基于随机图的 Floyd-Warshall 算法运行分析

北京市陈经纶中学 杨熙坤

摘要

Floyd-Warshall 算法是图论中最重要的算法之一，其可以解决结点对之间的最短路径问题。松弛 (relaxation) 操作是 Floyd-Warshall 的重要步骤之一，对于不同的图，松弛操作的操作次数也不同。本文假定一张图是随机的，其中所有边的权重均服从正态分布，即边权重w的取值满足如下概率密度分布函数：

$$P(w=x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

并且依此假定，计算了算法进行松弛操作次数的期望：

$$n_0=\frac{N^3}{2}\operatorname{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)=\Theta(N^3)$$

同时，本文还得到了一个常数，

$$C=\frac{1}{2}\operatorname{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)$$

此常数和算法运行效率正相关，由此，本文证明了算法处理权重分散的图比处理权重集中的图更慢。

关键词

图论、Floyd-Warshall、最短路径、正态分布、松弛 (relaxation)、数学期望、常数分析

基于随机图的 Floyd-Warshall 算法运行分析

北京市陈经纶中学 杨熙坤

一、引入

考虑如下场景，一张图$G=(N,M)$，其中N是结点数 (node)，M是边数。在存储图G时，我们需要一张$N\times N$的矩阵G。最开始时，G_{ij}表示结点$i\to j$的边的权重。

我们希望在本文解决一个问题：在最开始G_{ij}服从正态分布（$G_{ij}\sim N(\mu,\sigma^2)$，$\mu>0$）时，整个算法的松弛操作应该进行多少次（松弛次数的期望）。

对于任意一边$i\to j$，其权重概率密度函数表示为：

$$P(G_{ij}=x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

由于负权重不存在，我们可以估算：

$$\int_0^{+\infty}P(G_{ij}=x)\,\mathrm{d}x\approx\int_{-\infty}^{+\infty}P(G_{ij}=x)\mathrm{d}x=1$$

在最初状态，G_{ij}表示边的权重，如果不存在此条边，可认为$G_{ij}=\infty$。注意，**在运行 Floyd-Warshall 算法时，G_{ij}将表示从结点i到结点j的路线值，其是不断刷新的，在算法运行结束之后，G_{ij}表示从结点i到结点j的最短路线值。**

二、Floyd Warshall 算法的原理

Floyd-Warshall 的本质是动态规划 (dynamic programming)，其拟定图G中一个中间结点k ($k\in\{1,2,\dots,N\}$)，查看$i\to j$的路线能否因为经过k而变得更短，即能否满足三角不等式：

$$G_{ij}>G_{ik}+G_{kj}$$

如果可以，那么其将赋值$G_{ij}=G_{ik}+G_{kj}$，这个操作称为松弛 (relaxing)。对于每个松弛操作，可以认为复杂度为$O(1)$。

具体来讲，其算法思路（伪代码）为：

```
for k = 1 to N
    for i = 1 to N
        for j = 1 to N
            G[i][j] = min(G[i][k] + G[k][j])
```

三、$k=1$循环的分析

对于中间节点$k=1$，其所有边尚未完成松弛，三角不等式判定为：

$$G_{ij} > G_{i1} + G_{1j}$$

我们需要求解此三角不等式成立的概率。首先将不等式写为：

$$G_{ij} - G_{i1} - G_{1j} > 0$$

我们定义$H \coloneqq G_{ij} - G_{i1} - G_{1j}$。由于$G_{ij}, G_{i1}, G_{1j} \sim N(\mu, \sigma^2)$，即：

$$\begin{cases} \mathrm{E}G_{ij} = \mathrm{E}G_{i1} = \mathrm{E}G_{1j} = \mu \\ \mathrm{D}G_{ij} = \mathrm{D}G_{i1} = \mathrm{D}G_{1j} = \sigma^2 \end{cases}$$

根据期望和方差的运算关系

$$\begin{cases} \mathrm{E}(aX + bY) = a\mathrm{E}X + b\mathrm{E}Y \\ \mathrm{D}(aX + bY) = a^2\mathrm{D}X + b^2\mathrm{D}Y \end{cases}$$

