

天津某百货大楼内部相继出现 5 例新冠肺炎确诊病例，从起初的 3 个病例来看，似乎找不到任何流行病学上的关联性。在这种背景之下，作为技术人员可以通过什么技术来找寻病例之间的联系呢？

摘要

最初，nCoV 新冠病毒的扩散过程是由一个人（节点）向各其他人（节点）扩散的树状结构，但随着病毒的变异和人员交叉感染、“百家宴”、“联欢会”、“春运”等人员聚集，疫情扩散为网状结构。可以使用图数据库来存储相关人员、地理位置、感染时间等数据，本文将使用图数据库 **Nebula Graph** 作为工具，带大家一起探讨疫情的传播路径，并找到相关的疑似病例。

天津案例简述

下面用 Usr1、Usr2、Usr3、Usr4、Usr5 来代指这 5 例病例，看一下他们的行为轨迹：

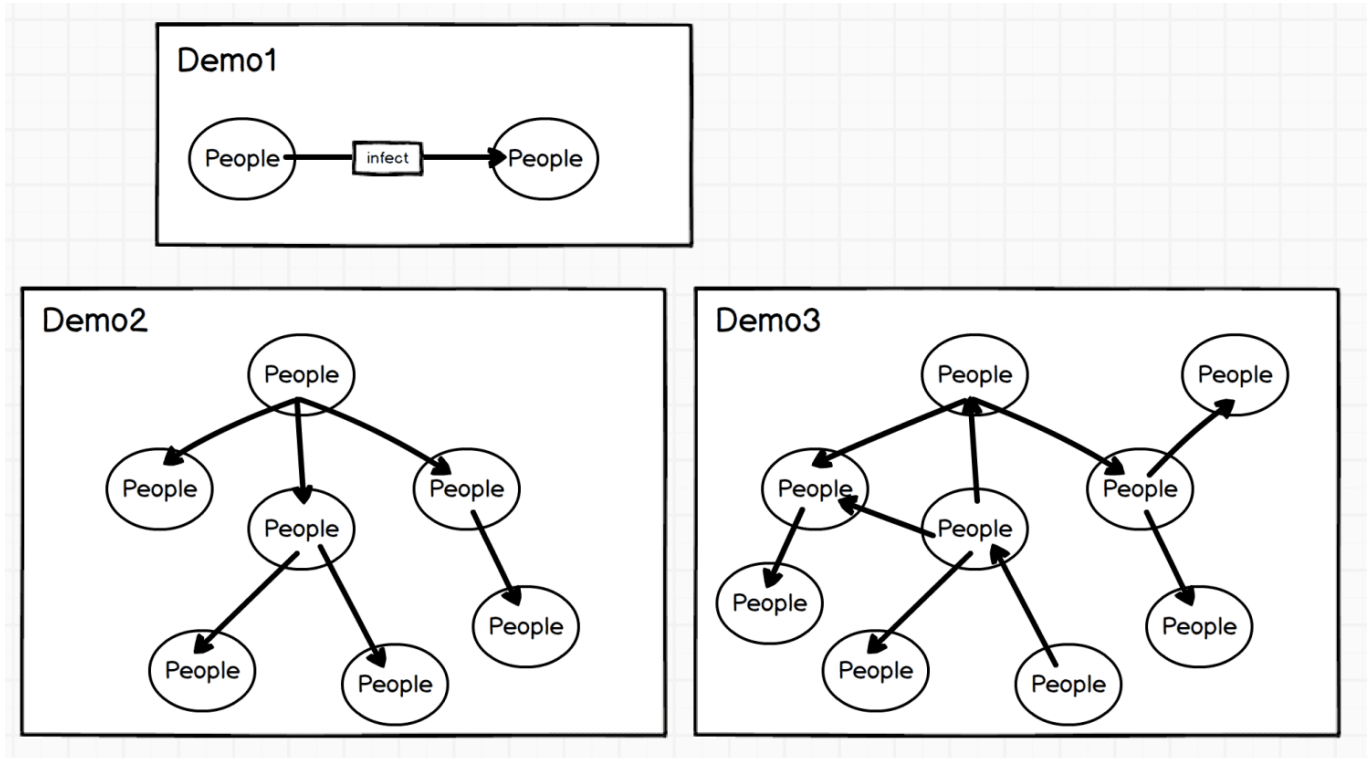
Usr1 信息：

- Usr1 于 1 月 24 日开始发热，在 1 月 22 日至 1 月 30 日期间在天津百货大厦 A 区工作，于 1 月 31 日确诊；
- Usr2 信息：Usr2 为 Usr1 丈夫，于 1 月 25 日开始出现腹泻症状，于 2 月 1 日确诊；
- Usr3 信息：Usr3 于 1 月 18 日接触过一个疑似病例，而后在天津百货大厦 B 区工作，于 1 月 24 日开始发热，于 2 月 1 日确诊；
