基于线性代数的一般图匹配

周子鑫　　杨家齐

2017 年 1 月 30 日

目录

Section 1

Tutte 定理

定义 1

对于一个无向图 $G=(V,E)$, 定义 G 的 Tutte 矩阵为一个 $n\times n$ 的矩阵 $\tilde{A}(G)$, 其中

$$\tilde{A}(G)_{ij}=\begin{cases} x_{ij} & \text{若 } (i,j)\in E \text{ 并且 } i<j \\ -x_{ji} & \text{若 } (i,j)\in E \text{ 并且 } i>j \\ 0 & \text{其他情况} \end{cases}$$

上文定义中的 x_{ij} 是一个变量. 因此矩阵 A 中总共有 $|E|$ 个变量.

定义 2

图 $G=(V,E)$ 的一个环覆盖是指用若干个环去覆盖图 G 中的所有结点，使得 G 中的任意一个结点恰好在一个环中.

如果这个环覆盖中的所有环的长度都是偶数，那么我们称这是一个偶环覆盖.

定理 1

图 $G=(V,E)$ 有完美匹配当且仅当 G 有一个偶环覆盖.

定理 2 (Tutte)

图 G 有完美匹配当且仅当 $\det \tilde{A}(G) \neq 0$.

Section 2

构造完美匹配方案 (naive approach)

先介绍一个引理.

引理 3 (Schwartz-Zippel)

对于域 $\mathbb{F}$ 上的一个不恒为 0 的 n 元 d 度多项式 $P(x_1, x_2, \ldots, x_n)$,

设 $r_1, r_2, \ldots, r_n$ 为 n 个 $\mathbb{F}$ 中的独立选取的随机数, 则

$$\Pr[P(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|\mathbb{F}|}$$

有了这个定理之后, 我们就可以选 $|E|$ 个 $\mathbb{F}_p$ 中的随机数, 根据 ($\mathbb{F}_p$ 下) $\det \tilde{A}(G)$ 是否为 0 来判断 G 是否存在完美匹配.

这样的话错误概率不超过 $\frac{n}{p}$, 如果我们取 $p = 10^9 + 7$, 由于匹配题一般 n 不超过 500, 因此这个错误概率基本可以忽略不计.

既然我们可以判定图 G 是否有完美匹配了, 那么一个 naive 的想法就是枚举一条边, 然后删了它, 看看剩下的图是否有完美匹配.

Algorithm 1 A naive matching algorithm

$M \leftarrow \emptyset$
for $i \leftarrow 1$ to n **do**
  for $j \leftarrow 1$ to n **do**
    if $v_i v_j \in E(G)$ **and** $\det \tilde{A}(G - \{v_i, v_j\}) \neq 0$ **then**
      $G \leftarrow G - \{v_i, v_j\}$
      $M \leftarrow M \cup \{v_i, v_j\}$
    end if
  end for
end for
return M

如果计算行列式的算法是 $O(\omega)$, 那么上面那个算法是 $O(n^{\omega+2})$ 的, 看起来实在是太暴力了, 那么有没有什么更高效的做法呢?

考虑到每次判定一条边是否合法只用知道 $\det \tilde{A}(G-\{v_i, v_j\})$ 是否为 0, 每次判定要重新计算一次行列式, 效率太低了. 我们考虑用一些线代的知识来加速判定.

Section 3

一点线代知识

定义 3

$A^{i,j}$ 为 A 去掉第 i 行和第 j 列后所剩下的子矩阵.

定义 4

A 关于第 i 行和第 j 列的余子式 (记做 $M_{i,j}$) 是 $\det A^{i,j}$.

定义 5

A 关于第 i 行和第 j 列的代数余子式 (记做 $C_{i,j}$) 是 $(-1)^{i+j}M_{i,j}$.

定义 6

A 的余子矩阵是一个 $n \times n$ 的矩阵 C, 使得其第 i 行第 j 列的元素是 A 关于第 i 行和第 j 列的代数余子式.

