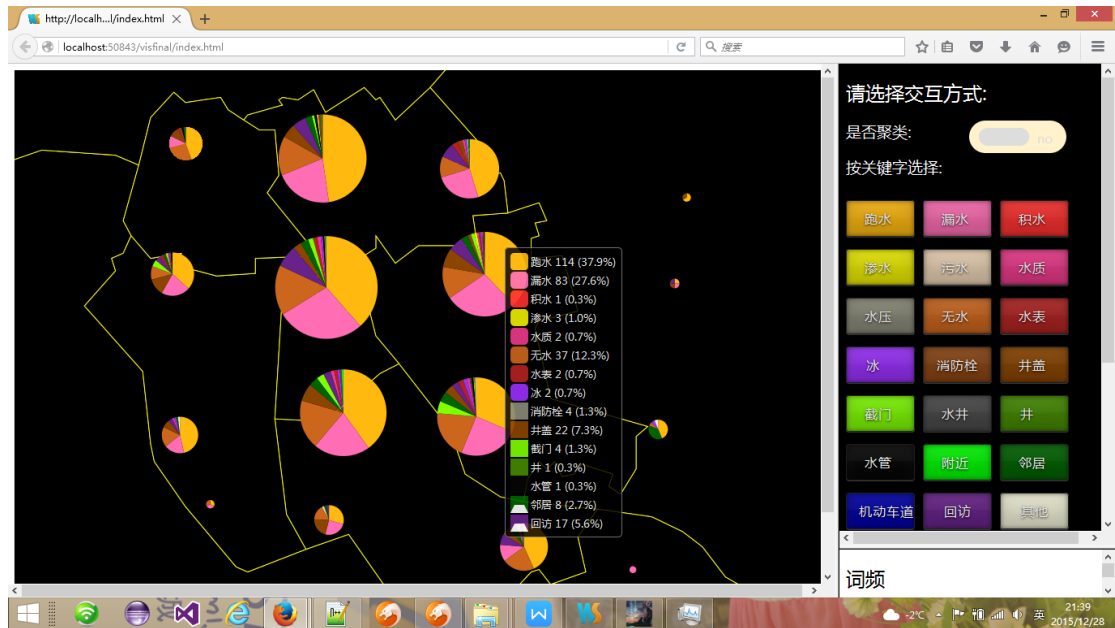
(7) 随时间演变

动态时间浏览，要求动画同时不能卡死，允许交互（不了解 js 不能写异步线程）使用 `setTimeout` 伪异步执行。



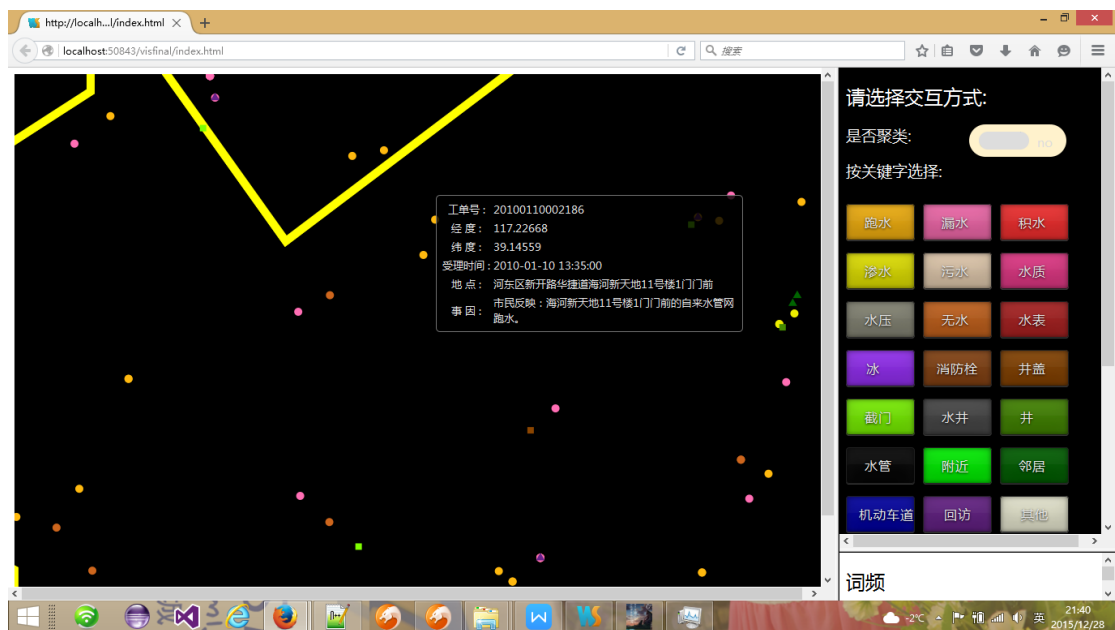
(8) 鼠标悬停在聚类事件上方

使用事件、时间进行筛选，在地图上进行投影映射，使用聚类查看统计结果，以及事件分布情况。



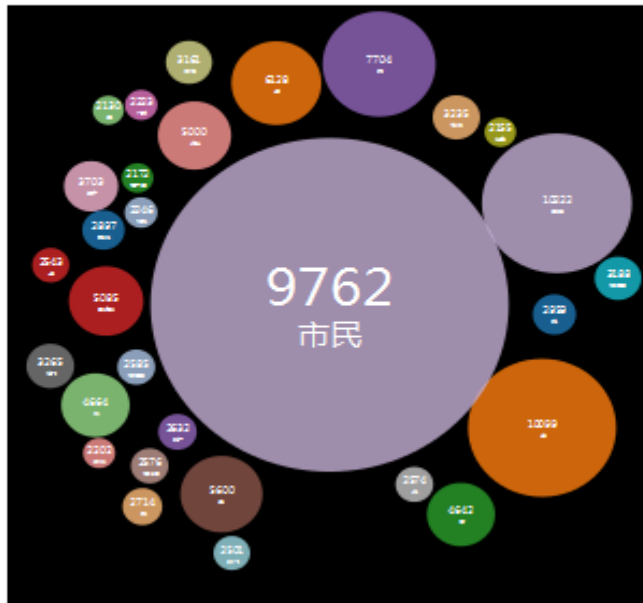
(9) 鼠标悬停在单个事件上方

使用事件、时间进行筛选，在地图上进行投影映射，不聚类时的 toolBar 查看数据详情



(10) 词频部分

词频



最后是火狐与 google 的兼容。对于一些 js 的风格，两种浏览器解释的不相同。

比如

```
{  
  var width = 1000;  
  var height = 700;  
  var svg = d3.select("body").append("svg")  
    .attr("width", width)  
    .attr("height", height)  
    .attr("fill", "#000000");  
  svg.call(d3.behavior.zoom().scaleExtent([0.3, 10]).on("zoom",  
onZoom));  