可得到

$$\begin{cases} \mathrm{E}H = \mu - \mu - \mu = -\mu \\ \mathrm{D}H = \sigma^2 + \sigma^2 + \sigma^2 = 3\sigma^2 \end{cases}$$

根据正态分布的性质，我们可以得到

$$\mathrm{P}(H > 0) = \int_0^{+\infty} \frac{1}{\sigma\sqrt{6\pi}} \exp\left(-\frac{(x+\mu)^2}{6\sigma^2}\right) \mathrm{d}x$$

借助误差分析函数中的Erfc函数，我们可以得到，

$$\mathrm{P}(G_{ij} > G_{i1} + G_{1j}) = \mathrm{P}(H > 0) = \frac{1}{2}\mathrm{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)$$ [1~2]

因此对于$N \times N$的图，在第一次k循环中，松弛次数的期望$\mathrm{E}[\text{times of relaxation}, k=1]$为：

$$\mathrm{E}[\text{times of relaxation}, k=1] = \frac{N^2}{2}\mathrm{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)$$

---

[1] 成立的前提是$\mathrm{Re}[\sigma^2] > 0$和$\mathrm{Re}[\mu] > 0$。根据假设，这两个条件都符合。

[2] $\mathrm{Erfc}(\cdot)$定义为$\mathrm{Erfc}(\cdot) \coloneqq 1 - \mathrm{Erf}(\cdot)$，其中$\mathrm{Erf}(\cdot)$是误差函数，定义为

$$\mathrm{Erf}\ z = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} \mathrm{d}t$$

在正态分布中，$N(\mu, \sigma^2)$的概率密度分布函数为：

$$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

概率累计分布函数为：

$$\int_{-\infty}^{+\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \mathrm{d}x = \frac{1}{2}\left(1 + \mathrm{Erf}\left(\frac{x-\mu}{\sqrt{2}\sigma}\right)\right)$$

四、$k=2$循环的分析

我们首先需要确定，对于边$i\to j$，经过松弛操作$G_{ij}=G_{ik}+G_{kj}$之后，G_{ij}的概率密度分布是什么样子的。

由于分布函数和具体取值无关，以及第三节所指出的期望、方差运算方法，不难看出经过松弛后，G_{ij}的期望为2μ，方差为$2\sigma^2$，因此概率密度分布为

$$\mathrm{P}(G_{ij}=x)=\frac{1}{\sigma\sqrt{4\pi}}\exp\left(-\frac{(x-\mu)^2}{4\sigma^2}\right)$$

由于在上一步获得松弛的概率为

$$\mathrm{P}(G_{ij}>G_{i1}+G_{1j})=\frac{1}{2}\mathrm{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)$$

所以

$$\mathrm{P}(G_{ij}=x)=\begin{cases}\dfrac{1}{\sigma\sqrt{2\pi}}\exp\left(-\dfrac{(x-\mu)^2}{2\sigma^2}\right), & \mathrm{P}=1-\dfrac{1}{2}\mathrm{Erfc}\left(\dfrac{\mu}{\sqrt{6}\sigma}\right)\\ \dfrac{1}{\sigma\sqrt{4\pi}}\exp\left(-\dfrac{(x-\mu)^2}{4\sigma^2}\right), & \mathrm{P}=\dfrac{1}{2}\mathrm{Erfc}\left(\dfrac{\mu}{\sqrt{6}\sigma}\right)\end{cases}$$

