The Story of Dijkstra's

2024 春 - 人工智能原理与技术

May 28, 2024

问题陈述

现在的问题是：给定源点 $v_0 = s$ 和汇点 $v_n = t$（这里的 n 是不定项。另外，后面 v_0^* 和 v_n^* 也同样指的是 s 和 t，只不过书写起来更整齐所以会加星号），我们希望找到一个最佳的顶点序列 $p_{s,t}^* = [v_0^*, ..., v_n^*]$，使得路径的 Total Cost

$$\mathcal{L}(p_{s,t}) = \sum_{i=0}^{n-1} w_{v_i, v_{i+1}}$$

能够被最小化（我们默认 j 不是 i 的邻居时，$w_{ij} = +\infty$）。也就是说，$p_{s,t}^*$ 是如下方程的解

$$p_{s,t}^* \in \underset{p_{s,t}[0]=s,\, p_{s,t}[-1]=t}{argmin} \mathcal{L}(p_{s,t})$$

之所以用属于符号而不是等于符号，是因为可能有多个 p 能使得总代价最小，而将 argmin 视为一个集合也是更加严谨和一般的表述。

动态规划与 Bellman Equation

首先一个非常显然的观察是，对于任意一条始于 s 终于 t 的路径 $p_{s,t}$，它的代价总是可以被表示为如下形式

$$\underbrace{\mathcal{L}(p_{s,t})}_{A} = w_{v_0,v_1} + \underbrace{\mathcal{L}(p_{v_1,t})}_{B}$$

我们不难发现左右两边的 A, B 其实是具有相同的数学结构的。并且由于最优解 p^* 的性质，我们同时也能看到一个不等式

$$\mathcal{L}(p_{s,t}^*) \leq \mathcal{L}(p_{s,t}) = w_{v_0,v_1} + \mathcal{L}(p_{v_1,t})$$

是恒成立的。由于 $\mathcal{L}(p_{s,t}^*), w_{v_0,v_1}$ 都是常数，则由 $\mathcal{L}(p_{v_1,t}^*)$ 的性质，我们又可以导出一个不等式链（即对恒成立问题取下确界）

$$\underbrace{\mathcal{L}(p_{s,t}^*) \leq w_{v_0,v_1} + \mathcal{L}(p_{v_1,t}^*)}_{C} = w_{v_0,v_1} + \underset{p_{v_1,t}[0]=s,\, p_{v_1,t}[-1]=t}{min} \mathcal{L}(p_{v_1,t})$$

对任意的 v_1 是恒成立的。用同样的思路，我们把 C 处的 RHS 对 v_1 再一次取下确界，就得到了这么一个不等式

$$\mathcal{L}(p_{s,t}^*) \leq min_{v_1 \in V}\{w_{v_0,v_1} + \mathcal{L}(p_{v_1,t}^*)\}$$

或者等价表述为

$$\mathcal{L}(p^*_{s,t}) \le min_{v\in V}\{w_{v_0,v} + \mathcal{L}(p^*_{v,t})\} \tag{1}$$

而将 $\mathcal{L}(p^*_{s,t})$ 的表达式展开，我们又知道

$$\mathcal{L}(p^*_{s,t}) = w_{v_0^*,v_1^*} + \mathcal{L}(p_{v_1^*,t}) \ge w_{v_0^*,v_1^*} + \mathcal{L}(p^*_{v_1^*,t}) \ge min_{v\in V}\{w_{v_0,v} + \mathcal{L}(p^*_{v,t})\} \tag{2}$$

这里 $\mathcal{L}(p_{v_1^*,t}) = \sum_{i=1}^{n-1} w_{v_i^*,v_{i+1}^*}$ 是给定路径 $[v_1^*, ..., v_n^*]$ 的总代价，$\mathcal{L}(p^*_{v_1^*,t})$ 则是由 v_1^* 到 $v_n^* = t$ 的最短路径长度。

结合 (1) 和 (2) 两侧的不等式，我们便证明了最短路径所满足的（Bellman）方程

$$\mathcal{L}(p^*_{s,t}) = min_{v\in V}\{w_{v_0,v} + \mathcal{L}(p^*_{v,t})\} \tag{3}$$

类似地，我们可以从 t 的一侧导出另一种形式的 Bellman Equation

$$\mathcal{L}(p^*_{s,t}) = min_{v\in V}\{\mathcal{L}(p^*_{s,v}) + w_{v,t}\} \tag{4}$$

（思考：在强化学习中，我们同样有期望形式下的 Bellman 方程和 Bellman 算子。它们可以被类似地从动态规划问题里导出吗?）

更好的选择？

直接从顶点全集 V 搜索 v 来求解等式 (3)（即 Bellman 方程）总归不是一件非常容易的事情。我们不禁要问：是否可以想办法缩小 v 的搜索空间呢？也就是说，会不会存在一些点，它们永远不可能是最优解的 Candidate?