定义 7

矩阵 A 的伴随矩阵是 A 的余子矩阵的转置矩阵:

$$\operatorname{adj} A = C^T$$

i.e.

$$(\operatorname{adj} A)_{i,j} = C_{j,i}$$

定理 4 (拉普拉斯展开)

A 为一个 $n \times n$ 的矩阵, 对于任意的一行 $i \in \{1, ..., n\}$ 有:

$$\begin{aligned}\det A &= \sum_{j=1}^{n} (-1)^{i+j} A_{i,j} \det A^{i,j} \\ &= \sum_{j=1}^{n} A_{i,j} (\text{adj } A)_{j,i}\end{aligned}$$

类似的, 对于任意的一列 $j \in \{1, ..., n\}$ 有:

$$\det A = \sum_{j=1}^{n} A_{i,j} (\text{adj } A)_{j,i}$$

定理 5

如果矩阵 A 可逆，那么有:

$$A^{-1} = \operatorname{adj} A / \det A$$

Section 4

反对称矩阵的性质

想必大家对上面讲的这些基础线代知识早就懂了, 接下来我们回到一开始所讲的 Tutte 矩阵. 来看一看它的一些性质.

$$\tilde{A}(G)_{ij} = \begin{cases} x_{ij} & \text{若 } (i,j) \in E \text{ 并且 } i < j \\ -x_{ji} & \text{若 } (i,j) \in E \text{ 并且 } i > j \\ 0 & \text{其他情况} \end{cases}$$

观察到对于任意的 i, j

$$\tilde{A}(G)_{ij} = -\tilde{A}(G)_{ji}$$

定义 8

反对称矩阵是一个 $n \times n$ 的矩阵，其满足:

$$A^T = -A$$

前文所讲的 Tutte 矩阵就是一个反对称矩阵.

引理 6

A 是一个 $n \times n$ 的反对称矩阵，如果 n 是奇数，那么 $\det A = 0$.

定义 9

A 是一个 $n \times n$ 的矩阵, $I = \{i_1, i_2, ..., i_k\}$ 是集合 $\{1, ..., n\}$ 的一个子集.A 的一个主子式 $A_{I,I}$ 为 A 选取 I 中的行和列组成的子矩阵.

定理 7

A 是一个 $n \times n$ 的反对称矩阵, $I = \{i_1, i_2, ..., i_k\}$, $A_{i_1,\cdot}, A_{i_2,\cdot}, ..., A_{i_k,\cdot}$ 是 A 关于行的一组极大线性无关组, 那么 $A_{I,I}$ 的行列式不为 0.

推论 8

A 是一个 $n \times n$ 的反对称矩阵, $I = \{i_1, i_2, ..., i_k\}$, $A_{i_1,\cdot}, A_{i_2,\cdot}, ..., A_{i_k,\cdot}$ 是 A 关于行的一组最大线性无关组, 那么 $A_{I,I}$ 的行列式不为 0.

推论 9

对于一个反对称矩阵 A, 存在一个集合 $I = \{i_1, i_2, ..., i_k\}$, 使得:

$$\operatorname{rank} A = \operatorname{rank} A_{I,I} = k$$

定理 10

一个反对称矩阵的秩为偶数.

引理 11

A 是一个 $n \times n$ 的可逆反对称矩阵. 那么对于任意的 $i, j \in \{1, ..., n\}, i \neq j$ 有: $\det A^{i,j} \neq 0$ 当且仅当 $\det A^{\{i,j\},\{i,j\}} \neq 0$.

Section 5

构造完美匹配方案 (Rabin and Vazirani)

上面说了这么多，是不是有些同学已经睡着了．其实讲了这么多都是为了优化最开始的那个算法：

Algorithm 2 A naive matching algorithm

$M \leftarrow \emptyset$
for $i \leftarrow 1$ to n **do**
  for $j \leftarrow 1$ to n **do**
    if $v_i v_j \in E(G)$ **and** $\det \tilde{A}(G - \{v_i, v_j\}) \neq 0$ **then**
      $G \leftarrow G - \{v_i, v_j\}$
      $M \leftarrow M \cup \{v_i, v_j\}$
    end if
  end for
end for
return M

这个算法的瓶颈在于需要判断每一条边是否在完美匹配中，如果能快速判断与一个点相邻的哪条边是在完美匹配中的，那就很不错了.

结合之前所讲的一些定理及引理，确实能够得到这样的一个算法.

定理 12 (Rabin, Vazirani)

$G=(V,E)$ 是一个有完美匹配的图，$\tilde{A}=\tilde{A}(G)$ 是它对应的 *Tutte* 矩阵. 那么，$(\tilde{A}^{-1})_{i,j}\neq 0$ 当且仅当 $G-\{v_i,v_j\}$ 有完美匹配.