- Usr4 信息：Usr4 于 1 月 12 日、13 日接触过疑似病例，而后在天津百货大厦 C 区工作，于 1 月 21 日开始发热，于 2 月 1 日确诊；
- Usr5 信息：Usr5 于 1 月 23 日下午 16 点到 23 点到过天津百货大厦 A、B、C 区，1 月 29 日开始发热，2 月 2 日确诊；

下面我们来建立一个传播路径的模型。

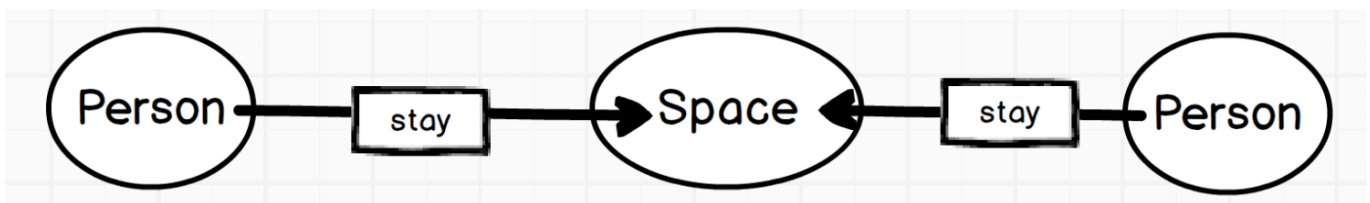
nCoV 新冠病的数据分析

以我们现有的资料显示，本次 nCoV 的传播路径为人传人（图 Demo1），即一个点通过特定访问路径连接到一个点。单个节点看来传播路径为一个树形结构（图 Demo2）——确诊病人 A 感染 B，B 再感染 C，C 再感染 D...。根据现在疫情传播情况，存在多个确诊病人，所以整个传播链路呈网状结构（图 Demo3）。而无论是树形结构还是网状结构都很适合用图（网络）这种数据结构来存储、查询和分析。



图模型

在建模之前我们需要清楚人和人之间的关系载体是什么？根据现有的病例信息，我们知道 A 和 B 会的接触场景最常见的是：同一个时间段逗留在某个相同的空间。这也是本次疫情筛选需隔离人群的重要指标：是否和确诊 / 疑似病例在酒店、火车、超市有过密切接触。



