> 课程链接：CS224W: Machine Learning with Graphs
> 课程视频：【课程】斯坦福 CS224W: 图机器学习 (2019 秋 | 英字)

目录

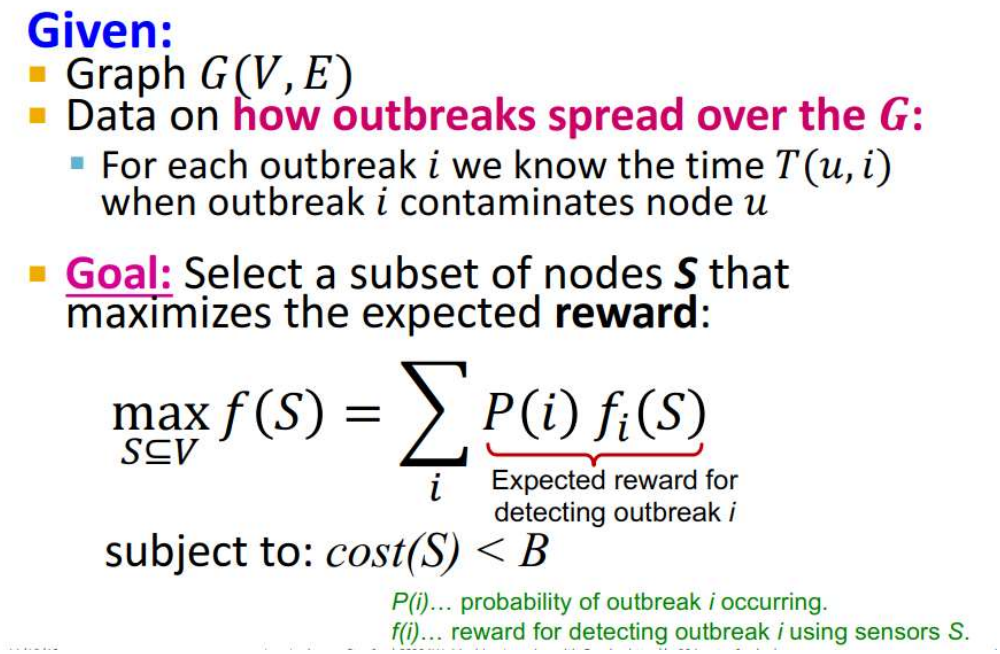
1. 问题的引入——outbreak detection（断点/爆发检测）

问题的引入：Given a dynamic process spreading over a network we want to select a set of nodes to detect the process effectively（给定一个动态的网络，我们怎样确定一组节点用以检测信息传播的过程？）

这个问题在Epidemics、Influence propagation、Network security等方面都是实用的问题。（例如，水污染源的确定，流行病的传染源确定、信息爆发的源头确定、网络安全确定等。）比如说，给定一个水网，你怎样设置检测点，能够快速的检测到水污染的发生。

> Outbreak detection 和 influence maximization 问题很类似，都是在网络中选择一些节点来达到某个目的。以社交网络为例，influence maximization 是为了通过选中的节点使传播更广，而 outbreak detection 是为了通过这些选中的点来及时全面地侦测网络上可能会爆发的话题。——图机器学习 - cs224w Lecture 13 & 14 - 影响力最大化 & 爆发检测

问题的定义：



给定图$G(V, E)$，以及数据在图上的传播方式（也就是对于一个爆发节点i，我们知道从该爆发节点i传播到节点u所需要的时间$T(u, i)$。）

目标是找到一组节点S，使得这组节点传播的消息的效益$f(S) = \sum_i P(i) f_i(S)$最大化。其中$P(i)$表示发生outbreak的概率，$f_i(S)$为对应的收益。如果能够检测所有的点，当然是直接有效的办法。但是，对于每个节点的检测都是需要成本的（比如水网中的检测站）。所以我们可以看到，检测到这个节点集合S需要成本$cost(S)$。这个成本包括舆情管理中对舆论（博客、新闻）等的阅读/检测成本，在水资源污染重对水源的传感器设置成本（Placing a sensor in a remote location is expensive）等。

对于这个效益的定义至关重要，主要从以下三个角度去考虑：

- Minimize time to detection 使检测的时间最短（即能尽早地检测到爆点）
- Maximize number of detected propagations 使检测到的传播数量最多（即能检测到尽量多的爆点）
- Minimize number of infected people 使感染的人数最少（即一旦爆发后，波及的范围尽量小——感觉还是要表达尽早介入的意思）