啊哈，有了上面这个定理就可以得到一个更加高效的算法了.

Algorithm 3 Matching algorithm of Rabin and Vazirani

$M \leftarrow \emptyset$
for $i \leftarrow 1$ to n **do**
  if v_i is not yet matched **then**
    compute $\tilde{A}(G)^{-1}$
    find j, such that $v_i v_j \in E(G)$ and $(\tilde{A}(G)^{-1})_{j,i} \neq 0$
    $G \leftarrow G - \{v_i, v_j\}$
    $M \leftarrow M \cup \{v_i, v_j\}$
  end if
end for
return M

这个算法的时间复杂度为 $O(n^{\omega+1})$.

Section 6

构造完美匹配方案 (A simple approach by gaussian elimination)

上一个算法的瓶颈在于求逆. 因此, 如果我们能够通过一些办法来维护矩阵的逆, 使得在删掉一行一列之后, 我们不需要再重新求一遍逆矩阵, 那么上面的算法的复杂度就可以除一个 n.

这实际上是可以做到的.

定理 13

如果

$$A = \begin{pmatrix} a_{1,1} & v^T \\ u & B \end{pmatrix} \qquad A^{-1} = \begin{pmatrix} \hat{a}_{1,1} & \hat{v}^T \\ \hat{u} & \hat{B} \end{pmatrix}$$

其中 $\hat{a}_{1,1} \neq 0$. 那么 $B^{-1} = \hat{B} - \hat{u}\hat{v}^T/\hat{a}_{1,1}$.

有了这个定理, 我们就可以得到如下算法:

Algorithm 4 Simple matching algorithm

```
M ← ∅ compute Ã(G)^{-1}
for i ← 1 to n do
    if the i-th row is not yet eliminated then
        find j such that v_i v_j ∈ E(G) and (Ã(G)^{-1})_{j,i} ≠ 0
        G ← G − {v_i, v_j}
        M ← M ∪ {v_i, v_j}
        update Ã(G)^{-1} by
        eliminating the i-th row and the j-th column
          and then the j-th row and the i-th column
    end if
end for
return M
```

这个算法的时间复杂度是 $O(n^3)$.

Section 7

完美匹配与最大匹配

到目前为止，我们已经能够判定一个图是否存在完美匹配，如果存在我们还能给出一组方案.

但在实 (O) 际 (I) 应 (比) 用 (赛) 中，我们要求的往往是最大匹配.

如何把求最大匹配变成求完美匹配呢？我们可以想办法选出 G 的一个极大的存在完美匹配的子图. 从代数的观点来看，这就是要求出 $\tilde{A}(G)$ 的一个极大满秩子矩阵.

显然，这个子矩阵的秩不超过 $\mathrm{rank}\tilde{A}(G)$. 并且要想达到这个上界，我们只需要做一遍高斯消元，把所有主元所对应的列选出来，即可得到这个点集.

并且我们还可以得到一个求最大匹配大小的方法

推论 14

$\mathrm{rank}\tilde{A}(G)$ 等于最大匹配大小的两倍.

这里大家可以联系一下我们之前讲的线代知识：比如说，我们曾经证明了，$\mathrm{rank}\tilde{A}(G)$ 一定是个偶数，以及，当 G 的结点数为奇数时，$\det \tilde{A}(G)$ 为 0. 可以看到，之前结论和我们现在得到的推论是相洽的.

Section 8

总结

综上, 我们能够用 $O(n^\omega)$ 的复杂度求出最大匹配的大小, 并且能够用 $O(n^3)$ 的复杂度求出最大匹配.

实际上, 通过一些奇怪的方法可以把求最大匹配的复杂度也降到 $O(n^\omega)$, 但那个做法太复杂了, 有兴趣的同学可以看文末给出的参考文献, 这里不再赘述.

谢谢大家!

Section 9

参考文献

1. Rabin M O, Vazirani V V. Maximum matchings in general graphs through randomization[J]. Journal of Algorithms, 1989, 10(4): 557-567.
2. Mucha M, Sankowski P. Maximum matchings via Gaussian elimination[C]. Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on. IEEE, 2004: 248-255. MLA