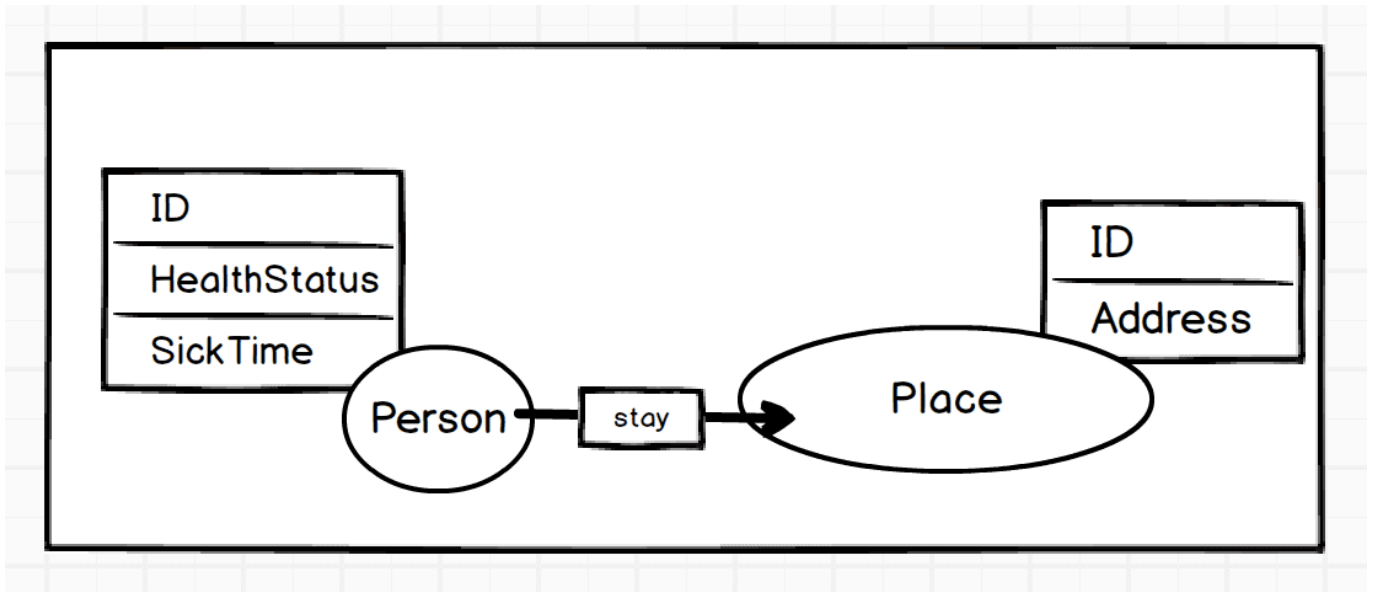
可见最小模型中有两类节点 **Person** 和 **Space**，关系为 **stay**。最小模型有了，那么我们需要 Person 和 Space 的什么信息呢？

Person 类型节点的属性：

- *ID*: Person 的身份证，用来标识人
- *HealthStatus*: 健康状态，有 2 种状态
 - Health: 健康
 - Sick: 生病
- *SickTime*: 发热开始时间，可以用来追溯病人发病的先后次序

Space 类型节点的属性:

- *ID*: Space ID, 用来唯一标识 space
- *Address*: space 地址



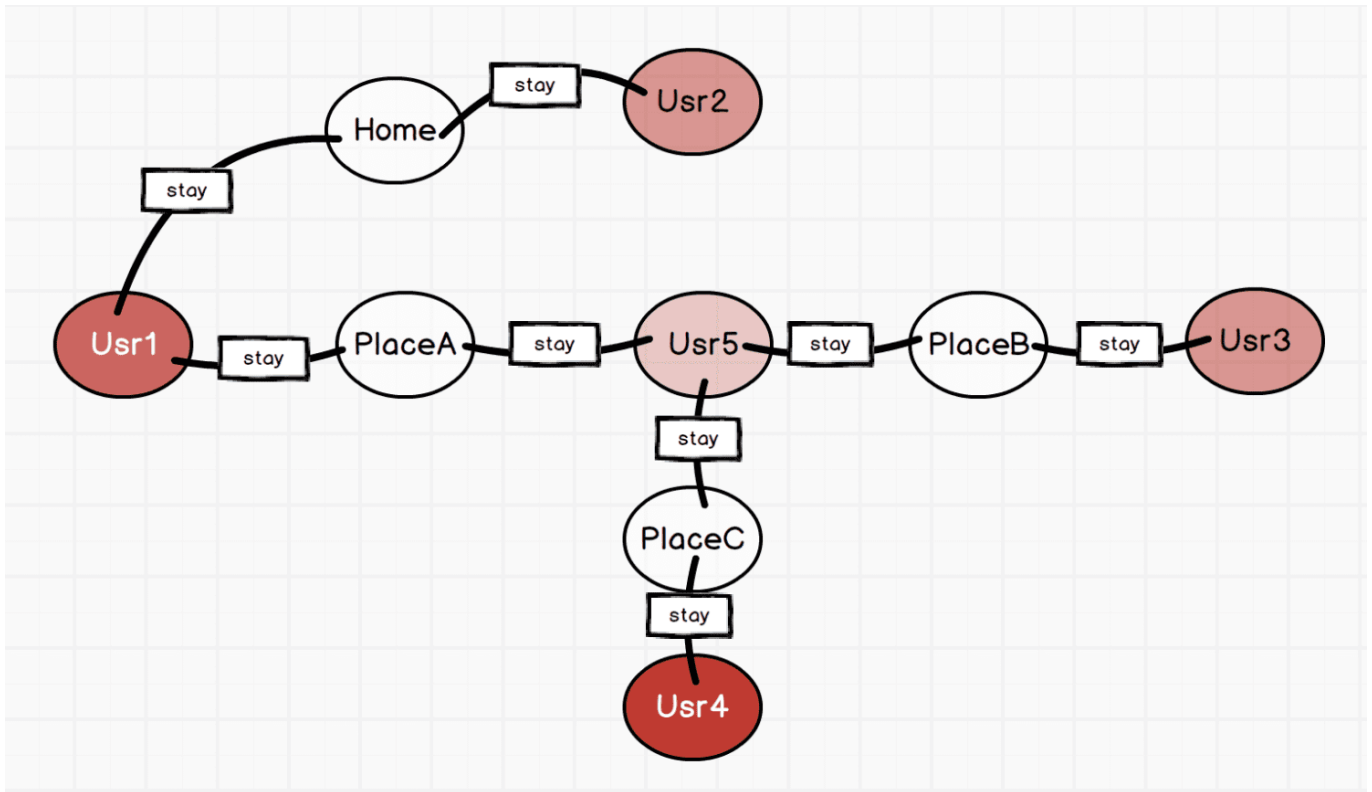
我们构建完 Person 和 Space 的模型之后, 再构建人和位置之间的关系:

在 stay 关系上, 记录有逗留的起始时间和终止时间。这样就可以帮助我们判断两个人是否有过时间和空间上的交集。

天津案例的建模

构建完最小模型之后, 我们来分析一下天津病例中的信息, 将模型应用在这个案例中。并通过图数据库 Nebula Graph 构建病例间关系、找寻病例1的发病原因——病例1怎么被传染的, 及病例1确诊后我们需要观察/隔离哪些人?

整个模型的示意如下:



数据录入

Usr1:

- Person 信息：ID 2020020201, HealthStatus: Sick, SickTime: 20200124;
- Stay Time: 起始时间 1 月 23 日 12 点, 终止时间 18 点;
- Place 信息：天津百货大厦 A 区;
- Stay Time: 起始时间 1 月 23 日 18 点, 终止时间 24 日 8 点;
- Place 信息：天津市和平区 A 小区;

Usr2:

- Person 信息：ID 2020020202, HealthStatus: Sick, SickTime: 20200125;
- Stay Time: 起始时间 1 月 23 日 12 点, 终止时间 23 点;
- Place 信息：天津市和平区 A 小区;

Usr3:

- Person 信息：ID 2020020203, HealthStatus: Sick, SickTime: 20200125;
- Stay Time: 起始时间 1 月 23 日 15 点, 终止时间 19 点;
- Place 信息：天津百货大厦 B 区;
- Stay Time: 起始时间 1 月 23 日 12 点, 终止时间 23 点;
- Place 信息：天津市河西区 B 小区;

Usr4:

- Person 信息：ID 2020020204, HealthStatus: Sick, SickTime: 20200121;
- Stay Time: 起始时间 1 月 23 日 11 点, 终止时间 20 点;
- Place 信息：天津南开区某火锅店;
- Stay Time: 起始时间 1 月 23 日 20 点, 终止时间 23 点;