Lemma 1 *给定起始点 s 和终点 t，顶点集 $V = \{v_0, ..., v_{m-1}\}$（$m$ 是图中顶点的总数目）。当图中仅存在正权边时，我们记顶点序列 $[u_0, ..., u_{m-1}]$ 是顶点全集的一个排列（按照最短路径距离 s 的远近排序，而且显然有 $u_0 = s$），即 $\mathcal{L}(p^*_{s,u_k}) \le \mathcal{L}(p^*_{s,u_{k+1}})$。那么对于排在第 j 位的点 u_j，我们有改进版的 Bellman Equation*

$$\mathcal{L}(p^*_{u_0,u_j}) = min_{0\le i<j}\{\mathcal{L}(p^*_{u_0,u_i}) + w_{u_i,u_j}\} \tag{5}$$

这个结果是显然的。否则如果存在 $i > j$（不需要考虑 $i = j$，这一项并不能贡献任何额外的信息）使得 $\mathcal{L}(p^*_{u_0,u_j}) = \mathcal{L}(p^*_{u_0,u_i}) + w_{u_i,u_j}$，集合排序前提我们又知道 $\mathcal{L}(p^*_{u_0,u_j}) = \mathcal{L}(p^*_{u_0,u_i}) + w_{u_i,u_j} \le \mathcal{L}(p^*_{u_0,u_i}) \Rightarrow w_{u_i,u_j} \le 0$，这与图中仅存在正权边矛盾。因此，我们可以很高兴地看到在正权图的假设下（而且这个假设是非常合理的，因为现实中的大多数图数据也都是只考虑正权图的形式），我们的确可以将搜索空间从顶点全集 V 缩小到 $[u_0, ..., u_{j-1}]$（对于 u_j 而言）。

有了这个定理，我们似乎看到了贪心思想的影子。我们如果能设计一个算法，它可以保证我们正确计算出 $[u_0, ..., u_{j-1}]$ 的最短路径（归纳假设）；那么应用上面的定理，我们理应就可以通过贪心地取最小值找到 u_j 的最短路径（归纳推理），因此我们又成功找到了 $[u_0, ..., u_j]$ 这个序列中每个点的最短路径 $\Rightarrow$ 数学归纳法就可以保证这个算法的正确性了！

Dijkstra's: The Art of Labelling Techniques

Dijkstra 实际上就是我们上述的算法。只不过还有一些细节需要做出处理和说明。

Algorithm 1 Dijkstra 算法计算单源最短路径

1: **输入:** 有向图 $G = (V, E)$，权重函数 $w : E \to \mathbb{R}^+$，源点 $s \in V$
2: **输出:** 从源点 s 到所有其他顶点的最短路径距离 $d_s(v)$
3: 初始化所有顶点的标签为无穷大: $d^0(v) \leftarrow \infty$ 对 $\forall v \in V \setminus \{s\}$
4: 初始化源点的标签为 0: $d^0(s) \leftarrow 0$
5: 令集合 $S \leftarrow \emptyset$
6: 令优先队列 Q 包含所有顶点并用 $d^0(v)$ 初始化
7: 令迭代计数器 $k \leftarrow 0$
8: **while** $Q \neq \emptyset$ **do**
9:     从 Q 中取出具有最小标签的顶点 u，即 $d^k(u)$ 最小
10:     将 u 加入集合 S
11:     **for** 每个与 u 相邻的顶点 v **do**
12:         **if** $v \notin S$ **then**
13:             计算候选标签: $d^{k+1}(v) \leftarrow \min(d^k(v), d^k(u) + w(u, v))$
14:             更新优先队列中的标签值
15:         **end if**
16:     **end for**
17:     迭代计数器 $k \leftarrow k + 1$
18: **end while**
19: **返回**所有顶点的最短路径距离 $d_s(v)$

为了证明 Dijkstra's 确实能在第 $k(0 \leq k < m)$ 轮迭代时计算得到升序排列中第 k 个顶点 u_k 距离 s 的最短距离，我们实际上是要说明这么两件事（在伪代码的第 k 轮迭代中）:

- A. 迭代开始前，S 恰为我们先前所定义的顶点序列 $[u_0, ..., u_{k-1}]$；
- B. 结点 u_k 的标签估计 $d^k(u_k)$ 恰好对应它们的真实最短路径长度 $d_s(u_k)$，并且不存在 $l > k$，使得 $d^k(u_l) < d^k(u_k)$。

需要注意的是，d^k 本身是算法对结点距离的一个 Label 估计，在严格证明前我们是不能默认 $d^k = d_s$（即某些结点在某些轮次的标签恰好等于真实的最短路径 d_s）的。下面我们来证明上述这组命题。

Proof 0.1
$t = 0$ 时，A,B 命题显然成立。
假设 $t \leq k$ 时，A,B 命题成立。那么在算法的第九行，从 $Q = V - S = [u_k, ..., u_{m-1}]$（利用假设 A）取出的顶点 u 就应该是 u_k（利用假设 B），从而 S 被扩充为 $[u_0, ..., u_k]$，我们即证明了在 $t = k + 1$ 时，命题 A 是成立的。为了接着证明 B 在下一步仍然保持正确，我们希望说明：在更新 $d^{k+1}(v) \leftarrow \min(d^k(v), d^k(u) + w(u, v))$ 之后，

$$d^{k+1}(u_{k+1}) = d_s(u_{k+1})$$

并且

$$d^{k+1}(u_{k+1}) \leq d^{k+1}(u_l)(\forall l > k + 1)$$

首先不难验证标签估计 $d^k(v)$ 要么是正无穷，要么就是某条 s 到 v 的真实路径长度（不一定是最

短)，因此恒有 $d^k(v) \geq d_s(v)$。与此同时，对于任意的点 $v \in [u_{k+1}, ..., u_{m-1}]$，根据 *Dijkstra* 的算法过程我们知道，它们的标签 d^k 一定是按照顺序由 $S = [u_0, ..., u_{k-1}]$ 中的每个点作为第九行的 u（在不同轮次）更新得到，即

$$d^k(v) = min\{+\infty, d^0(u_0) + w(u_0, v), ..., d^{k-1}(u_{k-1}) + w(u_{k-1}, v)\}$$

由归纳假设 B，我们得知

$$d^k(v) = min\{+\infty, d_s(u_0) + w(u_0, v), ..., d_s(u_{k-1}) + w(u_{k-1}, v)\}$$

在这个基础上，我们按照更新公式的定义，可以进一步推导

$$d^{k+1}(u_{k+1}) = \min(d^k(u_{k+1}), d^k(u) + w(u, u_{k+1}))$$

$$\overset{u=u_k}{=} \min(d^k(u_{k+1}), d^k(u_k) + w(u_k, u_{k+1}))$$

$$\overset{\text{假设 } B}{=} \min(d^k(u_{k+1}), d_s(u_k) + w(u_k, u_{k+1}))$$

$$\overset{\text{刚才论证的}}{=} \min(min\{+\infty, d_s(u_0) + w(u_0, u_{k+1}), ..., d_s(u_{k-1}) + w(u_{k-1}, u_{k+1})\}, d_s(u_k) + w(u_k, u_{k+1}))$$

$$= min\{+\infty, d_s(u_0) + w(u_0, u_{k+1}), ..., d_s(u_{k-1}) + w(u_{k-1}, u_{k+1}), d_s(u_k) + w(u_k, u_{k+1})\}$$

$$= min_{u \in \{u_0, ..., u_k\}}\{d_s(u) + w(u, u_{k+1})\}$$

利用“更好的选择”一章的结论，我们从而证明了

$$d^{k+1}(u_{k+1}) = min_{u \in \{u_0, ..., u_k\}}\{d_s(u) + w(u, u_{k+1})\} = d_s(u_{k+1}) \tag{6}$$

因此命题 B 的前半句在 $t = k + 1$ 时仍然成立。对于其后半句，不难验证对于 $l > k + 1$，我们有

$$d^{k+1}(u_l) \geq d_s(u_l) \geq d_s(u_{k+1}) = d^{k+1}(u_{k+1})$$

因而命题 B 整句在 $t = k + 1$ 时也成立！

到此为止，我们证明了命题 A 和命题 B 在 $t = k + 1$ 步也是成立的。那么由数学归纳法，我们便证明了 *Dijkstra's* 的正确性。