}
```

这在 google 中定义的是全局量，在 Firefox 中定义的是局部变量。

在 google 中 Date("2010-10-10")可以正确创建 Date，而 Firefox 必须用 Date("2010/10/10")。

四、程序安装及部署：

本项目全部代码和文件封装在 visfinal 文件夹下，该项目文件夹包含 5 个文件夹：.idea、css、img、js、lib、tool。css 文件夹下存放主要的样式代码，img 文件夹下存放所用到的图片，js 文件夹下存放主要的 javascript 代码，lib 包含所用到的库，除此之外，还有 5 个 html 文件，index.html 是主页面。

通过 frameset 技术将页面分成左右两部分。left_map.html 是嵌在主页面左边的地图显示页面。right.html 是右部页面，在 right.html 页面下又分成上下两部分。上面部分为工具栏部分，下面部分为词频显示部分。通过鼠标移动下部页面的边框即可将其拖动到上面。

安装该程序只需下载 visfianl 项目。并将其放在本地服务器运行目录下，用 webstorm 或者 visual studio 等程序编译软件打开该项目文件，并启动本地服务器。在浏览器中输入 <http://localhost:8080>(服务器端口号) /visfinal/index.html 即可打开主页面。本项目的操作，视图均在该页面显示。打开页面后用户就可使用该软件了。