- Place 信息：天津市滨海区 B 小区；

Usr5:

- Person 信息：ID 2020020205, HealthStatus: Health, SickTime: NULL (无) ；
- Stay Time: 起始时间 1 月 23 日 11 点, 终止时间 15 点；
- Place 信息：天津南开区某火锅店；
- Stay Time: 起始时间 1 月 23 日 16 点, 终止时间 23 点；
- Place 信息：天津百货大厦 A、B、C 区；

将它导入到 NebulaGraph 中，建立人和空间之间的关系。这里以 Usr1 的轨迹为例，其余几份病例类似。

```
-- 插入 Usr1
INSERT VERTEX person(ID, HealthStatus, SickTime) VALUES 1:(2020020201,
'Sick', '2020-01-24');
-- 插入 位置 “天津百货大厦 A 区”
INSERT VERTEX place(name) VALUES 101:("天津百货大厦 A 区")
-- Usr1 到 “天津百货大厦 A 区”
INSERT EDGE stay (start_time, end_time) VALUES 1 -> 101: ('2020-01-23
12:00:00', '2020-01-23 18:00:00')
-- 插入 位置 “天津市和平区 A 小区”
INSERT VERTEX place(name) VALUES 102:("天津市和平区 A 小区")
-- Usr1 回家
INSERT EDGE stay (start_time, end_time) VALUES 1 -> 102: ('2020-01-23
18:00:00', '2020-01-24 8:00:00')
```

病例数据分析

数据导入后，让我们一步步揭开病例1被感染之谜：

1. 查询 Usr1 在 1 月 23 日 15 点至 23 点之间去过哪里

```
$PlaceUsr1Goto = GO FROM 1 OVER stay WHERE stay.start_time > '2020-01-23
00:00:00' AND
stay.start_time < '2020-01-24 00:00:00'
YIELD stay._dst AS placeid
```

```
=====
| Place                | star_time                | end_time                |
=====
| 天津百货大厦 A 区    | 2020-01-23 12:00:00     | 2020-01-23 18:00:00    |
-----
| 天津市和平区 A 小区  | 2020-01-23 18:00:00     | 2020-01-24 8:00:00     |
-----
```

2. 查询这段时间 Usr1 是否接触过任何（已发病的）病例

```
GO FROM $PlaceUsr1Goto OVER stay REVERSELY WHERE $.person.HealthStatus ==
'Sick'
  AND $.person.SickTime <= "2020-01-23"
```

```
=====
| Person          | HealthStatus          | SickTime              |
=====
| 2               | Sick                  | 2020-01-25           |
-----
| 5               | Health                | NULL                  |
-----
```

很奇怪，在 Usr1 发病的时候(2020-01-24)，他接触的人群里面并没有发热患者。那会不会是这些人又接触过其他的患者呢（从而成为携带者），让我们继续分析。

3. 查询这些人又接触过谁

```
$PersonUsr1Meet = GO FROM $PlaceUsr1Goto OVER stay REVERSELY YIELD
stay._dst AS id
$PlaceThosePersonGoto = GO FROM $PersonUsr1meet.id OVER stay YIELD
stay.start_time AS start
  stay.end_time AS end
```

```
=====
| Person | Place                | start_time            | end_time              |
=====
| 5      | 天津南开区某火锅店 | 2020-01-23 11:00:00 | 2020-01-23 15:00:00 |
-----
| 5      | 天津百货大厦 A 区   | 2020-01-23 16:00:00 | 2020-01-23 23:00:00 |
-----
| 5      | 天津百货大厦 B 区   | 2020-01-23 16:00:00 | 2020-01-23 23:00:00 |
-----
| 5      | 天津百货大厦 C 区   | 2020-01-23 16:00:00 | 2020-01-23 23:00:00 |
-----
```

```
GO FROM $PlaceThosePersonGoto.id FROM stay REVERSELY
  WHERE $.person.HealthStatus == 'Sick'
  AND $.person.SickTime <= "2020-01-23" -- 在此之前已经发病
  AND ((stay.start_time > $PlaceThosePersonGoto.start
        AND stay.start_time < $PlaceThosePersonGoto.end)
        OR (stay.end_time > $PlaceThosePersonGoto.start
            AND stay.end_time < $PlaceThosePersonGoto.end) -- 并且有过接触
  )
```

```
=====
| Person1 | HealthStatus | SickTime | stay.start_time | Place
| Person2 |
=====
```

```
| 4          | Sick          | 2020-01-21 | 2020-01-23 11:00:00 | 天津南开区某火锅店
店 | 5          |              |                    |                    |
-----
-----
```