设$\pi_i(t)$表示在t时间检测到爆点i的惩罚（penalty）。同时，我们希望所有的爆点越快能被检测到越好（In all cases detecting sooner does not hurt!）。我们针对上面提到的三个方面，定义一些符号：

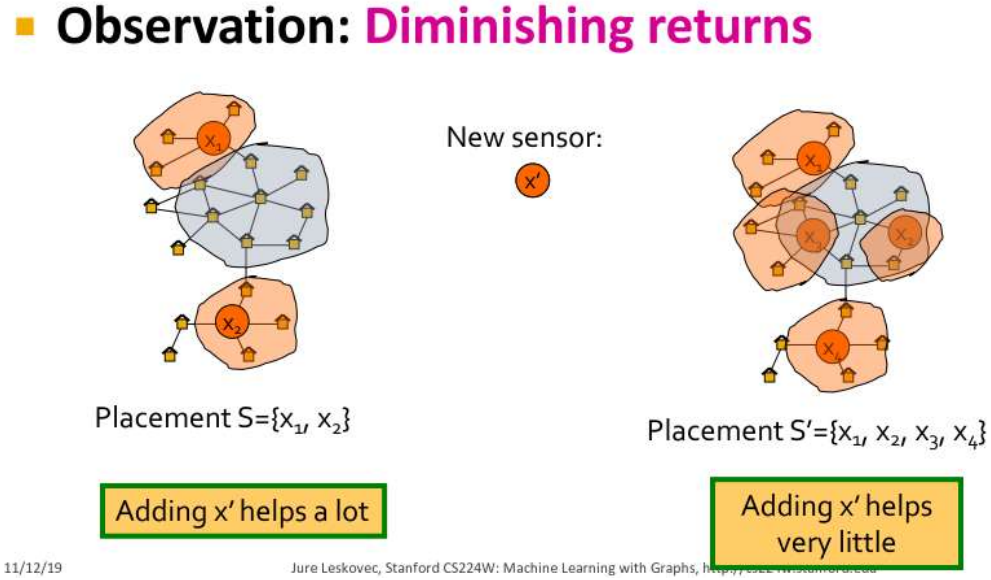
- DT（Time to detection）表示检测到爆点的时间，在时间t检测到爆点的惩罚$\pi_i(t) = t$。
- DL（Detection likelihood）——How many contaminations do we detect?。在时刻t的惩罚为：$\pi_i(t) = 0$或者$\pi_i(t) = \infty$（即我们要么检测，要么不检测）。

- PA（Population affected）表示检测到爆点时，爆点事件已经波及的人数。在时刻t的$\pi_i(t) = \{在这个时刻感染的节点数\}$

我们从penalty reduction的角度来定义效益$f_i(S)$:

$$f_i(S) = \pi_i(\infty) - \pi_i(T(S,i))$$

也就是说，假设我们有一个监测点集合S，对于集合中的每个点i，将它设为监测站的意义在于通过其他节点检测到它所需要的惩罚$\pi_i(\infty)$要大于将它设置为监测站的惩罚。



我们可以假设这是一个水网，对于左边的监测站设置来说，增加一个监测站，可以更好地覆盖这个网络，即能覆盖到中间灰色的部分。而对于右边的监测站设置来说，就没有必要了，再增加一个监测站，就相当于要增加检测成本。

> 这里又出现了 submodular 的身影。把网络设想为一个黑暗的房间，侦测点是一个光源，那么放的第一个光源对这个黑暗房间的意义应该是最大的，因为你从伸手不见五指到能看清房间的一部分了。放第二个的时候你能看到的地方更多了，但它的效果明显没有第一个光源那么有意义。而当我们放入更多光源到一定程度后，整个房间都亮了，再放入光源已经完全没有必要了。——图机器学习 - cs224w Lecture 13 & 14 - 影响力最大化 & 爆发检测

可以证明$f_i(S)$也是一个次模函数（PPT里有），那么我们也可以用贪心算法求得较优解。但是使用贪心的前提是每个监测点的成本是一致的。

2. 贪心算法的改进

这部分我确实没法很好地理解，只能上网找了一些辅助的资料，大概了解了一下。

（1）**CELF算法（Cost-effective Lazy-forward）**——在成本约束下对次模函数的优化算法

（2） **Lazy Evaluation 惰性计算**——爬山算法的加速

（3）**Data Dependent Bound——**对结果的优化

参考文章：CELF算法原理