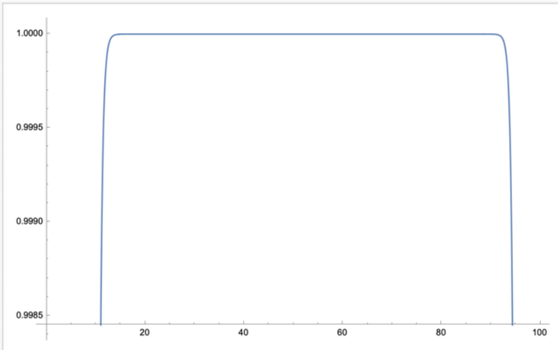
根据全概率公式，$\mathrm{P}(G_{ij}=x)$可以继续化简：

$$\begin{aligned}\mathrm{P}(G_{ij}=x)&=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\left(1-\frac{1}{2}\mathrm{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)\right)\\&\quad+\frac{1}{\sigma\sqrt{4\pi}}\exp\left(-\frac{(x-\mu)^2}{4\sigma^2}\right)\left(\frac{1}{2}\mathrm{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)\right)\\&=\frac{1}{\dfrac{2\sqrt{2}}{2\sqrt{2}+\left(\exp\left(\frac{(x-\mu)^2}{4\sigma^2}\right)-\sqrt{2}\right)\mathrm{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)}\cdot\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\end{aligned}$$

我们针对

$$L:=\frac{2\sqrt{2}}{2\sqrt{2}+\left(\exp\left(\frac{(x-\mu)^2}{4\sigma^2}\right)-\sqrt{2}\right)\mathrm{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)}$$

绘制函数$L-x$的函数图像。见 *Fig* 1,2。



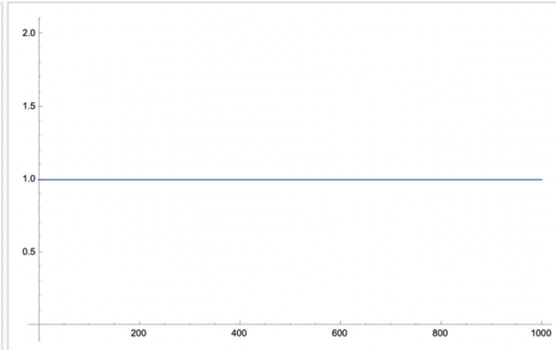


Fig 1,2. $x \in (0,100)$和$x \in (0,1\ 000)$时的$L-x$图。两张图均选择了$\mu = x_{\max}/2$，$\sigma = 2.5$。事实上，不同的μ,σ对图像影响不大。可以认为在一般的图中，$L=1$。

误差允许，在算法分析中，我们可以直接选择

$$L \approx 1$$

即：

$$\mathrm{P}(G_{ij}=x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

因此G_{ij}在松弛后仍满足$G_{ij}\sim N(\mu,\sigma^2)$。

根据第三节的计算，$k=2$循环松弛次数的期望为：

$$\mathrm{E}[\text{times of relaxation}, k=2]=\frac{N^2}{2}\mathrm{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)$$

五、$3 \le k \le N$循环的分析

我们在三、四节确定了如下结论：

1. 不论如何松弛，任意一条边$i \to j$的权重均满足$G_{ij}\sim N(\mu,\sigma^2)$；

2. 任意一次k循环，进行松弛操作的期望次数均为

$$n=\frac{N^2}{2}\mathrm{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)$$

这在往后的循环中依然适用。

六、结论

因此，Floyd-Warshall 运行完成后，其进行松弛操作的总期望次数为：

$$n_0 = Nn = \frac{N^3}{2}\,\mathrm{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)$$

Floyd-Warshall 算法的本身时间复杂度为$\Theta(N^3)$，本论文证明了其松弛操作的运行次数也为$\Theta(N^3)$。

七、常数分析

我们注意到存在一个常数

$$C = \frac{1}{2}\,\mathrm{Erfc}\left(\frac{\mu}{\sqrt{6}\sigma}\right)$$

是本文章分析的产物。其不会决定算法的渐进复杂度，但是在N较大时仍会影响算法的运行。

根据Erf函数的性质，可知Erfc函数是减函数，因此$\mu\sigma^{-1}$的值越大越利于算法的运行。在保证μ稳定的情况下，σ越大，即边权重值越分散，算法运行效率越低；反之，σ越小，即边权重值越集中，算法运行效率越高。

参考文献

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. *Introduction to Algorithms*. 3rd Edition.

[2] Normal distribution. Wikipedia

[3] Error function. Wikipedia