我们发现，虽然 Usr1 在 1 月 23 日 12 点到 1 月 24 日 8 点之间接触的人 (Usr2, Usr5) 都还没有发热，但是 Usr5 却在之前接触过发热病人 Usr4。至此，我们找到了这条传播链路：

Usr4 在 1 月 21 日发病。发病后，他仍前往天津南开区某火锅店（1 月 23 日 11 点- 20 点）。在这里，他接触到（当时健康的）Usr5（1 月 23 日 11 点-15 点）。在接触过程中使得 Usr5 成为一个携带者。之后 Usr5 前往天津百货大厦 A、B、C 区（1 月 23 日 16 - 23 点），在这段时间内，他将病毒传染给在 A 区上班的 Usr1（1 月 23 日 12 点 - 18 点）。最终 Usr1 在 1 月 24 日发病。

4. 之后排查需要隔离哪些人

Usr1 确诊之后，我们需要查看她在哪些时候到过哪些地方。而对应这个时间段相同地点内，又有哪些人同她接触。我们判断这些亲密接触者，需要重点隔离和观察。

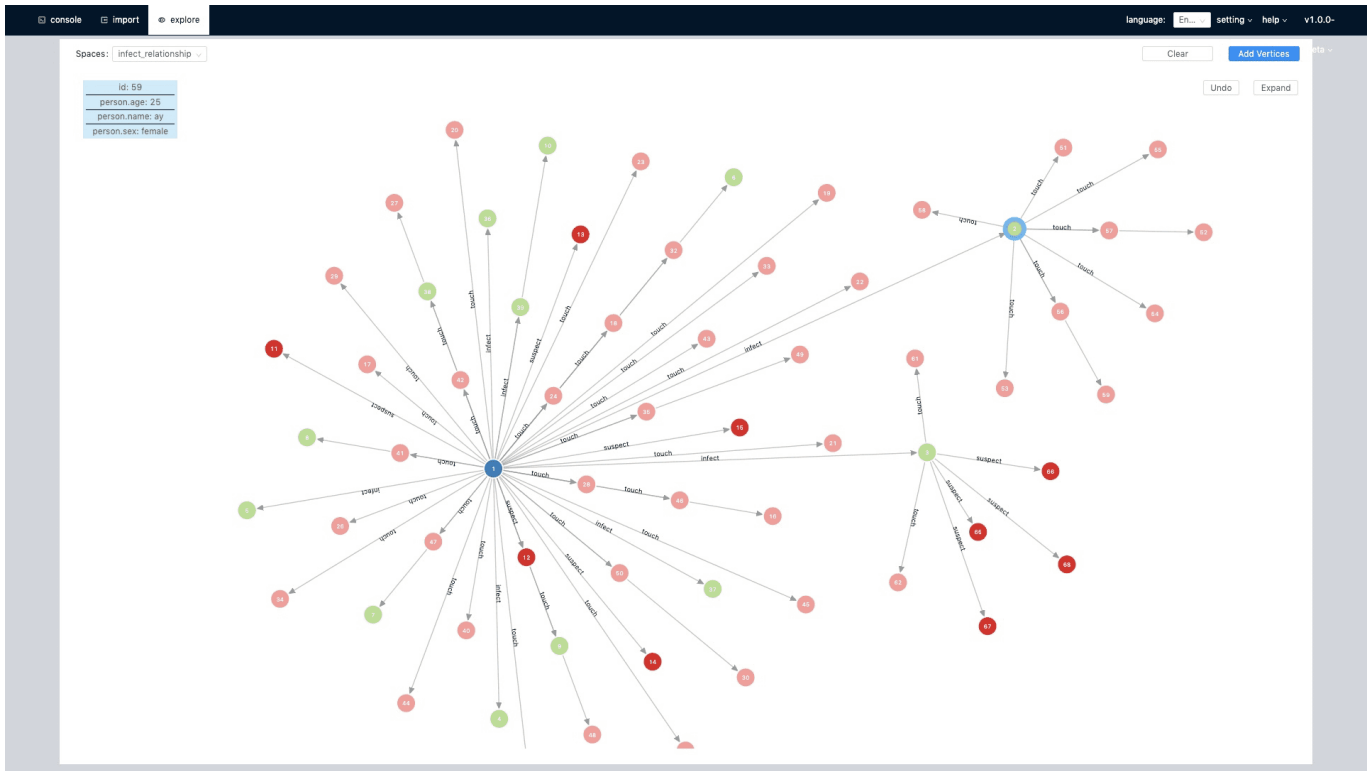
```
GO FROM 1 OVER stay YIELD stay.start_time AS usr1_start,
stay.end_time AS usr1_end, stay._dst AS placeid
| GO FROM $placeid OVER stay REVERSELY WHERE
(stay.start_time > usr1_start AND stay.start_time < usr1_end)
OR (stay.end_time < usr1_start AND stay.end_time < usr1_end)
YIELD $.person.ID

=====
===
| Person | Place          | star_time          |
end_time          |
=====
===
| 2      | 天津市和平区 A 小区 | 2020-01-23 12:00:00 | 2020-01-23 13:00:00
|
-----
---
```

可以发现 Usr1 和 Usr2 在天津市和平区 A 小区有过交集，这使得 Usr2 需要被重点观察。

传播路径可视化展示

上面这段分析过程，也可以使用图形化界面的方式来交互分析，这样更加直观。



当然，如果有非常大批量的关注点（例如上千万离开湖北的潜在人员和他们的二次三次传播轨迹），通过批量程序查询的方式会更加高效。

小结

由于春节返乡和一些不可描述的影响，导致冠状病毒的大面积扩散。从报道和社交媒体上可以看到，各个社区、村庄、企业都采用了相当严格的隔离措施，要求个人每日汇报行踪和健康状态，并密切跟踪从疫区来的人员。这样十几亿人的隔离和追踪需要极大的人力物力和动员能力，充分体现了“集中力量办大事”的制度优越性。

但另外一方面，这样的自我申报和层层统计，非常依赖个人的自觉，也依赖于汇报体系的响应速度。特别是当生死攸关（或者一刀切的歧视政策）的时候，个人反而有很强的动机隐瞒过去的行为和病史，导致未能得到及时的隔离和救治，也极大的影响了需要专业分工合作的现代经济生产活动。

另外一方面，随着大数据技术的发展和智能设备的普及，使得安防、运营商、交通、医疗部门的数据体系已经建立的较为全面。

在天津这个案例中，只选取了少数几个病例和场所作为示意，我们相信如果能够结合前述数据体系，并通过采用新的大数据和人工智能技术，能够极大提高定位和隔离疑似患者的速度，大大减少各种“有效人传人”和“超级毒王”的发生，减少一线医疗和社区工作人员的压力。也能降低全社会的全面隔离时间，尽快恢复经济活动。

参考资料

- 堪比推理小说！天津百货大楼5病例“迷局”：<http://www.bjd.com.cn/a/202002/03/WS5e37d067e4b002ffe994092e.html>
- 图数据库技术 <https://github.com/vesoft-inc/nebula>

推荐阅读

- [使用图数据库 Nebula Graph 数据导入快速体验知识图谱 OwnThink](#)